

# CURSO: LINUX COMO ENTORNO DE TRABAJO EN UN CENTRO EDUCATIVO

Cursos THALES-CICA 2003



José Á. Bernal - Paco Villegas

3 de noviembre de 2003

Derechos de Autor (c) 2003 JOSÉ ÁNGEL BERNAL & PACO VILLEGAS. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes, sin Textos de Portada, y sin Textos al respaldo. Una copia de la licencia es incluida en la sección titulada "Licencia de Documentación Libre GNU".

# Índice general

<b>I</b>	<b>Linux no es ventanas</b>	<b>10</b>
<b>1.</b>	<b>¿Qué es Linux? Posibilidades de Linux. Estructura del Sistema Operativo</b>	<b>11</b>
1.1.	Introducción.	11
1.2.	Una breve historia de Linux	12
1.3.	Posibilidades de Linux	13
1.4.	Distribuciones de Linux	14
1.5.	Estructura del sistema operativo.	15
<b>2.</b>	<b>Linux e Internet: fuentes de información y programas.</b>	<b>18</b>
2.1.	¿Dónde encuentro información?	18
2.1.1.	Documentos y manuales	19
2.1.2.	Web interesantes sobre Linux	21
2.1.3.	News.	22
2.1.4.	Revistas sobre Linux	22
2.1.5.	Direcciones de Linux y educación.	23
<b>3.</b>	<b>Instalación</b>	<b>24</b>
3.1.	Antes de la instalación.	24
3.2.	Métodos de instalación.	25
3.3.	Instalando Linux.	26
3.3.1.	Fips.	26
3.3.2.	Comencemos. Instalando Linux.	28
3.3.3.	Instalación de Linux desde un CD.	29
3.3.4.	Lenguaje de instalación, teclado y ratón.	31
3.3.5.	Selección del tipo de instalación.	33
3.3.6.	DiskDruid (Fdisk): crear las particiones.	34
3.3.7.	Configuración de GRUB.	39
3.3.8.	Red	39
3.3.9.	Selección de más idiomas y de la Zona Horaria	40
3.3.10.	Gestión de usuarios y configuración de la autenticación.	41
3.3.11.	Seleccionar paquetes.	41
3.3.12.	Instalación de los paquetes, disco de arranque.	42
3.3.13.	Configuración del entorno gráfico	42
3.3.14.	Configuración del monitor	43
3.4.	Comenzamos a trabajar con LINUX	44
3.4.1.	Arranque en modo gráfico	44
3.4.2.	Arranque en modo texto	45
3.4.3.	Apagado y reinicio del sistema	45

<b>4. Conexión a Internet: RTB, ADSL, RDSI.</b>	<b>47</b>
4.1. Introducción.	47
4.2. RTB	48
4.2.1. Configuración del módem.	49
4.2.2. Configuración del interfaz ppp	49
4.2.3. Pistas para detectar problemas	52
4.2.4. Conexión mediante Script	55
4.3. RDSI	57
4.4. ADSL	58
<b>5. Información en línea</b>	<b>62</b>
5.1. Introducción	62
5.2. manpages	63
5.3. info-pages	66
5.4. /usr/share/doc (/usr/doc)	67
<b>6. Apéndice: Midnight Commander</b>	<b>68</b>
6.1. Introducción	68
6.1.1. Instalación	68
6.1.2. Inicio de una sesión	69
6.1.3. Soporte de Ratón	69
6.1.4. Teclas	70
6.2. Barra de Menú	70
6.2.1. Menús Izquierdo y Derecho	71
6.2.2. Menú de Fichero	72
6.2.3. Menú de Comando	77
6.2.4. Menú de Opciones	79
6.3. Ejecutando Comandos del Sistema Operativo	83
6.3.1. El Comando cd Interno	83
6.4. Sistema de Ficheros Virtual	83
6.4.1. Sistema de Ficheros FTP	84
6.4.2. Sistema de Archivos Tar	84
6.4.3. Sistema de Ficheros de Red	84
6.4.4. Sistema de Archivos de Recuperación	85
6.4.5. ➡ Para practicar	85
<b>II Linux en modo gráfico. Configuración</b>	<b>87</b>
<b>7. Configuración de las X window. Gestores de Ventanas.</b>	<b>89</b>
7.1. ¿Qué es un gestor de ventanas?	89
7.1.1. Introducción:	89
7.1.2. ¿Es suficiente con los gestores de ventanas?: GNOME y KDE	93
7.2. Utilidad redhat-config-xfree86	93
7.2.1. ¿Qué hemos hecho?: fichero /etc/X11/XF86Config.	97
7.3. Otros gestores de ventanas	103
7.3.1. Utilidad switchdesk.	103
7.4. Varias sesiones abiertas	106
7.5. Ideas para configurar WindowMaker e Icewm	109
7.5.1. WindowMaker	110
7.5.2. Icewm	110
7.6. Fichero \$HOME/.Xresources	111
7.6.1. Modo texto	111
7.6.2. Modo gráfico	111



7.6.3.	\$HOME/.Xresources . . . . .	112
7.6.4.	Fuentes en Linux . . . . .	115
<b>8.</b>	<b>Moviéndose por Linux: GNOME (y KDE)</b>	<b>118</b>
8.1.	GNOME . . . . .	119
8.1.1.	El escritorio . . . . .	119
8.1.2.	Nautilus . . . . .	119
8.1.3.	El panel . . . . .	121
8.1.4.	Personalizar GNOME . . . . .	127
<b>9.</b>	<b>Configuración: Impresora, Sonido, etc.</b>	<b>128</b>
9.1.	Configuración personalizada: impresora, tarjeta de sonido. . . . .	128
9.2.	Impresoras . . . . .	128
9.2.1.	/etc/printcap . . . . .	133
9.2.2.	Órdenes básicas de impresión. . . . .	134
9.3.	Tarjeta de sonido . . . . .	135
9.3.1.	Más sonido . . . . .	139
9.3.2.	Más periféricos . . . . .	139
<b>10.</b>	<b>Paquete Ofimático: OpenOffice</b>	<b>142</b>
10.1.	Introducción . . . . .	142
10.2.	Características de OpenOffice 1.0.1 . . . . .	143
10.3.	El programa . . . . .	144
10.3.1.	Inicio del programa . . . . .	144
10.3.2.	Write . . . . .	144
10.3.3.	Calc . . . . .	145
10.3.4.	Impress . . . . .	145
10.3.5.	Draw . . . . .	146
<b>III</b>	<b>Primeros pasos con Linux</b>	<b>147</b>
<b>11.</b>	<b>Inicio del Sistema</b>	<b>148</b>
11.1.	Gestores de arranque . . . . .	148
11.1.1.	Grub . . . . .	148
11.1.2.	Interfaces de GRUB . . . . .	149
11.1.3.	Fichero /boot/grub.conf . . . . .	151
11.1.4.	Por si se opta por LILO. . . . .	154
11.2.	Fichero /etc/inittab . . . . .	157
11.2.1.	Niveles de arranque . . . . .	160
11.2.2.	Control de acceso a servicios . . . . .	162
11.2.3.	Distintos programas para arrancar en modo gráfico . . . . .	163
<b>12.</b>	<b>Sistemas de ficheros en Linux</b>	<b>166</b>
12.1.	Introducción. . . . .	166
12.2.	Tipos de Sistemas de Ficheros. . . . .	167
12.3.	Organización de los directorios . . . . .	168
12.3.1.	Ficheros de configuración del sistema . . . . .	168
12.3.2.	Logs del sistema . . . . .	171
12.4.	Creación de un Sistema de Ficheros. . . . .	172
12.5.	Montaje y Desmontaje. . . . .	173
12.5.1.	El fichero /etc/fstab . . . . .	174
12.5.2.	El comando mount . . . . .	175
12.5.3.	El comando umount . . . . .	175



12.5.4. Herramientas gráficas para montar dispositivos . . . . .	175
12.5.5. ➡ Para practicar: Montar una partición ntfs . . . . .	176
12.5.6. ➡ Para practicar: Paquete mtools . . . . .	177
12.6. Chequeo y recuperación: fsck. . . . .	178
12.7. Enlaces . . . . .	179
<b>13. Permisos. Gestión de Usuarios</b> . . . . .	<b>180</b>
13.1. Introducción. . . . .	180
13.2. Permisos de Acceso a los distintos Objetos. . . . .	180
13.2.1. chmod . . . . .	181
13.2.2. Más fácil todavía . . . . .	182
13.2.3. Más sobre permisos . . . . .	184
13.3. Gestión de usuarios en modo texto. . . . .	186
13.4. Gestión de usuarios en modo gráfico. . . . .	187
13.5. ➡ Para practicar . . . . .	188
13.5.1. SOLUCIÓN . . . . .	189
<b>14. Instalación/desinstalación de paquetes.</b> . . . .	<b>193</b>
14.1. Introducción . . . . .	193
14.2. ¿Qué es esto de rpm? . . . . .	194
14.3. Comando rpm . . . . .	195
14.3.1. Instalación de paquetes . . . . .	196
14.3.2. Actualización de paquetes . . . . .	199
14.3.3. Desinstalación de paquetes . . . . .	201
14.3.4. Información sobre los paquetes. . . . .	201
14.3.5. Verificación de los paquetes. . . . .	202
14.4. ➡ Para practicar: Paquetes fuentes. . . . .	203
14.5. ➡ Para practicar: redhat-config-packages . . . . .	204
14.5.1. Instalación del KDE . . . . .	204
<b>IV La Shell Bash</b> . . . . .	<b>206</b>
<b>15. La Shell Bash I</b> . . . . .	<b>207</b>
15.1. La Shell Bash . . . . .	207
15.1.1. ¿Qué es una shell? . . . . .	207
15.1.2. Características básicas de la Shell. . . . .	208
15.1.3. Variables de entorno de la Bash . . . . .	208
15.1.4. Ficheros de inicio de la bash . . . . .	209
15.1.5. Personalizando el Prompt . . . . .	210
15.1.6. Los Alias . . . . .	211
15.1.7. Historia de órdenes. . . . .	212
15.1.8. Los Builtins (Órdenes internas) . . . . .	212
15.2. Redirección . . . . .	213
15.2.1. Redirección de la entrada (<) . . . . .	213
15.2.2. Redirección de la salida (>) . . . . .	213
15.2.3. Añadir a la salida redirigida (>>) . . . . .	214
15.3. Comandos de la Shell I . . . . .	215
15.3.1. Comandos simples . . . . .	215
15.3.2. Tuberías . . . . .	215
15.3.3. Listas de comandos . . . . .	215

<b>16. Comandos básicos de Unix/Linux</b>	<b>217</b>
16.1. Introducción . . . . .	217
16.1.1. Convenciones en cuanto a la sintaxis . . . . .	218
16.1.2. Comodines . . . . .	219
16.2. Resumen comandos . . . . .	219
16.2.1. Ayuda . . . . .	219
16.2.2. “Construir” comandos . . . . .	219
16.2.3. Gestión de usuarios y grupos . . . . .	219
16.2.4. Manipulación de archivos y directorios . . . . .	220
16.2.5. Localización de archivos . . . . .	220
16.2.6. Procesamiento de archivos . . . . .	220
16.2.7. Guardar y comprimir ficheros . . . . .	221
16.2.8. Procesos de control . . . . .	221
16.2.9. Control de usuarios . . . . .	221
16.2.10. Administrar ficheros . . . . .	221
16.2.11. Comunicaciones y redes . . . . .	221
16.2.12. Comandos de Impresión . . . . .	222
16.2.13. Módulos del kernel . . . . .	222
16.2.14. Varios . . . . .	222
16.3. Algunos ejemplos . . . . .	222
16.3.1. “Construir” comandos . . . . .	222
16.3.2. Manipulación de archivos y directorios . . . . .	223
16.3.3. Localización de archivos . . . . .	225
16.3.4. Procesamiento de archivos . . . . .	227
16.3.5. Empaquetando y comprimiendo ficheros. . . . .	228
16.3.6. Procesos de control . . . . .	233
16.3.7. Administrar ficheros . . . . .	237
16.3.8. Comunicaciones y redes. . . . .	239
16.3.9. Varios . . . . .	242
<b>17. La Shell Bash II</b>	<b>243</b>
17.1. Control de trabajos . . . . .	243
17.2. Expresiones . . . . .	245
17.2.1. Expresiones condicionales . . . . .	245
17.3. Comandos de la Shell II . . . . .	247
17.3.1. Bucles . . . . .	247
17.3.2. Sentencias condicionales . . . . .	248
17.4. Ficheros de órdenes (Shell Scripts) . . . . .	248
17.4.1. Argumentos para los Procedimientos Shell . . . . .	249
17.4.2. Entrada/Salida de Datos . . . . .	250
17.4.3. Bucles y condiciones . . . . .	251
17.5. Ejemplos de scripts. . . . .	251
17.5.1. Conversión a minúsculas de los nombres de archivos. . . . .	251
17.5.2. Pasar varios argumentos . . . . .	253
17.5.3. Script para compilar un fichero $\text{\LaTeX}$ . . . . .	254
17.5.4. Texto en movimiento por la pantalla. . . . .	254
17.5.5. Adivinar el número de palabras de un fichero. . . . .	255
17.5.6. Conectar a Internet con varios servidores de acceso . . . . .	256

<b>V</b>	<b>Programas para Linux</b>	<b>259</b>
<b>18.</b>	<b>Instalación de programas</b>	<b>261</b>
18.1.	Instalación a partir del código fuente.	261
18.1.1.	Comando tar	261
18.2.	Instalación a partir de binarios	264
18.3.	En formato RPM	265
<b>19.</b>	<b>Gráficos</b>	<b>267</b>
19.1.	xv	267
19.2.	The Gimp	270
19.3.	Image Magick	277
19.4.	xfig	278
19.5.	Qcad	280
19.6.	Blender	281
19.7.	Visor gráfico GQview	282
<b>20.</b>	<b>Internet</b>	<b>284</b>
20.1.	Mozilla	284
20.2.	Editores HTML	286
20.2.1.	Bluefish	286
20.2.2.	Quanta	286
20.3.	Varios	287
20.3.1.	FTP: gftp	287
20.3.2.	wget	288
<b>21.</b>	<b>Emuladores</b>	<b>289</b>
21.1.	Wine	289
21.2.	DosEmu	291
<b>22.</b>	<b>Textos</b>	<b>293</b>
22.1.	WordPerfect 8	293
22.1.1.	Instalación	294
22.1.2.	Inicio de WordPerfect. Selección de la Impresora.	299
22.2.	Otros	302
22.3.	Abiword	302
22.4.	Vi no, gvim	303
<b>23.</b>	<b>Programas para L<sup>A</sup>T<sub>E</sub>X</b>	<b>305</b>
23.1.	L <sup>A</sup> T <sub>E</sub> X	305
23.2.	Ficheros en formato pdf/ps.	308
23.2.1.	Formato PostScript	308
23.2.2.	Formato pdf	309
23.3.	LyX	311
23.3.1.	Instalación de LyX	312
23.3.2.	Algunas cuestiones sobre LyX	315
23.3.3.	Exportar a HTML	317
23.4.	Otros	319
23.4.1.	Kile	319
23.4.2.	TexMacs	319
23.4.3.	Ampliación: Epix	320

<b>24. Matemáticas</b>	<b>324</b>
24.1. Gráficos	325
24.1.1. geg	325
24.1.2. Gnuplot	325
24.2. Estadística	328
24.2.1. R	328
24.2.2. Grace	333
24.3. Álgebra simbólica: MuPAD	336
24.4. scilab	341
24.5. Geometría	342
24.5.1. DrGenius	342
24.5.2. Geomview	344
<b>25. Astronomía</b>	<b>347</b>
25.1. xephem	347
25.2. xplns	348
 <b>VI Redes en Linux</b>	 <b>350</b>
<b>26. Introducción a los protocolos TCP/IP.</b>	<b>351</b>
26.1. Introducción	351
26.2. Protocolos de Red	352
26.3. Introducción a las direcciones IP.	353
<b>27. Construir una red con direccionamiento de red privado.</b>	<b>356</b>
27.1. Introducción	356
27.2. Creación de las interfaces de red.	358
27.3. Configurando el encaminamiento (routing).	361
27.4. Configuración del sistema DNS	363
27.5. Configuración: servidores y servicios de red.	366
27.5.1. El servicio xinetd	369
27.5.2. El mecanismo de control de acceso tcpd (tcp_wrappers)	371
27.5.3. Control de acceso a servicios	373
27.6. Servidor Web (apache).	375
27.6.1. Servidores Seguros	381
27.6.2. Páginas PHP	382
27.7. Servicio Proxy-cache. Squid	384
<b>28. Más servicios de red</b>	<b>387</b>
28.1. Servidor de ftp.	387
28.2. Correo electrónico.	390
28.2.1. Una configuración práctica.	390
28.2.2. Poner todo en marcha.	395
28.2.3. Montemos un servidor de correo con nuestro propio dominio	395
28.3. Otros servicios	398
28.3.1. Telnet	398
28.3.2. Ssh	399
28.4. Linux como enrutador	400
28.4.1. iptables	401
28.5. Sistema de ficheros NFS	403
28.6. Compartir impresoras entre máquinas Linux	405

<b>29. Samba</b>	<b>407</b>
29.1. ¿Qué es Samba?	407
29.2. Instalación	409
29.3. Configuración de la máquina Windows 9x	410
29.4. Configuración de la máquina Linux	412
29.4.1. Análisis del archivo de configuración de Samba	412
29.4.2. Swat	417
29.5. A “bailar” la Samba	418
29.5.1. Acceder desde una máquina Linux a una Windows	418
29.5.2. Acceder desde Windows a la máquina Linux	422
<b>30. Manteniendo nuestro sistema seguro</b>	<b>425</b>
30.1. Modelos de seguridad.	425
30.2. Amenazas a la seguridad de nuestro servidor.	425
<b>31. Apéndice.</b>	<b>429</b>
31.1. PHP con MySQL: instalación de phpnuke	429
 <b>VII Compilación del núcleo</b>	 <b>432</b>
<b>32. Introducción a C.</b>	<b>433</b>
32.1. Lenguajes de Programación	433
32.2. Introducción a C.	434
32.2.1. Librerías en C.	436
32.2.2. Trabajando con proyectos. Makefile.	439
<b>33. Compilación del kernel</b>	<b>443</b>
33.1. Subiendo de versión nuestro kernel mediante un RPM	444
33.2. Construyendo un kernel a medida.	445
<b>A. Licencia de Documentación Libre GNU (traducción)</b>	<b>452</b>
A.1. GFDL	452

## **Parte I**

# **Linux no es ventanas**

# Capítulo 1

## ¿Qué es Linux? Posibilidades de Linux. Estructura del Sistema Operativo

Aprenderás qué es Linux, sus virtudes y sus defectos, cómo puedes conseguirlo, qué distribución elegir a la hora de instalarlo, cómo administrar tu sistema correctamente, cómo trabajar con X-Window, cómo conectarte con él a Internet...Pero no todo va a ser tan fácil con Linux, porque como comprobarás, no es un sistema hecho para cobardes. Tendrás que ser valiente (*Manual Avanzado de Linux* de RAÚL MONTERO RIVERO, Ed. Anaya)

Aclaremos en primer lugar qué es el Software Libre, ...

\* "Software Libre" se refiere a la libertad de los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software.

\* [1ª libertad ] La libertad de ejecutar el programa, con cualquier propósito.

\* [2ª libertad ] La libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades. (El acceso al código fuente es una precondition para esto)

\* [3ª libertad ] La libertad de distribuir copias de manera que se puede compartir con los demás.

\* [4ª libertad ] La libertad de mejorar el programa, y liberar las mejoras al público de tal manera que toda la comunidad se beneficia de la colaboración. (El acceso al código fuente es una precondition para esto.) (*El "rigor" de los que hablan de Linux*, HISPALINUX 2003)<sup>1</sup>

### 1.1. Introducción.

Esta es la primera entrega del curso de Linux y, como es obligado, hay que dar una visión inicial de ¿qué es Linux? y ¿cómo surgió?. Las respuestas a estas preguntas nos pueden hacer entender cuáles son las grandes ventajas que posee este sistema operativo, así como alguna de sus limitaciones.

En esta primera entrega daremos también información básica para sumergirnos en el fascinante mundo de Linux. Una de las dificultades aparentes de Linux es la falta de documentación. Como veremos a lo largo de esta entrega en la actualidad esto no se corresponde con la realidad. Hay material suficiente a nuestra disposición, en castellano, para documentar todos los aspectos de Linux.

Linux es un sistema operativo dinámico, en continua evolución y del que siempre hay que cosas que aprender. Linux no es Windows. Si lo único que esperamos de Linux es un sistema Windows gratuito posiblemente que la decepción no tarde en aparecer. Con Linux, como con el buen vino, hay que tener tiempo

---

<sup>1</sup>Se puede acceder al artículo completo en: <http://www.hispalinux.es/gabinete-prensa/respuesta-astic-1/>



y paciencia: cuanto más se paladea, más se disfruta con él. Esperamos que con este curso comencemos a “paladear” Linux y que aprendamos y disfrutemos con él.

## 1.2. Una breve historia de Linux

**Linux** es una implementación **gratuita y de libre distribución** de Unix, y, por tanto, su origen está ligado al inicio de Unix en 1969<sup>2</sup>. Si bien ese es su origen, su nacimiento es bastante posterior, hay que esperar más de 20 años para que esto ocurra.

El nacimiento de Linux hay que situarlo a principios de la década de los 90, cuando un estudiante de informática empieza a trabajar sobre una variante educativa de UNIX llamada Minix, con la idea de crear un nuevo núcleo de UNIX basándose en Minix (de hecho el sistema de archivos es muy similar pero más estable y libremente accesible) pero con una filosofía diferente<sup>3</sup>, la idea era que su sistema estuviese encuadrado dentro del proyecto GNU<sup>4</sup>.

¿Y quién es ese estudiante?, su nombre es LINUS BENEDICT TORVALDS, nació en Helsinki en 1969, él es el padre de la “criatura” (conserva los derechos de autor del núcleo básico). El 5 de Octubre de 1991 LINUS dio a conocer la primera versión oficial: la 0.02 (la 0.01 no la dio a conocer al público), con ella podía ejecutar bash (el shell<sup>5</sup> de GNU) y gcc (el compilador de C de GNU). Para dar a conocer esta primera versión, Linus puso en un grupo de noticias el siguiente mensaje:

¿Suspiráis al recordar aquellos días de Minix-1.1, cuando los hombres eran hombres y escribían sus propios drivers? ¿Os sentís sin ningún proyecto interesante y os gustaría tener un verdadero S.O. que pudierais modificar a placer?

¿Os resulta frustrante el tener solo a Minix? Entonces, este artículo es para vosotros.

Como dije hace un mes, estoy trabajando en una versión gratuita de algo parecido a Minix para ordenadores At-386. He alcanzado la etapa en la que puede ser utilizable y voy a poner las fuentes para su distribución. Es sólo la versión 0.02. . . pero he conseguido ejecutar en él bash, gcc, gnu-make, gnu-sed, compress, etc. (*Linux Instalación y Primeros Pasos*, de MATT WELSH).

Tras esta versión y con el apoyo de un grupo de voluntarios con acceso a Internet se empiezan a producir las mejoras, de forma continuada hasta hoy, de ese proyecto inicial:

- A principios de 1992 Linus añadió Linux al proyecto GNU.
- En abril de 1992 aparece la primera versión de Linux capaz de ejecutar el entorno gráfico X-window. Es la versión 0.96.
- El 16 de abril de 1994 aparece la primera versión “completa” de Linux, la 1.0.
- En Diciembre de 1996 aparece la revisión 2.0 de Linux y se presenta en sociedad la mascota oficial de Linux: el pingüino Tux.



<sup>2</sup>Unix fue desarrollado por KEN THOMPSON en 1969 en los laboratorios AT&T.

<sup>3</sup>El creador de Minix (ANDY TANNENBAUM) cedió todos los derechos sobre Minix a una empresa que comenzó a cobrar 150\$ por licencia.

<sup>4</sup>El proyecto GNU de la Fundación de Software Libre en Cambridge ya estaba en funcionamiento desde 1983

<sup>5</sup>Es el programa intermediario entre el usuario y el núcleo. Si lo comparamos con el MSDOS, un shell de Unix equivaldría al intérprete de comandos COMMAND.COM (realmente es más que eso, un shell además es un lenguaje de programación)

- En la actualidad el núcleo<sup>6</sup> va por la versión 2.4.20 estable<sup>7</sup>.

Hoy se calcula que de unos 500 millones de equipos existentes en el mundo, entre 15 y 25 millones trabajan con Linux.

Antes de continuar es conveniente aclarar qué significado tiene el sistema de numeración de las revisiones del kernel. Por convenio, las versiones del núcleo siguen un sistema de numeración basado en tres números separados por puntos  $x.y.z$ , en las que cada número tiene un significado diferente:

- El primer número,  $x$ , es el número principal, el incremento de  $x$  indica que se ha producido un cambio sustancial en las posibilidades de ese núcleo. Desde 1996  $x$  toma el valor 2.
- El segundo número,  $y$ , tiene un doble significado, su incremento indica mejoras en el núcleo pero manteniendo el criterio de que si  $y$  es par se trata de versiones estables, si  $y$  es impar son versiones de desarrollo<sup>8</sup>.
- El tercer número indica la revisión de la versión actual, el incremento de  $z$  indica que se han corregido errores de la versión anterior pero sin características nuevas.
- También se utiliza otra nomenclatura consistente en añadir la “coletilla” *preX*, en donde  $X$  es un número que generalmente oscila entre 1 y 16. Por ejemplo el número 2.4.1-pre12 significa que el núcleo en cuestión es posterior al 2.4.1 pero casi listo para ser el 2.4.2.

### 1.3. Posibilidades de Linux

Como ya hemos comentado, el núcleo es el verdadero corazón del sistema, ya que mediante él podemos controlar el hardware de nuestro ordenador. El núcleo de Linux está disponible en código fuente y, por tanto, es susceptible de ser modificado por cualquier programador si lo ve necesario. Además, la mayoría de las aplicaciones existentes para Linux comparten esta filosofía.

Las características más relevantes de Linux son<sup>9</sup>:

- Multitarea: posibilidad de ejecutar varios programas (procesos) a la vez sin tener que detener una aplicación para ejecutar otra.
- Multiusuario: varios usuarios pueden acceder a las aplicaciones o recursos en el mismo PC al mismo tiempo (¡y **sin** licencias para todos!).
- Multiplataforma: corre en muchas CPUs distintas (Intel 386/486/Pentium y compatibles como K6/7 de AMD, procesadores de la familia Motorola 680x0, Sun Sparc, etc).
- Tiene Shell programables, lo que hace que sea el sistema operativo más flexible que existe.
- Independencia de los dispositivos, permite que se pueda conectar cualquier número y tipo de dispositivos mediante un enlace individual al núcleo.<sup>10</sup>
- Linux es el sistema operativo (junto con Unix) con mayor número de funciones de conexión a red diferentes.
- Soporta varios sistemas de ficheros.

<sup>6</sup>El núcleo (kernel) de Linux es el encargado de que el software y el hardware del ordenador trabajen conjuntamente

<sup>7</sup>La versión 2.4.0 se presentó el 04/01/2001. La “última” versión vio la luz el 28/11/2002. Para comprobar si esta información está ya anticuada:

<http://www.kernel.org>

<sup>8</sup>Versiones de evaluación en espera de ser totalmente depuradas

<sup>9</sup>Para ampliar este tema se puede consultar Linux Documentation Project <http://www.tldp.org/HOWTO/INFO-SHEET.html> un documento mantenido por MICHAEL K. JHONSON en donde hay información sobre las características, requerimientos y recursos sobre Linux. La última actualización de este documento es la 4.14.1 del 1 de septiembre de 1998.

Una explicación más detallada es posible encontrarla en el libro *Linux Instalación y Primeros Pasos*, de MATT WELSH en el apartado 1.3 Características del sistema.

<sup>10</sup>Aquí radica aún uno de los problemas de Linux

- Ejecuta las aplicaciones según el modelo de memoria virtual, es decir, un programa se puede ejecutar sin que sea necesario que esté cargado en su totalidad en la memoria del ordenador.

## 1.4. Distribuciones de Linux

Linux se puede dividir en cuatro componentes:

1. El núcleo.
2. El shell
3. El sistema de archivos.
4. Programas básicos con los que trabajar.

Al conjunto formado por estos cuatro componentes es a lo que se llama distribución. Es decir, al núcleo junto con las aplicaciones y utilidades necesarias para realizar nuestro trabajo. En la actualidad hay más de treinta<sup>11</sup>. Sólo vamos a enumerar las “más importantes” ya que muchas de las existentes se basan en alguna de las aquí listadas:

- **Redhat**

Web: <http://www.redhat.com>

FTP: <ftp://ftp.redhat.com/pub/>

- **Debian**

Web: <http://www.debian.org/>

FTP: <ftp://ftp.debian.org/debian/>

- **SuSE**

Web: <http://www.suse.de/es/>

FTP: <ftp://ftp.suse.com/>

- **Caldera Openlinux**

Web: <http://www.caldera.com/>

FTP: <ftp://ftp.caldera.com/pub/OpenLinux/>

- **Slackware**

Web: <http://www.slackware.com/>

FTP: <ftp://ftp.slackware.com>

- **Mklinux**

Web: <http://www.mklinux.org>

FTP: <ftp://ftp.mklinux.org/pub/>

- **Mandrake**

Web: <http://www.linux-mandrake.com/es/>

FTP: <ftp://ftp.rediris.es/pub/linux/distributions/mandrake/>

- **Conectiva**

Web: <http://www.conectiva.com/>

FTP: <ftp://ftp.conectiva.com/pub/conectiva/>

---

<sup>11</sup> Hay más información sobre las distribuciones de Linux en Linux Distribution HOWTO <http://www.linuxdoc.org/HOWTO/Distributions-HOWTO/index.html>.

En España hay en la actualidad tres distribuciones, son:

- **Hispafuentes** (basada en RedHat)

Web: <http://www.hispafuentes.com>

y dos en Debian, de las cuales la segunda merece mención especial:

- **Esware**

Web: <http://www.esware.com/>

- **Linex** la distribución Linux de la Junta de Extremadura

Web: <http://www.linex.org/>

En este curso vamos a trabajar con la distribución más extendida en la actualidad, la distribución Red Hat (en su versión 8.0).



Entre sus características principales<sup>12</sup> se puede destacar:

- Todo su código se distribuye bajo licencia GNU.
- Núcleo 2.4.18-14 preparado para trabajar con redes TCP/IP y una gran variedad de hardware<sup>13</sup>.
- Servicios de Internet como redes SLIP/PPP, servidor web Apache, correo electrónico, noticias y ftp.
- Software Samba LAN Manager para trabajar con sistemas Windows en red.
- XFree86 4.2.0 con posibilidad de elegir entre varios gestores de ventanas o escritorios gráficos de trabajo.
- Compiladores e intérpretes para casi todos los lenguajes de programación.
- Una gran cantidad de aplicaciones y utilidades, tanto en modo texto como en modo gráfico.
- Procesadores de textos y herramientas de escritura.
- Permite la instalación en modo gráfico y en castellano (con ayuda contextual en castellano pulsando F1).

## 1.5. Estructura del sistema operativo.

Sin el software la computadora es sólo una montaña de componentes electrónicos que disipa calor. Si el hardware es el corazón de una computadora, el software es su alma. Un sistema operativo es una colección de programas del sistema que permiten al usuario ejecutar aplicaciones. El sistema operativo hace abstracción del hardware del sistema y presenta a los usuarios del sistema y a sus aplicaciones una máquina virtual (mucho más fácil de manejar que la máquina real, que sólo entiende de ceros y unos). En un sentido muy auténtico, el software da el carácter del sistema. La mayor parte de los PCs pueden ejecutar uno o varios sistemas operativos y cada uno puede tener una apariencia y comportamiento muy diferentes. Linux está hecho de varias piezas funcionales diferentes que, combinadas, forman el sistema operativo. Una parte obvia de Linux es el núcleo en sí, pero incluso éste sería inútil sin bibliotecas (librerías) o intérpretes de comandos.

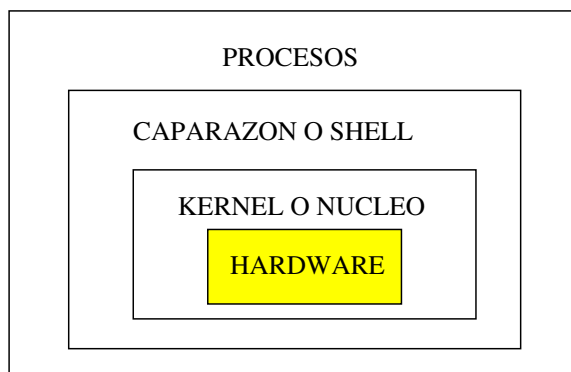
<sup>12</sup>La primera no es común al resto de distribuciones.

<sup>13</sup>Por ejemplo dispositivos USB.

**Estructura general del sistema operativo UNIX** Se puede dividir en varios componentes perfectamente diferenciados:

- Núcleo o Kernel: Comprende un 5-10 % del código total.
- Caparazón o Shell: Actúa como intérprete de comandos.
- Programas de utilidad.

Los diversos componentes del sistema operativo pueden verse de forma gráfica en la figura



**Kernel** Es el núcleo del S.O. UNIX. Es el encargado de controlar el sistema. Tiene diversas tareas asignadas:

- Planificar, coordinar y gestionar la ejecución de los procesos. Para ello, hace uso de las prioridades asignadas a cada proceso y utiliza algoritmos específicos para repartir el tiempo entre los diversos procesos que compiten por él.
- Dar servicios del sistema, como entrada/salida y gestión de ficheros.
- Manejar las operaciones dependientes de hardware, es decir, realiza las funciones de más bajo nivel de manera que se oculten al usuario.

Las fuentes del último kernel tienen un tamaño de más de 30 MB de código<sup>14</sup> de los cuales un 70-80 % está escrito en C y el resto depende del procesador. Tras compilarlo ocupa casi 1 MB en un PC y para máquinas grandes puede llegar a 2 Mb.<sup>15</sup>

**Shell** Desde el punto de vista del usuario, actúa como un intérprete de comandos. Es un programa que siempre está en ejecución.

El Shell lee las órdenes suministradas, las descodifica y lo comunica al núcleo para realizar la acción especificada.

Prácticamente, todas las órdenes son programas ejecutables que el shell busca en el sistema de ficheros, siguiendo el orden especificado en la variable global PATH.

Existen varios tipos de shells en función, principalmente, de la versión de UNIX utilizada:

- Bourne shell (System V, Xenix)
- C shell (Berkeley)
- Korn shell (Ambos)

El shell puede constar, en total, de unas 200.000 líneas de código en C.

<sup>14</sup>Para instalar las fuentes y documentación del último núcleo de RedHat necesitamos más de 115 MB de espacio libre en nuestro disco duro.

<sup>15</sup>Este archivo recibe el nombre de `vmlinuz` y se encuentra en el directorio `/boot`.



**Procesos (o Programas de Utilidad)** Son los programas que realmente nos facilitan la vida, como procesadores de texto, programas de dibujo o programas matemáticos. Se invocan por medio de un intérprete de comandos (shell).

## Capítulo 2

# Linux e Internet: fuentes de información y programas.

LINUX es un producto de Internet y casi toda la información al respecto se encuentra disponible en la Web. Sin embargo, la Web es enorme. Incluso con motores de búsqueda como Yahoo! y Alta Vista, puede resultar difícil localizar lo que se está buscando. Además, la Web está continuamente en proceso de cambio. La gente cambia de afición como de ISP. Los proveedores de servicio quiebran, etc. Como resultado de ello, el mejor URL, hoy por hoy, es el “Error 404” del mañana. (*Adiministración de Sistemas Linux*, CARLING, M, STEPHEN DEGLER y JAMES DENNIS, Prentice Hall)

### 2.1. ¿Dónde encuentro información?

Linux es un sistema operativo en constante evolución, y la rapidez de esta evolución está asociada al fenómeno Internet. La mayoría de fuentes de información sobre aspectos concretos de Linux están a nuestra disposición en la red. Aparte de los sitios Web de las distintas distribuciones, hay multitud de páginas que nos permiten acceder a programas y manuales sobre casi cualquier tema del que tengamos dudas<sup>1</sup>.

La lista es tan amplia que de hecho hay páginas de usuarios de Linux tratando sólo este tema. En este apartado vamos a intentar exponer, aún a riesgo de “olvidar” algunas de las más importantes, aquellas que pensamos que pueden ser más útiles para el desarrollo del curso.

Merecen mención especial dos páginas de las cuales la primera es casi de obligada visita:

- Red Hat Online Documentation <http://www.europe.redhat.com/documentation/>. Aquí podéis encontrar en formato HTML las guías (las 4 primeras en castellano) que acompañan a la versión “oficial” de RedHat hasta la versión 8.0:
  - *The Official Red Hat Linux x86 Installation Guide*
  - *The Official Red Hat Linux Getting Started Guide*
  - *The Official Red Hat Linux Customization Guide*
  - *The Official Red Hat Linux Reference Guide*
  - *The Official Red Hat Linux Security Guide*
- Si preferimos el formato pdf (en inglés): Red Hat Linux Manuals <http://www.redhat.com/docs/manuals/linux/>
- Imprescindible, toda la documentación sobre Linux: The Linux Documentation Project <http://www.tldp.org> y la traducción al castellano de LuCAS <http://es.tldp.org/>

---

<sup>1</sup>Además, casi todas las utilidades Linux tienen sus propias fuentes de información que se instalan a la vez que el programa (man pages, info-pages, documentación). Véase el Capítulo 5 en la página 62



### 2.1.1. Documentos y manuales

La información existente se ha dividido en tres categorías:

#### Linux HOWTO y mini-HOWTO

Son documentos cortos que explican de forma práctica cómo solucionar algún aspecto concreto sobre Linux. La traducción de estos documentos la están realizando los miembros del grupo INSFLUG, la dirección de su página web es <http://www.insflug.org> y podemos acceder a todos vía ftp en <ftp://ftp.insflug.org/es/>. Algunos de los COMOS traducidos:

- |                                |                                     |                               |
|--------------------------------|-------------------------------------|-------------------------------|
| ■ Abogacia-Por-Linux-Como      | ■ Ifmail-Como                       | ■ Propiedad-Intelectual-Como  |
| ■ Accesibilidad-Como           | ■ InfoSheet-Como                    | ■ QuakeI-II-Como              |
| ■ Bash-Prompt-Como             | ■ Infobia-Como                      | ■ RDSI-Como                   |
| ■ Benchmarking-Como            | ■ Inn-Suck-Como                     | ■ RPM-Como                    |
| ■ BitchX-Como                  | ■ Kernel-Como                       | ■ Redes-En-Linux-Como         |
| ■ CDRom-Como                   | ■ LDAP-Linux-Como                   | ■ Reproduccion-De-Sonido-Como |
| ■ Coda-Como                    | ■ La_empresa_ante-el_software_libre | ■ Samba-Como                  |
| ■ conceptos-de-redes-COMO      | ■ Linux-Tips-Como                   | ■ Serie-Como                  |
| ■ Configuracion-Impresion-Como | ■ Linuxdoc-Como                     | ■ Servidor-IRC-Como           |
| ■ Cortafuegos-Como             | ■ MD5                               | ■ Software-RAID-Como          |
| ■ DNS-Como                     | ■ MP3-Como                          | ■ Sonido-Como                 |
| ■ Diald-Como                   | ■ MetaFAQ-Como                      | ■ Spanish-Como                |
| ■ Dos-Win-Linux-Como           | ■ Mutt-GnuPG-PGP-Como               | ■ Term-Como                   |
| ■ Dos-a-Linux-Como             | ■ NAT-Como                          | ■ Terminales-Como             |
| ■ Dosemu-Como                  | ■ Nis-Como                          | ■ UUCP-Como                   |
| ■ FEddi-Como                   | ■ Noticias-Como                     | ■ Umsdos-Como                 |
| ■ Ftp-Anonimo-Como             | ■ PCMCIA-Como                       | ■ Uso-Impresion-Como          |
| ■ Fuente-ISO-Como              | ■ PPP-Como                          | ■ WWW-Como                    |
| ■ Grabadoras-Como              | ■ Pilot-Como                        | ■ XFree86-Como                |
| ■ IP-Masquerade-Como           | ■ Programacion-Serie-Como           |                               |

Con respecto a los mini-HOWTO, tanto en la página antes comentada como en LuCAS <http://es.tldp.org> hay un apartado con el listado de los que hay disponibles.

#### Listas de FAQ

Las **FAQ**<sup>2</sup> (*Frequently Asked Questions*) son documentos que recogen las respuestas dadas a las preguntas que se realizan con más frecuencia sobre problemas con Linux (y en Internet en general). En estos documentos pueden aparecer respuestas a cuestiones más generales que en los anteriores.

Una FAQ muy buena en español es:

<sup>2</sup>Castellanizando, las hemos visto como PUF (Preguntas de Uso Frecuente) o PRF (Preguntas Realizadas Frecuentemente)





- *FAQ sobre Linux para principiantes* - es.comp.os.linux

Disponible en es.comp.os.linux.\* <http://www.escomposlinux.org/Faq/> y en El rincón de Linux <http://www.linux-es.com>. En esta última Web hay disponibles más FAQ sobre otros temas.

### Libros del LDP

En LuCAS <http://es.tldp.org> podemos encontrar además de FAQ y HOWTO, una serie de libros y manuales en castellano (algunos son traducciones) que abarcan casi todos los aspectos de Linux. En este momento (Febrero de 2003) hay disponibles:

- *Guía de Administración de Redes*, OLAF KIRCH.
- *Guía del Usuario de Linux*, LARRY GREENFIELD.
- *Guía Linux de Programación*, SVEN GOLDT.
- *Guía del enRootador de Linux*, ERIC DUMAS.
- *Guía del Núcleo*, OLAF KIRCH.
- *Manual de GNU Bison*, CHARLES DONNELLY y RICHARD STALLMAN.
- *Guía de instalación oficial de Red Hat Linux*, Red Hat Software, Inc..
- *Introducción a la administración de redes tcp-ip*, CHARLES L. HEDRICK.
- *Manual de GNU Flex*, VERN PAXSON.
- *DNS*. JOSÉ VICENTE NÚÑEZ ZULETA
- *Seguridad en Unix*, ANTONIO VILLALÓN.
- *Guía de Seguridad del Administrador de Linux*, KURT SEIFRED.
- *Usando SAMBA*, ROBERT ECKSTEIN, DAVID COLLIER-BROWN y PETER KELLY.
- *LDP Manifiesto*, MICHAEL K. JONSHON.
- *Guía de Programación de Módulos del Núcleo Linux*, ORI POMERANTZ.
- *Dentro del núcleo Linux 2.4*, TIGRAN AIVAZIAN.
- *Guía de usuario de Ruby*, por MATZ.
- *Interconexión IrDA con Linux*, VICENTE D. FERNÁNDEZ.
- *Administración de una red local basada en Internet*, CHARLES L. HEDRICK.
- *Análisis de seguridad de la familia de protocolos TCP/IP y sus servicios asociados*, RAÚL SILES PELÁEZ.
- *Guía de usuario de Windowmaker*, ALFREDO K. KOJIMA.
- *Linux Instalación y Primeros Pasos*, MATT WELSH.
- *Guía Informal al Bloqueo*, PAUL RUSTY RUSSELL.
- *Controladores de Ratón*, ALAN COX.
- *Interfaz de Programación del Controlador MCA*, ALAN COX, DAVID WEINEHALL, CHRIS BEAUREGARD.
- *Guía de Usuario del Controlador Tulip*, JEFF GARZIK.
- *Guía de Programación de PPP Síncrono y Cisco HDLC*, ALAN COX.
- *Tutorial de PHP y MySQL*, JOSÉ ANTONIO RODRÍGUEZ.
- *Accesos a Dispositivos Independientes del Bus*, MATTHEW WILCOX.
- *Guía del Proefs del Núcleo Linux*, ERIK (J.A.K.) MOUW.
- *Desarrollo de Código Abierto con CVS*, KARL FOGEL.
- *El manual para el clustering con openMosix*, MIQUEL CATALÁN I COÏT.
- *Cómo crear un entorno chroot en Debian*, COLIN WALTERS.
- *Experiencias instalando Red Hat 6.2*, ANTONIO REGIDOR.
- *Un paseo detallado por la instalación de Debian 3.0*, CLINTON DE YOUNG.
- *Aprendiendo a Aprender Linux: Guías para colegios con plataforma de referencia S-Helio 1.1*, VLADIMIR TÁMARA, JAIME IRVING DÁVILA, PABLO CHAMORRO, IGOR TÁMARA.

### Tutoriales disponibles en Lucas



- *Linux: Manual de aprendizaje para novatos*. DANIEL WYRYTOWSKIJ.
- *Tutorial Linux*. JAVIER OROVENGUA MIGUEL
- *Curso de Linux*. DAVID FLORES SANTACRUZ FRANCISCO y JAVIER AHIJADO MARTÍN NAVARRO
- *Curso para Novatos*, ANTONIO CASTRO SNUR-MACHER
- *Tutorial de L<sup>A</sup>T<sub>E</sub>X*. SERGIO GARCÍA REUS, ANTONIO FLORES GIL
- *Tutorial Bocbook*, JAIME IRVING DÁVILA
- *Linux Fácil*. JOSÉ M. LAVEDA MOLINA
- *Manual de Unix*. JONATHAN NOEL TOMBS y JORGE CHÁVEZ ORZÁEZ
- *Taller de Unix*. JOSÉ VICENTE NÚÑEZ ZULETA
- *Linux: de Novato a Novato*, GERARDO TRAVESEDO CABEZUELO
- *Tutorial de Python*, GUIDO VAN ROSSUM
- *Tutorial de PERL*, J. JULIÁN MERELO
- *Introducción a la programación en Perl, CGI y Javascript*, JOSÉ V. NÚÑEZ ZULETA
- *Introducción a la plataforma de publicación Web Cocoon*, SAÚL ZARRATE
- *Tutorial de NIS y NFS*, JOSÉ V. NÚÑEZ ZULETA
- *Tutorial de NCurses*, ERIC S. RAYMOND y ZEYD M. BEN-HAMLIM
- *Curso de bases de datos y PostgreSQL*, VICTOS HUGO DORANTES y OTROS.
- *El sistema operativo Unix*, FERNANDO MAGAÑAS LAMAS.
- *Tutorial ADA y Glade*, ÁLVARO LÓPEZ.
- *Programación Básica de Sockets en Unix para Novatos*, BRACAMAN.
- *Guía breve Qmail*, DIEGO BRAVO ESTRADA.
- *Guía breve Sendmail*, DIEGO BRAVO ESTRADA.
- *Guía breve Tripwire*, DIEGO BRAVO ESTRADA.
- *Tutorial de Wincvs*, JAIME IRVING DÁVILA, JUAN PABLO QUIROGA
- *Tutorial para escribir plugins en Jedit*, JAIRO MARTÍNEZ.
- *Recomendaciones de seguridad en sistemas distribuidos de cómputo*, DIEGO BRAVO ESTRADA.
- *Modelado de Sistemas con UML*, POPKIN SOFTWARE AND SYSTEMS

### 2.1.2. Web interesantes sobre Linux

Además de las ya citadas (LuCAS e INSFLUG), hay que destacar al menos varias Web más en castellano:

- HispaLinux <http://www.hispalinux.es>: Web de la Asociación de usuarios Españoles de Linux. Desde esta web hay enlaces a casi todos los recursos de Linux en Castellano.
- LinuxLandia <http://www.linuxlots.com/~barreiro>: LinuxLandia, página personal de Manuel de Vega coordinador de la traducción del manual de usuario de la Red Hat 5.0 y de la Guía de Usuario de Gnome (disponibles en esta Web, además de más cosas)
- El Rincón de Linux <http://www.linux-es.com/>: pretende ser un punto de partida para aquellos que necesitan encontrar información sobre Linux.
- Web de los grupos de noticias es.comp.os.linux.\* <http://www.escomposlinux.org>. Muy buena, imprescindible (sobre todo cuando hablemos de *hardware*).
- La Web de Pedro Reina <http://www.pedroreina.org/>

Con respecto a web internacionales, existen muchísimas, sólo vamos a enumerar:

- Linux Organization [www.linux.org](http://www.linux.org): Completa web sobre todo lo concerniente al mundo de Linux.
- SAL <http://sal.kachinatech.com>: Scientific Applications on Linux.



- <http://rpmfind.net/linux/RPM>: Base de Datos con todos los paquetes disponibles para Linux.
- <http://freshmeat.net>: Software para Linux
- <http://sourceforge.net>: Servicio gratuito para desarrolladores de Software abierto.
- <http://www.ayamura.org/installer/ftpsites2.html> Lista de programas para Linux y servidores ftp de dónde bajarlos.
- Free software Europa <http://www.fsfeurope.org/index.es.html>: página de la Free Software Foundation Europe

### 2.1.3. News.

Linux es un sistema operativo dinámico, tanto el núcleo como las aplicaciones mejoran (y aumentan en número) cada día. Los grupos de noticias y las listas de correo son los medios de transmisión idóneos para que todas las noticias y mejoras se conozcan por todo el planeta en cuestión de minutos.

En la página El Rincón de Linux (Documentación) <http://www.linux-es.com/documentacion.php>, hay un enlace mediante el cual podemos conocer los grupos de noticias existentes sobre Linux.

Aparte de ese listado amplio, hay que destacar:

- News:
  - [es.comp.os.linux.instalacion](mailto:es.comp.os.linux.instalacion)
  - [es.comp.os.linux.redes](mailto:es.comp.os.linux.redes)
  - [es.comp.os.linux.programacion](mailto:es.comp.os.linux.programacion)
  - [es.comp.os.linux.misc](mailto:es.comp.os.linux.misc)
  - [es.comp.os.linux.anuncios](mailto:es.comp.os.linux.anuncios)

### 2.1.4. Revistas sobre Linux

#### En formato electrónico.

- Linux Gazette <http://www.linuxgazette.com>
- Linux Gazette, edición en castellano <http://gaceta.piensa.com>
- LinuxFocus <http://tldp.org/linuxfocus/Castellano/>

#### En castellano

- Sólo Linux (Prens@ Técnica)
- Linux Actual (Prens@ Técnica)
- Sólo Programadores Linux (Revistas Profesionales S.L.)
- Linux Magazine (Megamultimedia)
- Todo Linux (Iberprensa)
- Linux LXFormat (MC Ediciones)



### 2.1.5. Direcciones de Linux y educación.

- Linux va a la escuela <http://escuela.linux.org.ve>
- Software educativo para Linux
  - SEUL <http://www.seul.org/>
  - <http://www.offset.org/>

## Capítulo 3

# Instalación

**P:** Tengo problemas con mi máquina en Windows. ¿Podrías ayudarme?

**R:** Claro. Tira esa basura de Microsoft e instala Linux

(*Cómo hacer preguntas de manera inteligente*, ERIC S. RAIMOND)

### 3.1. Antes de la instalación.

Todo el proceso de instalación está perfectamente documentado en el *Manual oficial de instalación de Red Hat Linux para x86*.

Antes de instalar Linux es necesario conocer bien el hardware del que disponemos. Para evitarnos quebraderos de cabeza y tener que reinstalar varias veces Linux en nuestro equipo, es conveniente que hagamos un listado de los elementos básicos que tenemos.

También deberíamos asegurarnos de que la versión con la que vamos a trabajar tiene los controladores de dispositivo necesarios para gestionar los distintos periféricos de que disponemos, para los más estándar no debería haber ningún problema. En las siguientes direcciones podemos comprobar si nuestro hardware está soportado por Linux:

- Hardware soportado por la RedHat <http://hardware.redhat.com>
- De obligada visita: <http://www.escomposlinux.org/hardware/index.php> página de los grupos de noticias [es.comp.os.linux.\\*](mailto:es.comp.os.linux.*) en donde se contempla el hardware soportado por Linux, está en castellano.

En el listado de componentes tendrían que estar al menos los siguientes:

- Procesador: Linux se puede ejecutar en cualquier 386 o superior y compatibles.
- Discos duros: especificando el número, tamaño y tipo. Si disponemos de varios tenemos que tener claro en cuál vamos a instalar Linux. Si usamos un interfaz IDE y está en el primer canal como maestro se llamará */dev/hda*<sup>1</sup> (en cierto sentido la unidad C: del Dos), si es el esclavo de ese canal será */dev/hdb*. Si el disco es SCSI sería */dev/sd0*, */dev/sd1*, ..<sup>2</sup>
- Memoria RAM de la que dispone nuestro equipo: si queremos trabajar aceptablemente en modo gráfico deberíamos disponer de más de 64 MB<sup>3</sup>.

---

<sup>1</sup> Volveremos sobre este tema cuando usemos DiskDruid y en temas posteriores

<sup>2</sup> Para las unidades de disquetes sería:

- */dev/fd0* para la unidad A.
- */dev/fd1* para la unidad B

<sup>3</sup> Teóricamente se puede trabajar en modo texto con tan sólo 4 MB de RAM.

Si queremos trabajar con el OpenOffice holgadamente es necesario disponer de 128 MB de memoria RAM

- Tipo de CDROM, marca y modelo, interfaz que utiliza: IDE, SCSI, otros.
- Si disponemos de adaptador SCSI y en su caso la marca y el modelo.
- Tarjeta gráfica: Linux funciona bien con cualquier tarjeta gráfica en modo texto, pero si queremos trabajar en modo gráfico necesitamos que sea compatible con la versión de XFree86 que estemos instalando. Tenemos que conocer de nuestra tarjeta:
  - Marca
  - Modelo
  - Memoria
  - Chipset
- Monitor: el tema del monitor es menos delicado, lo más que puede pasar es que trabajemos con resoluciones menores de las que admita el que tenemos. Los datos necesarios son:
  - Resolución máxima.
  - El intervalo de trabajo de barrido horizontal y de refresco vertical de nuestro monitor<sup>4</sup>.
- Ratón: marca, modelo y tipo.
- Tarjeta de red: marca y modelo. Si nuestro ordenador está conectado a una red local (no Internet) necesitaríamos conocer además:
  - Nombre de la máquina.
  - Nombre del dominio.
  - Dirección IP de nuestra máquina.
  - Máscara de red.
  - Dirección del encaminador (router o gateway).
  - Dirección del servidor de nombres.

### 3.2. Métodos de instalación.

La RedHat y casi cualquier versión de Linux permite diferentes métodos de instalación:

1. Desde CD<sup>5</sup>
  - Con arranque desde el propio CD.
  - Con un disco de arranque.
2. Desde un disco duro local<sup>6</sup>.
3. Utilizando una Red.
  - a) Vía NFS.
  - b) Vía FTP.
  - c) Vía HTTP.

La más normal es la primera y será la que analicemos.

---

<sup>4</sup>El programa de instalación detecta estos valores en la mayoría de los modelos más nuevos.

<sup>5</sup>Es el que vamos a utilizar nosotros y el único que vamos a documentar.

Es posible usar discos de arranque PCMCIA si optamos por instalar Red Hat Linux usando un dispositivo PCMCIA. También tenemos la posibilidad de usar una unidad de disco USB. Si es necesario usar alguno de los métodos anteriores o se opta por instalar Linux usando una red, consultar *The Official Red Hat Linux x86 Installation Guide* y la nota a pie de página 28

<sup>6</sup>Usando las imágenes ISO de la distribución y sólo desde sistemas de ficheros ext2, ext3 o FAT

### 3.3. Instalando Linux.

La versión de Linux que vamos a instalar es la RedHat 8.0, esta versión vio la luz hace 6 meses aproximadamente y presenta una serie de características que hacen que sea adecuada como inicio:

- Es fácil de instalar: el proceso de instalación se ha traducido al castellano.
- Es fácil de usar.
- Su actualización es sencilla
- Es potente.
- Es la más extendida.

Lo idóneo para instalar Linux sería disponer de un disco dedicado para su uso exclusivo con **al menos 3 GB de capacidad**. Las ventajas de disponer de este disco es que nos permitiría “cacharrear” en él disminuyendo el peligro que supone “trastear” en un disco compartido con otros sistemas operativos en el que tengamos almacenado nuestro trabajo.

En caso de no disponer de un disco dedicado para Linux y de que ya tengamos ocupado todo el espacio de nuestro disco tendremos que conseguir espacio libre para él<sup>7</sup>. La cantidad de espacio que necesitamos estará condicionada por el número de programas que vayamos a instalar, pero con esta versión lo recomendable es que no baje de 3 GB. Hay que tener en cuenta que aunque la instalación mínima con Gnome es de aproximadamente 1,5 GB, lo normal es que después se instalen una serie de programas que hacen recomendable disponer de esos 3GB.

Para conseguir liberar ese espacio podemos utilizar<sup>8</sup>:

1. Defrag + Fips (Web de Fips: <http://www.igd.fhg.de/~aschaeffe/fips/fips.html>)
2. Partition Magic (programa comercial). Web: <http://www.powerquest.com>

Veamos unas pinceladas de cómo se haría con la 1ª opción.

#### 3.3.1. Fips.

##### Introducción

FIPS es un programa gratuito que permite dividir una partición primaria (tipo FAT) en dos particiones sin que haya que borrar el contenido de la partición inicial. FIPS reduce el tamaño de la partición cambiando unos valores de la tabla de partición del sector de arranque. Para conseguir realizar esta división es necesario que los datos estén al principio del disco. Como en general esto no es así, tendremos que usar un defragmentador de disco antes de utilizar Fips. Si tenemos “suerte” y Windows no ha escrito sectores inamovibles al final del disco, mediante el uso conjunto del Defrag de Windows y de Fips podemos reparticionar el disco y conseguir espacio para instalar Linux.

Para poder usarlo tenemos que tener alguna partición primaria sin usar en la que liberar el espacio. Si ya tenemos 4 particiones primarias, Fips dará error.

Con estos apuntes se pretende conseguir una mini-guía del uso de Fips, el objetivo no es cubrir todas las posibilidades de ejecución y arranque del programa sino sólo una de las más estándar: Sistema Windows 9.x. Para profundizar sobre el uso de Fips se aconseja leer la documentación que acompaña al programa y que permite conocer todas sus posibilidades.

<sup>7</sup>También se puede instalar sobre una partición DOS pero no es recomendable ya que el sistema será más lento que si usamos particiones nativas Linux.

<sup>8</sup>También se puede usar parted <http://www.gnu.org/software/parted/>

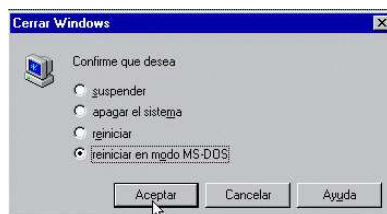
### Antes de usar Fips

Lo primero<sup>9</sup> que deberíamos de hacer siempre que “trasteemos” en nuestro disco duro es realizar una **copia de seguridad de nuestros datos**, si no lo hacemos nos arriesgamos a perder el trabajo que tengamos allí almacenado.

Además, deberíamos preparar un disco virgen para que *fips* guarde una copia de la FAT del disco por si queremos deshacer los cambios que hayamos realizado.

Los pasos siguientes serían:

1. Comprobar el estado de nuestro disco.
2. Copiar el directorio `dosutils\fips20`<sup>10</sup> del CD1 de la RedHat 8.0<sup>11</sup> en la unidad C:<sup>12</sup>.
3. Realizar una desfragmentación completa del disco.
4. Reiniciar el equipo en modo `msdos`<sup>13</sup>. Si reiniciamos el sistema podemos usar la tecla **F8**, la tecla **[Ctrl]** cuando se inicia Windows o bien usar un disco de arranque<sup>14</sup>.



### Ejecución de Fips

Una vez reiniciado nuestro equipo, situémonos en el subdirectorio donde esté el programa y ejecutemos FIPS tecleando *fips*. Las opciones que nos va a ofrecer Fips durante el transcurso de su ejecución son muy simples y se reducen a responder a 5 ó 6 preguntas y a introducir un disco cuando nos lo pida. Aclaremos esto:

- Lo primero que hace FIPS es comprobar el estado de nuestros discos<sup>15</sup>, si todo va bien nos mostrará una primera pantalla en la que nos informa del estado de las particiones y cómo están.

En este momento Fips nos pregunta en qué partición vamos a realizar la división, tendremos que pulsar un número correspondiente a la partición primaria que deseamos dividir.

- Tras seleccionar la partición, Fips la chequea y muestra información sobre ella, después nos aparecerá:

```
Do you want to make a backup copy of your root and boot sector before  
proceeding (y/n)?
```

es decir, si queremos realizar una copia de los sectores root y boot, en este paso pulsaremos *y* e introduciremos el disco virgen que teníamos preparado.

A continuación nos preguntará:

---

<sup>9</sup>Se parte de la idea de que el sistema operativo existente ya en el equipo es Windows 9x.

<sup>10</sup>Esta versión permite trabajar con fat32, la versión 1.5 de Fips no lo permite

<sup>11</sup>Se puede ejecutar Fips desde disquetes pero este método es más cómodo.

<sup>12</sup>También se puede descomprimir el fichero `fips20.zip` contenido en ese mismo directorio en C:

<sup>13</sup>Aunque se puede trabajar en modo `msdos` desde Windows no se recomienda.

<sup>14</sup>Para hacer un disco de arranque con fips necesitamos un disco con el sistema:

```
format a: /s
```

y después copiar a ese disco los ficheros: `fips.exe`, `restorrb.exe`, `errors.txt`, `fips.doc` y `fips.faq` del subdirectorio `\dosutils\fips20`

<sup>15</sup>Si tenemos varios discos lo primero es preguntarnos en cuál vamos a trabajar.



Do you have a bootable floppy disk in drive A: as described in the documentation (y/n)?

Pregunta a la que nuevamente responderemos y

- Cuando finalice el proceso de copia aparecerán dos números, correspondientes al tamaño de las particiones (vieja y nueva), usando las flechas del teclado podemos aumentar o disminuir estos números y modificar el tamaño de la partición inicial y, por tanto, de la nueva. Cuando todo esté a nuestro gusto, pulsaremos sobre la tecla *enter* para continuar.

**NOTA:** si Fips no nos permite “coger” todo el espacio libre que pensábamos que había en el disco, se debe a que en él hay sectores ocultos que impiden este proceso. Como primera medida podemos probar borrando el archivo de intercambio de windows y si no funciona tendremos que ir desinstalando programas hasta comprobar que esos sectores desaparecen.

- Tras una serie de chequeos nos preguntará si queremos continuar con los cambios o volver a editarlos:

Do you want to continue or reedit the partition table (c/r)?

Pulsaremos *c* para continuar con esos datos o *r* para volver y modificar de nuevo los tamaños de las particiones.

- Si hemos pulsado *c* aparecerá el mensaje

Ready to write new partition scheme to disk Do you want to proceed (y/n)?

Procederemos pulsando *y* y Fips modificará el tamaño de las particiones.

- El último paso es reiniciar el ordenador para que el sistema detecte los cambios que hemos realizado.

#### Para finalizar.

Después de reiniciar el equipo y antes de proceder a instalar Linux deberíamos comprobar el estado de nuestra “vieja” partición primaria, ejecutando el programa Scandisk.

Fips permite restaurar las particiones y dejarlas tal cual estaban antes de su utilización. Para esto necesitamos el disco donde Fips guarda la información sobre el sector root y boot. Para restaurar el estado de las particiones debemos utilizar el programa `restorrb.exe` que hay en el subdirectorio `fips20`. De igual manera que antes, la ejecución de este programa se debe de hacer reiniciando el equipo en modo MS-DOS e insertar el disquete de “rescate” en la unidad A:.

### 3.3.2. Comencemos. Instalando Linux.

Lo más usual es que instalemos Linux arrancando desde la unidad de CD ya que sólo en BIOS antiguas (primeros pentium y 486) esta función no está soportada. Como en general sí podremos hacerlo usando sólo el CD, podemos saltarnos los dos párrafos siguientes e ir a 3.3.3.

#### Usando un disco de arranque

Para instalar Linux de esta forma el proceso es básicamente el mismo que si arrancamos desde el CD y la única diferencia estriba en la creación del disco de arranque. Veamos cómo se hace:

1. Introducimos un disco formateado en la unidad *a* :
2. Desde Windows<sup>16</sup> 9.x ejecutar el programa del CD1 de RedHat `\dosutils\rawritewin\rawritewin.exe`<sup>17</sup>.
3. En la pestaña Cuando nos pregunte por las imágenes hay que decirle que es el fichero `\images\boot.img`<sup>18</sup>

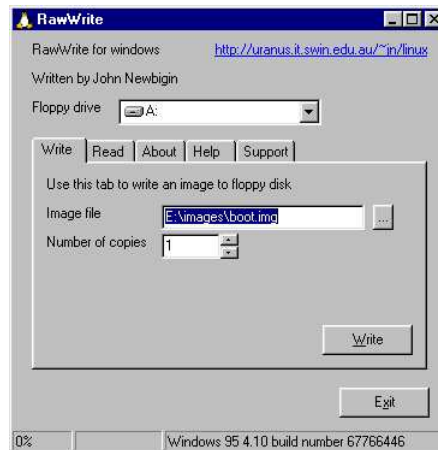
<sup>16</sup>Si ya tenemos un Linux instalado, podemos crear un disco de instalación usando la orden:

```
dd if=/mnt/cdrom/images/boot.img of=/dev/fd0 bs=1440k
```

dónde los path de los ficheros son los usuales en RedHat.

<sup>17</sup>Podemos hacer esto mismo en modo MSDOS usando el programa `\dosutils\rawrite.exe`

<sup>18</sup>Merece la pena un comentario, en `\images` hay varios ficheros más:



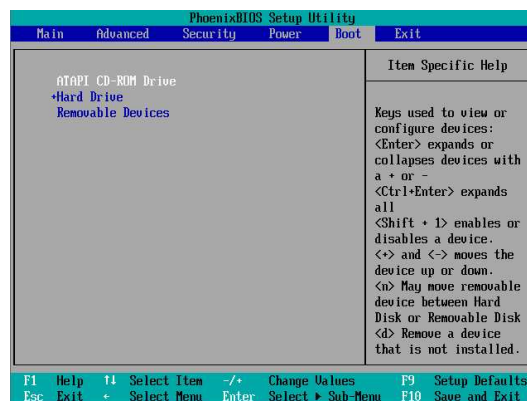
4. Reiniciamos el ordenador con el disco de arranque en la unidad A:
5. A partir de aquí el proceso es el mismo que si instalamos desde el CD.

### Desde DOS con acceso al CD

Si desde MSDOS (no desde una ventana en Windows) podemos acceder a la unidad de CD podemos iniciar la instalación usando el programa `/dosutils/autoboot.bat`.

### 3.3.3. Instalación de Linux desde un CD.

Si ya tenemos el equipo preparado y hemos recopilado la información necesaria sobre nuestro hardware podemos comenzar el proceso de instalación.



- `bootnet.img`
- `drvblock.img`
- `drvnet.img`
- `pcmcia.img`
- `pcmciaadd.img`

Si nos interesa instalar Linux en nuestra máquina desde una red local tendremos que usar `bootnet.img` en vez de `boot.img`, si queremos instalar Linux en un portátil usando una tarjeta pcmcia tendremos que crear un disco con el fichero `pcmcia.img` e iniciar el proceso de instalación con él.

Los ficheros `drvblock.img`, `drvnet.img`, `pcmciaadd.img` los usaremos como discos de drivers para que Linux reconozca algún hardware no estándar.

Lo primero que tendremos que hacer es modificar la BIOS para que arranque directamente desde el CD1 de la RedHat 8.0 o bien arrancar con nuestro disco de inicio.



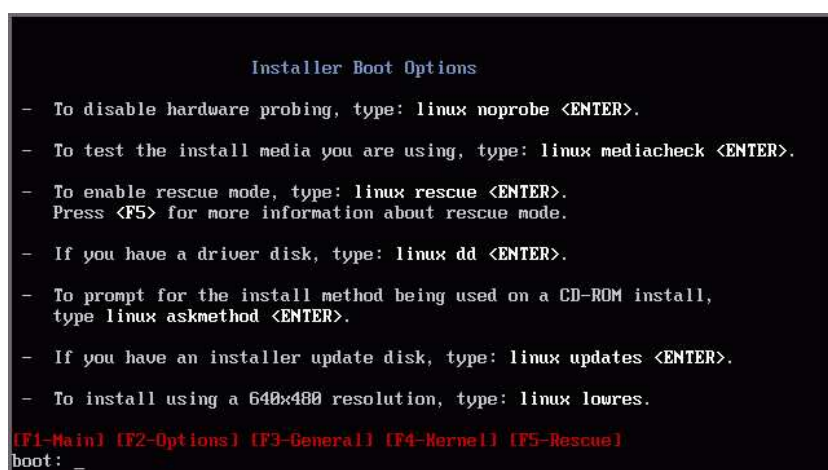
Se empieza a cargar Linux y nos aparecerá un menú con las siguientes opciones:

- Para instalar o actualizar Linux en modo gráfico pulsar Intro
- Para instalar o actualizar en modo texto escribir `linux text` y después Intro

Usando las teclas de función señaladas podemos acceder a distintas opciones:

**[F1-Main]** El menú anterior

**[F2-Options]** Otro menú con más opciones de instalación



Comentemos un poco esta ventana y el efecto de escribir tras `boot:`

**linux noprobe** permite la instalación en modo "experto".

**linux mediacheck** Para probar la integridad checksum del CD o la imagen ISO desde la que vamos a instalar Linux

**linux rescue** Permite entrar en Linux en modo de rescate (por si no hay otra forma de acceder al sistema)<sup>19</sup>

**linux dd** Si necesitamos usar un disco de drivers para trabajar con hardware no estándar<sup>20</sup>.

**linux askmethod** Permite seleccionar desde dónde vamos a instalar la Red Hat Linux (por si optamos por no instalar desde el CD)

**linux\_updates** Para actualizar nuestro sistema con el cd de actualizaciones.

**linux lowres** Inicia la instalación en un modo gráfico de baja resolución.<sup>21</sup>

[F3-General] Ayuda general sobre la instalación.

[F4-Kernel] Información sobre parámetros que podemos pasar al núcleo en el arranque

[F5-Rescue] Permite entrar en Linux en modo de rescate (por si no hay otra forma de acceder al sistema)

Optaremos para esta sesión por la instalación en modo gráfico. Para comenzar este proceso de instalación tan sólo tenemos que pulsar `intro` en la pantalla anterior (con [F1] podemos volver a ella) y en unos pocos segundos tendremos la pantalla de bienvenida a RedHat.<sup>22</sup>

Con respecto a si instalar en un modo o en otro comentar que son prácticamente iguales y resaltar que si bien vamos a optar por el modo gráfico, el modo texto es más rápido y no presenta mayor dificultad.

Si esperamos unos segundos o bien pulsamos `intro` comenzará el proceso de instalación:

### 3.3.4. Lenguaje de instalación, teclado y ratón.

Se nos preguntará si queremos testear el CD, optaremos por “pasar”: **Skip**.

Tras pulsar en **Next** en la pantalla de bienvenida, accederemos a la pantalla que permite seleccionar el lenguaje de instalación, en nuestro caso seleccionaremos **Español** y aceptaremos:

<sup>19</sup>Véase el Capítulo 8:Modo de rescate del *Manual oficial de personalización de Red Hat Linux*

<sup>20</sup>Véase la nota 18 de la página 28

<sup>21</sup>Si escribimos

```
boot: linux resolution=640x480
```

conseguimos el mismo efecto.

Para ampliar sobre los parámetros que podemos pasar al núcleo en el arranque consultar el Apéndice H del *The Official Red Hat Linux x86 Installation Guide*

<sup>22</sup>Antes de seguir, comentar que en todo el proceso de instalación podemos ver distintas pantallas mediante el uso de varias teclas. Casi nunca es necesario usarlas pero por si queréis ver su efecto son:

Teclas	Pantalla
[Ctrl]+[Alt]+[F7]	Pantalla por defecto (gráfica) de instalación
[Ctrl]+[Alt]+[F1]	Diálogo de instalación
[Ctrl]+[Alt]+[F2]	Permite ejecutar órdenes
[Ctrl]+[Alt]+[F3]	Registro de cómo va la instalación
[Ctrl]+[Alt]+[F4]	Mensajes del sistema
[Ctrl]+[Alt]+[F5]	Otros mensajes

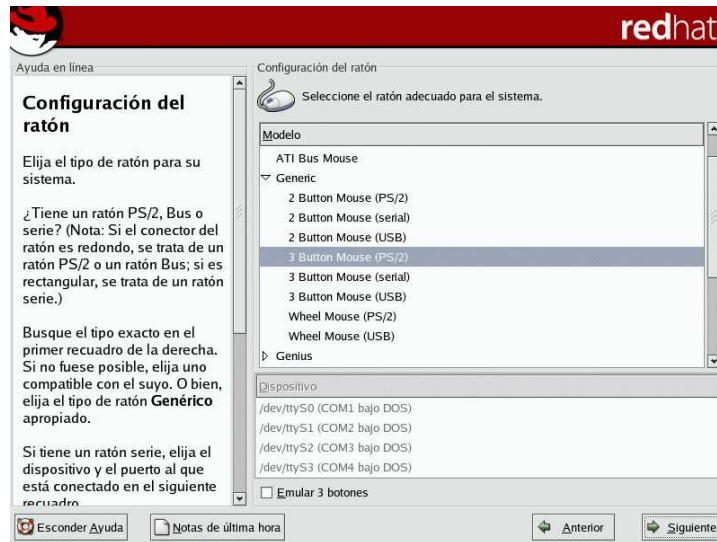
Si estamos en la pantalla gráfica, pasaremos a las de caracteres (o alfanuméricas) mediante [Ctrl]+[Alt]+[Fx]. Si estamos en una pantalla alfanumérica, pasamos a otra alfanumérica con [Alt]+[Fx] y a la pantalla gráfica con [Alt]+[F7].



Si continuamos, aparecerá una ventana que nos permite configurar el teclado, dicha ventana es



Cuando todo esté bien pulsamos sobre **Siguiente** y pasaremos a la pantalla mediante la cual vamos a poder configurar nuestro ratón:



Con los ratones estándar el programa de instalación detecta el puntero del que disponemos, si no es así tendremos que seleccionarlo de la lista desplegable.

Merece la pena resaltar dos aspectos de esta pantalla:

- Es mejor marcar la casilla de Emular 3 botones
- Los puertos del MSDOS en Linux reciben otros nombres distintos a los que tendremos que ir habituándonos. Así, el puerto COM1 del DOS, es en Linux el puerto `ttyS0`.

Tras configurar nuestro ratón, comienza el proceso de instalación propiamente dicho:

### 3.3.5. Selección del tipo de instalación.



Aquí es interesante detenerse un poco. Con esta versión de RedHat podemos seleccionar 5 tipos diferentes de instalaciones predefinidas. Como información añadida<sup>23</sup> a la que aparece en la pantalla:

<sup>23</sup>Si no se entiende toda la tabla, no nos asustemos. Estamos comenzando.

TIPO	Escritorio Personal	Estación de Trabajo	Servidor	Personalizada
<b>Por defecto</b>	1,5 GB con GNOME	2,0 GB con GNOME y herramientas de desarrollo	Sin entorno gráfico 1,3-1,4 GB	Sin entorno gráfico 0,4 GB
<b>GNOME + KDE</b>	1,8 GB	2,3 GB	2,1 GB	4,5 GB
<b>Particion. autom.</b>	Swap $\approx 2 * RAM$ /boot 100 MB / resto	Swap $\approx 2 * RAM$ /boot 100 MB / resto	Swap $\approx 2 * RAM$ /usr depende espacio libre /home depende espacio libre /var 256 MB /boot 100 MB	

Si tenemos una versión anterior de RedHat (desde la versión 6.2 o superior) podemos **actualizar**<sup>24</sup> el sistema a esta nueva versión. Para una explicación más detallada de los distintos tipos de instalación os remitimos a la *Guía de Instalación* de Red Hat antes comentada.

En nuestro caso optaremos por la segunda (**Estación de trabajo**) ya que permite trabajar con Linux de forma más que aceptable sin tener demasiados conocimientos iniciales.

### 3.3.6. DiskDruid (Fdisk): crear las particiones.

Llegamos a la parte más delicada del proceso de instalación, la de crear/modificar las particiones en las que instalar nuestro Linux. Para eso disponemos de dos herramientas:

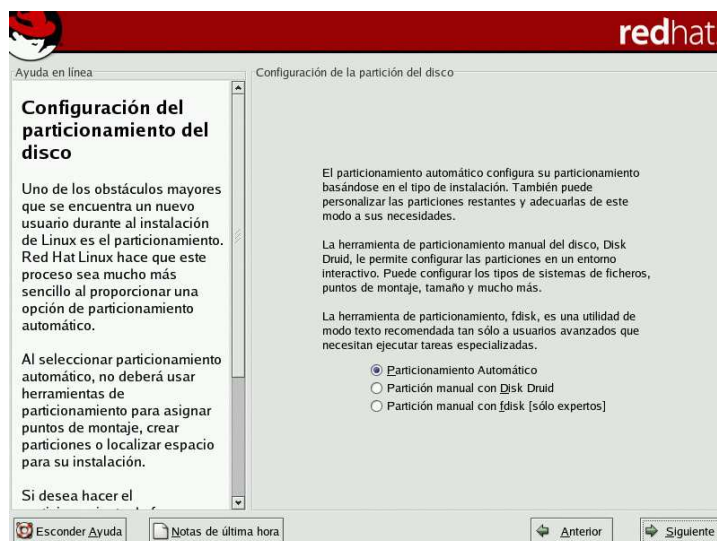
- **DiskDruid**: utilidad “amigable” para reparticionar discos.
- **fdisk**: similar en su funcionamiento al del MsDos. Es más potente, fiable y difícil de manejar que el anterior.

En esta versión de Red Hat, podemos hacer que sea el programa de instalación el que reparticione el disco sin que controlemos nosotros el proceso.

- 
- Si tenemos Windows NT instalado hay que elegir **Personalizada** y no instalar GRUB en el MBR. Lo mejor en este caso es crear un disco de arranque. Con el 2000 no hay problema.
  - El espacio que aparece en esta sección **Por defecto** es el mínimo necesario, por tanto, hay que añadir 1 GB al menos para poder trabajar después.

<sup>24</sup>Si no actualizamos y ya teníamos Linux instalado se borrará todo lo que tengamos.





Dependiendo de la forma en que tengamos estructurados nuestros discos podemos optar por:

- Usar el Particionamiento Automático de DiskDruid si
  - Disponemos de espacio libre (sin asignar a ninguna partición)
  - Vamos a borrar y reestructurar particiones Linux existentes
  - Vamos a borrar todas las particiones del disco (eliminaríamos los otros sistemas operativos instalados)
- Usar DiskDruid con particionamiento manual si:
  - Tenemos varias particiones y deseamos usar algunas para Linux
  - El espacio libre lo hemos conseguido usando Fips.



Como ya se ha comentado al inicio del tema, Linux llama a las particiones de nuestros discos duros de forma diferente a los sistemas MS-DOS. En Linux<sup>25</sup>:

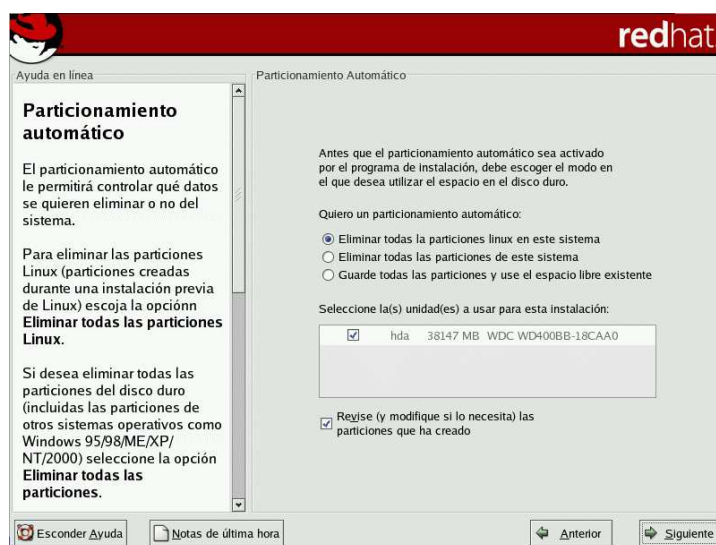
TIPO DE DISPOSITIVO	NOMBRE EN LINUX
<b>Maestro del primer canal IDE</b>	<b>/dev/hda</b>
Primera partición primaria	/dev/hda1
Segunda partición primaria	/dev/hda2
Tercera partición primaria	/dev/hda3
Cuarta partición primaria	/dev/hda4
Primera partición lógica	/dev/hda5
Segunda partición lógica	/dev/hda6
...	...
<b>Esclavo del primer canal IDE</b>	<b>/dev/hdb</b>
....	...
<b>Maestro del segundo canal IDE</b>	<b>/dev/hdc</b>
....	...
<b>Esclavo del segundo canal IDE</b>	<b>/dev/hdd</b>
....	...

<sup>25</sup>Para dispositivos SCSI tan sólo hay que cambiar **hdx** por **sdx**.




Tanto si disponemos de un disco dedicado como si nuestro disco es compartido el proceso de crear las particiones para instalar Linux es el mismo, lo único que necesitamos es habituarnos a la tabla anterior.

### Optamos por el particionamiento automático



Si optamos por el particionamiento automático, podremos elegir en qué discos vamos a instalar Linux (si disponemos de varios) y aunque no cambiemos nada deberíamos **revisar** la forma en que el programa DiskDruid ha particionado nuestro disco duro.

### En cualquier caso: manual o automático

 Linux necesita al menos dos particiones para trabajar<sup>26</sup>:

- **/**  $\mapsto$  Punto de montaje<sup>27</sup> (Mount Point), en ella se instalarán los paquetes que componen la distribución y almacenaremos nuestros datos, debería de ser de al menos 2 GB para no tener problemas de espacio.
- **Swap Partition**  $\mapsto$  Partición de intercambio, partición del disco duro que Linux utiliza como extensión de la memoria RAM del sistema. Hay una serie de normas a seguir para determinar el tamaño idóneo del archivo de intercambio<sup>28</sup>:
  - Para un trabajo “normal” con Linux el tamaño mínimo debería ser de 64 MB.
  - En general se recomienda que si tenemos 64 MB de RAM o menos sea el doble de ésta.

<sup>26</sup>Para nosotros, lo recomendable es disponer de al menos tres particiones, el uso de la tercera sería exclusivo de datos (partición /home )

<sup>27</sup>Por defecto Red Hat usa el sistema de ficheros *journaling* ext3. Se trata de una versión mejorada del anterior sistema de archivos de Linux (ext2). Al tratarse de un sistema transaccional se garantiza que casi siempre que se produzca una caída inesperada del sistema (por ejemplo por un corte eléctrico) no peligre la integridad de los datos.

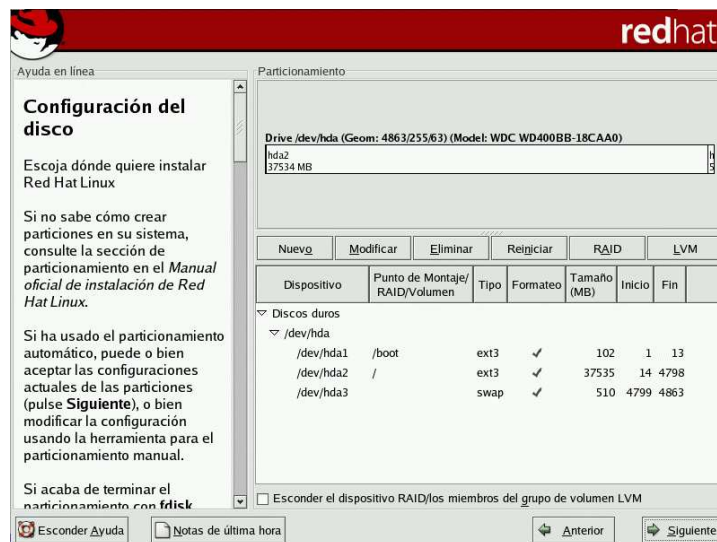
<sup>28</sup>Si se quiere ampliar sobre el tema se puede consultar:

- El capítulo 2 de *The Official Red Hat Linux Customization Guide*
- La *FAQ sobre Linux para principiantes* <http://www.linux-es.com/Faq>. En este documento, además de explicar ¿Qué es la Swap? y ¿cuánta necesitamos? hay enlaces a páginas más técnicas sobre el tema.

- La norma anterior se aplica hasta 128 MB de RAM. Si superamos este valor, el tamaño de la SWAP puede ser igual al tamaño de la RAM (sin superar 2 GB)<sup>29</sup>.

Con esas particiones es suficiente pero, según Red Hat, lo conveniente es disponer de tres (esas dos más una de 100 MB<sup>30</sup> para `/boot`). Además, recomienda como tamaño de la partición Swap el doble de la RAM instalada (con un máximo de 2 GB). Esta cuestión no nos parece estrictamente extrapolable a PCs domésticos con suficiente memoria RAM. Por ejemplo, en un PC de 256 MB o más de RAM y no demasiado disco duro puede ser más interesante optar por un tamaño similar al de la RAM (o incluso menor) para la partición de intercambio. Se puede optar por modificar los valores del particionamiento automático (no es obligatorio) y dejar la tabla de particiones como sigue:

- Una partición para `/boot` de 100 MB.
- Una partición Swap de 128 MB como mínimo y 512 MB como máximo aproximadamente (intentando respetar el criterio de doblar la cantidad de RAM instalada)
- El resto para el Punto de montaje `/`



Usando el programa DiskDruid podemos optar por:

- **Nuevo:** añadir una partición.
- **Modificar:** editar una partición existente.
- **Eliminar:** borrar una partición existente.
- **Reiniciar:** restablecer el estado de las particiones antes de comenzar a trabajar con DiskDruid.
- **RAID<sup>31</sup>:** con esta opción podemos, por ejemplo, combinar varios discos para conseguir mejores prestaciones o mayor seguridad (para usuarios expertos).

RAM	SWAP
RAM < 128 MB	2 * RAM < SWAP < 256 MB
RAM > 128 MB	256 MB < SWAP < 2 * RAM

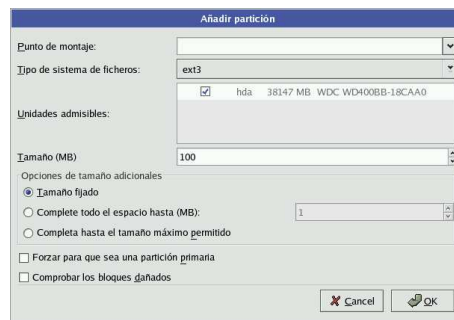
<sup>29</sup> No tiene sentido que sea mayor. En ella estará el Kernel compilado más otros archivos necesarios para iniciar el equipo.

<sup>31</sup> Para ampliar consultar *The Official Red Hat Linux Customization Guide*, capítulos 3 y 9.

- **LVM<sup>32</sup>**: Permite que usemos varios discos duros como un solo volumen lógico, es decir, disponer de todo el espacio como si se tratase de un sólo disco. Además, se puede modificar el tamaño de las particiones definidas dentro del volumen lógico con solo añadir otro disco duro (para usuarios expertos).

Si no hemos optado por el particionamiento automático, tendremos que crear nosotros las particiones. El primer paso es del de garantizarnos que disponemos de espacio libre suficiente para Linux (si es necesario borrando las particiones existentes que estemos seguros no vamos a necesitar para otros sistemas) y después:

1. Pulsaremos sobre el botón **Nueva**:



Rellenaremos los campos:

- Punto de montaje: `/boot`
- Tipo de sistema de ficheros: `ext3`
- Tamaño (MB): 100

y **OK**.

2. Continuemos con la partición de intercambio, **Nueva**

- Tipo de sistema de ficheros: `Linux Swap`
- Tamaño (megs): `2 * (Tamaño RAM)` (ya comentado anteriormente, debemos ajustarla a nuestro sistema)

Y **OK**, ya tenemos creada nuestra partición Swap.

3. La última, **Nueva** y, en la ventana que aparece, rellenaremos:

- Punto de montaje: `/`
- Tipo de sistema de ficheros: `ext3`
- Completa hasta el tamaño máximo permitido: marcaremos esta opción si queremos que esta partición ocupe todo el espacio libre que queda en el disco, si no es así la dejaríamos sin marcar y en el campo anterior pondríamos el espacio que deseamos que ocupe la partición.

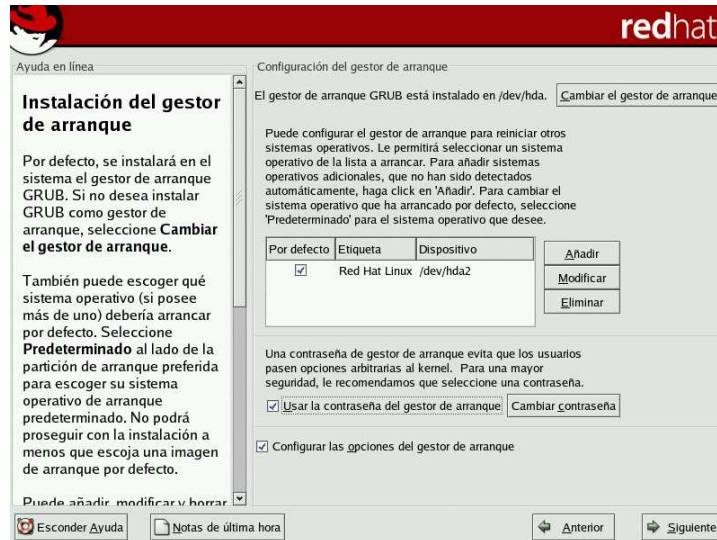
Y **OK**, ya tenemos creado nuestro punto de montaje.

Cuando estemos seguros de los cambios que hemos realizado continuaremos con la instalación.

<sup>32</sup>Para ampliar consultar *The Official Red Hat Linux Customization Guide*, capítulos 4 y 10.

### 3.3.7. Configuración de GRUB.

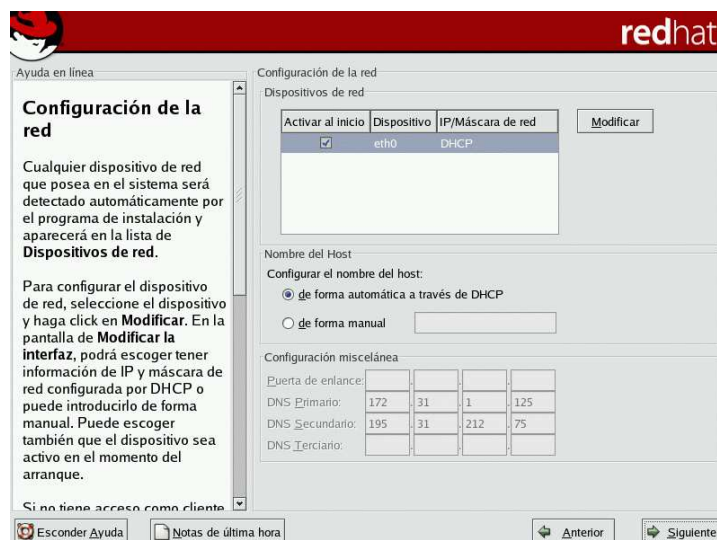
Desde aquí podemos controlar qué gestor de arranque vamos a usar: Grub o LILO y qué sistema operativo se va a cargar por defecto



Dejaremos todas las opciones que aparecen por defecto (no son las del gráfico). Si después queremos modificar alguna ya veremos cómo hacerlo una vez cargado Linux.<sup>33</sup>

### 3.3.8. Red

En caso de que tengamos una tarjeta de Red, comenzará el proceso de autodetección y después tendremos que configurar la red. Si no tenemos tarjeta de red este paso no aparece.

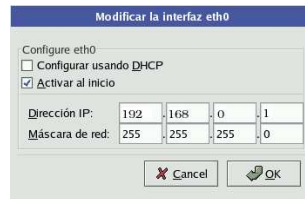


<sup>33</sup>Para los que tengáis NT comentar que se tiene que instalar en el primer sector de la partición raíz y no en el MBR. Esto se consigue a través de la ventana a la que se accede si marcamos en **Configurar las opciones del gestor de arranque**.

Si tenemos tarjeta de red ahora es el momento de configurarla. Tras comprobar que se activa al inicio y salvo que tengamos claro qué es DHCP, deberíamos optar por configurarla de forma manual. Para eso pulsemos sobre modificar y configuremos:

**Dirección IP:** si es local 192.168.0.1 u otra del mismo tipo (ojo que si usamos esta tarjeta para conectamos a internet con un router en modo multipuesto tendremos que poner aquí una IP de la misma red que la del router).

**Máscara de red:** 255.255.255.0 (o la que corresponda)



**De forma manual:** el nombre de nuestra máquina.

**Puerta de enlace:** si no sabemos bien qué poner lo mejor es dejarlo en blanco<sup>34</sup>.

**DNS:** si tenemos acceso a internet podemos poner las direcciones de nuestros servidores de nombres aquí. Si no es mejor dejarlos en blanco.

Tras configurar los parámetros de red dejaremos las opciones por defecto en la ventana **Configuración del firewall**, en entregas posteriores veremos la forma de cambiarlo<sup>35</sup>.

### 3.3.9. Selección de más idiomas y de la Zona Horaria



Pues eso, que estamos en España y si queremos que los menús salgan en castellano (podemos optar por cargar vamos idiomas) y tener el reloj a su hora habrá que seleccionar Madrid.

<sup>34</sup>Todos estos datos podremos modificarlos después.

En el caso de que usemos este interfaz de red para conectemos a internet con un router ADSL, en modo multipuesto, pondremos aquí la dirección del router.

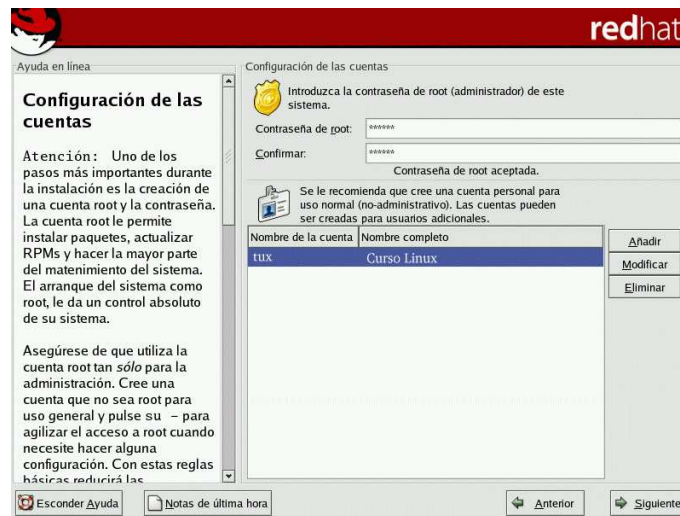
<sup>35</sup>Para saber ya de qué va esto, se puede consultar Configuración del firewall en el Capítulo 3 de *The Official Red Hat Linux x86 Installation Guide*

### 3.3.10. Gestión de usuarios y configuración de la autenticación.

En todo sistema Unix existen dos tipos de usuarios:

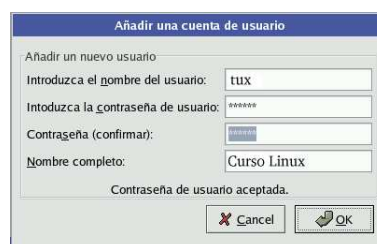
1. el “jefe”, superusuario o **root** que tiene acceso a todo el sistema. El equivalente al Administrador en sistemas NT/2000/XP, y
2. usuarios “normales y corrientes” que sólo pueden destrozar sus ficheros sin tocar partes delicadas del sistema.

En PCs “caseros” los dos tipos de cuentas se usan casi siempre por la misma persona, pero interesa trabajar como usuario normal y entrar como root sólo cuando vayamos a modificar la configuración de nuestro equipo.



Mediante esta ventana pondremos una contraseña al root (al menos 6 caracteres) que no debemos olvidar, ya que si lo hacemos no tendremos acceso<sup>36</sup> a modificar la configuración de nuestra máquina.

La contraseña para el root aparecerá como (\*\*\*\*\*) y hasta que no coincida la contraseña con la confirmación, no podremos continuar, apareciendo la opción *Siguiente* desactivada. Crearemos al menos un usuario con el que poder trabajar normalmente, seleccionando la opción *Añadir*.



### 3.3.11. Seleccionar paquetes.

Ya casi hemos terminado. Veremos una ventana que nos informa de algunas de las aplicaciones que se van a instalar. También podemos optar por seleccionar los paquetes de forma individual. Mediante esta opción accedemos a la posibilidad de seleccionar qué programas vamos a instalar, aunque por ahora lo vamos a dejar y vamos a instalar los que están seleccionados por defecto.

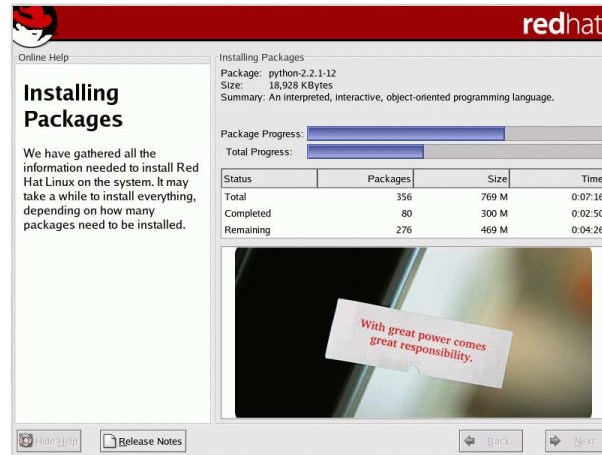
Si pulsamos **OK** comenzará el proceso de instalación.

<sup>36</sup>Existe un procedimiento de emergencia.



### 3.3.12. Instalación de los paquetes, disco de arranque.

Si continuamos comienza el proceso de instalación de los paquetes:

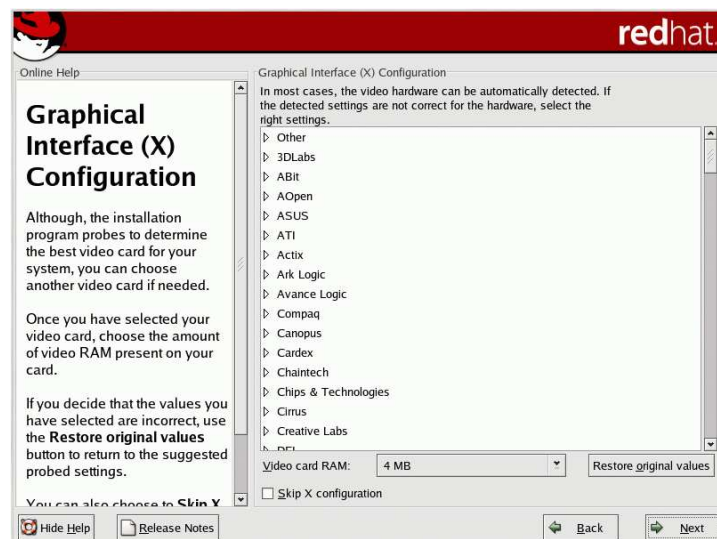


Como son tres discos, tendremos que introducirlos cuando nos los pida hasta que instale todos los programas por los que hemos optado..

Cuando termine la instalación de los paquetes se nos pedirá que introduzcamos un disco en `/dev/fd0` (¿o era unidad A:?) para crear un disco de arranque (dará la posibilidad de no hacerlo si no queremos)<sup>37</sup>. Si disponemos de un disquete lo mejor sería hacerlo. Lo agradeceremos en caso de emergencia.

### 3.3.13. Configuración del entorno gráfico

Pasamos a configurar las X<sup>38</sup>. En general, el propio programa de instalación detectará la tarjeta gráfica de nuestro equipo además de la memoria instalada.



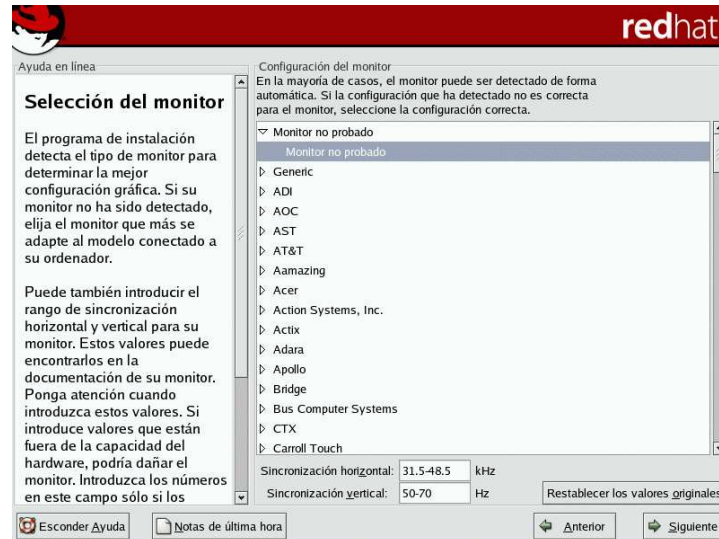
<sup>37</sup>En Linux podemos crearlo con el comando:

```
# mkbootdisk --device /dev/fd0 2.4.18-14
```

<sup>38</sup>Modo gráfico

### 3.3.14. Configuración del monitor

Seleccionaremos el monitor de la lista cuidando de que sea el nuestro o bien de que los valores de Sync. Horizontal y Vertical sí lo sean (es muy importante si no queremos dañarlo)



Podemos continuar sin tocar nada, pero se corre el riesgo de que con algunos monitores que admiten resoluciones elevadas las letras se vean muy pequeñas cuando trabajamos en modo gráfico. Es de todo punto aconsejable, antes de finalizar la instalación pulsar en **Comprobar la configuración** y, además probarlo para varios modos de vídeo/profundidades de color. El modo de 800x600 debe ser el de referencia.



Si marcamos **Tipo de inicio gráfico**<sup>39</sup>, accederemos al sistema en nivel 5<sup>40</sup>; además podemos elegir el entorno de escritorio por defecto.

Por fin, ya está, se acabó: cuando pulsemos sobre **Salir** se reiniciará el sistema y comenzaremos a trabajar con LINUX.

<sup>39</sup>En caso contrario, arranca en modo alfanumérico y una vez en Linux hay que ejecutar la orden `startx` para pasar al modo gráfico.

<sup>40</sup>Más adelante sabréis a qué nos referimos.



### 3.4. Comenzamos a trabajar con LINUX

Cuando reiniciemos el equipo nos aparecerá una ventana gráfica que indica que se ha cargado correctamente el gestor de arranque de Linux (será diferente si se opta por Grub o por LILO) y que nos va a permitir seleccionar el sistema operativo que deseamos ejecutar. Con las teclas del cursor podremos seleccionar **Red Hat Linux** (opción por defecto) o **DOS** (en el caso de tener sólo estos dos sistemas cargados). Si dejamos pasar un tiempo o pulsamos sobre **intro** él solo comienza la secuencia de arranque de Linux.



Cuando comience la carga del sistema veremos una serie de líneas que nos van informando sobre qué está ocurriendo, si a la derecha aparece OK es que todo va bien.

#### 3.4.1. Arranque en modo gráfico

Si nuestro sistema arranca en modo gráfico accederemos a una ventana similar a



si escribimos en el login

`root`<sup>41</sup>

y después la contraseña adecuada accederemos al entorno gráfico GNOME.



Recordemos de nuevo que en Linux podemos diferenciar dos tipos de usuarios (ya ampliaremos esto después), el

<sup>41</sup>Sólo se debe entrar como root cuando vayamos a modificar el sistema. Cuando trabajemos normalmente, se debe entrar usando la cuenta de usuario que hemos creado cuando instalábamos Linux, o que veremos cómo crear posteriormente.



**root** es el que manda sobre el sistema. En cierto modo, cada vez que alguien accede a una máquina windows 9x es el root<sup>42</sup>.

**otros\_usuarios** tienen que tener una cuenta registrada en el sistema para poder trabajar con él y si bien pueden trabajar con casi todos los programas de utilidad de la máquina, no pueden cambiar la configuración del sistema ni nada que afecte a otros usuarios. Sí que pueden modificar el aspecto de las aplicaciones, así como los ficheros que sean de su “propiedad”<sup>43</sup>.

### 3.4.2. Arranque en modo texto

Si no hemos podido configurar correctamente el modo gráfico o si hemos optado por iniciar Linux en modo texto, tras cargar todo el sistema aparecerá una pantalla en negro con el texto:

```
Red Hat Linux release 8.0 (Psyche)
```

```
Kernel 2.4.18-14 on an i586
```

```
localhost login:
```

Es el momento de escribir **root** y tras pulsar *intro* nos aparecerá:

```
Password:
```

e introduciremos la contraseña. Si todo ha ido bien y no nos hemos confundido con ningún carácter YA ESTÁ, ya estamos con Linux. Nos aparecerá una pantalla negra como la noche, con una sola línea a la derecha, una rayita blanca intermitente y la desesperación de no saber qué demonios hacer con aquello.

No desesperemos, esto es sólo el inicio y vamos a colorear pronto la pantalla.


En esta sesión inicial tan sólo nos interesa comenzar con Linux y poder trabajar con él de la forma más sencilla, y la más sencilla, que no siempre la más rápida ni la mejor, es en modo gráfico. Vamos allá, escribamos

```
startx
```

e *intro* y empezará a cargarse el entorno gráfico que tanto nos gusta. Ya estamos en GNOME uno de los entornos de escritorio que tenemos a nuestra disposición.

### 3.4.3. Apagado y reinicio del sistema

#### Inicio en modo gráfico

Una de las cosas que podemos hacer desde el entorno gráfico es reiniciar (reboot) o terminar con la sesión de trabajo (halt)<sup>44</sup>, para ello tan sólo tenemos que pulsar sobre el icono , aparecerá el menú

<sup>42</sup>Cuando un programa sólo lo puede usar el root lo indicaremos anteponiéndole el símbolo #.

<sup>43</sup>Si un programa lo puede ejecutar un usuario cualquiera lo indicaremos anteponiéndole el signo \$.

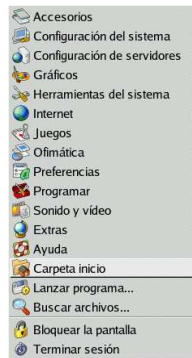
<sup>44</sup>En general el apagado y reinicio del sistema no se hace así salvo que trabajemos directamente en modo gráfico desde el inicio. Cuando se trabaja en modo texto se utiliza la orden `shutdown`, pero para facilitar las cosas en esta primera sesión se ha optado por este método en vez de la orden `shutdown`, que se verá más adelante. Las órdenes que “usualmente” utilizaremos serán:

- Para reiniciar el sistema (reboot):

```
shutdown -r now
```

- Para apagar el sistema (halt):

```
shutdown -h now
```




si pulsamos sobre **Terminar sesión** nos aparecerá la ventana:



que nos permite salir del entorno gráfico (Logout), reiniciar el equipo (Restart the computer o Reboot) o salir de Linux apagando la máquina (Shut Down o Halt).

Hay que tener en cuenta que para apagar el sistema es fundamental cerrar todos los procesos en ejecución y, por tanto, es necesario que el sistema termine correctamente antes de realizar la desconexión eléctrica.

 en modo gráfico no necesitamos iniciar una sesión de usuario para apagar el sistema. Si en la pantalla de registro (Página 44) pulsamos sobre **Sistema** aparece un menú que nos permite el apagado/reinicio del sistema.

### Inicio en modo texto (y gráfico)

Si hemos iniciado Red Hat en modo texto, tendremos que iniciar una sesión de usuario para ejecutar los comandos que permiten apagar/reiniciar el sistema, se trata de:

- Apagado:

```
halt
```

- Reinicio<sup>45</sup>

```
reboot
```

---

<sup>45</sup>Con la conocida combinación [Ctrl]+[Alt]+[Del] conseguimos el mismo efecto

## Capítulo 4

# Conexión a Internet: RTB, ADSL, RDSI.

Siempre me ha fascinado Internet, incluso antes de que existiera.

(*Servidor Apache*, RICH BOWEN & KEN COAR, Prentice Hall)

### 4.1. Introducción.

Linux e Internet van cogidos de la mano, sin Internet Linux posiblemente estaría “arrumbado” en el cajón de alguna universidad y no sería lo que es hoy.

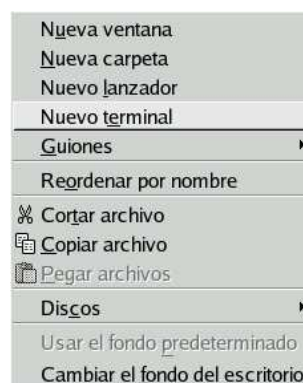
Mediante estos apuntes vamos a configurar la conexión a Internet de nuestro equipo. Se va a realizar la conexión utilizando un entorno gráfico (GNOME/KDE) e intentando que sea lo más estándar y guiada posible.

Además, aunque la conexión la vamos a establecer y utilizar como usuario el *root*, después veremos que podemos permitir que cualquier usuario se conecte o desconecte de internet.



Si tenemos que ejecutar comandos de Linux, mientras trabajamos en modo gráfico debemos usar un terminal gráfico o xterm<sup>1</sup>. Para abrir un terminal desde GNOME<sup>2</sup>, tenemos varias opciones, algunas son:

- En el menú principal de GNOME, pulsar en  → **Herramientas del sistema** → **Terminal**
- Pulsar con el botón derecho del ratón en una zona libre del escritorio y en el menú



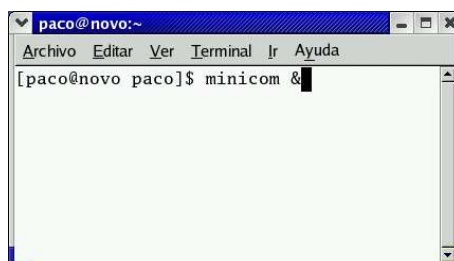
<sup>1</sup>En cierto sentido similar a una ventana MSDOS de Windows 9x.

<sup>2</sup>Desde KDE: **Menú principal de KDE** → **Sistema** → **Shell**

seleccionar **Nuevo Terminal**.

Si nuestra intención es la de ejecutar varios comandos podemos optar por que Linux los ejecute en segundo plano. Esto se consigue añadiendo al comando el símbolo &:

\$comando &



De esta manera podemos usar el terminal para ejecutar desde él varios programas (el programa se ejecuta pero aparece de nuevo el indicador de introducción de órdenes por si deseamos ejecutar otro comando)

La aclaración anterior permite constatar una constante en todo el curso: es inevitable que aparezcan conceptos y utilidades que no se corresponden con el tema que se está tratando. Esto ocurre en este caso y optaremos por introducirlos analizando sólo los aspectos básicos necesarios para poder conectar a internet.

Para ampliar:

- Capítulo 4 de *The Official Red Hat Linux Getting Started Guide*
- Capítulo 11 de *The Official Red Hat Linux Customization Guide*

## 4.2. RTB

Supondremos que nos asignan la dirección de forma dinámica, como ocurre con la mayoría de proveedores de Internet.

Antes de proceder a realizar la conexión a Internet usando un módem necesitamos una serie de datos:

1. Módem:

- a) Tipo de módem, puerto serie<sup>3</sup> al que está conectado.
- b) IRQ y direcciones de E/S.
- c) Velocidad del módem.

2. Datos relativos al proveedor (entre paréntesis los que usaremos de ejemplo<sup>4</sup>):

- a) Dominio de acceso (*cica.es*)
- b) N° de teléfono de acceso (950542000)
- c) Nombre de usuario (*codigo\_centro@cica*)
- d) Contraseña (\*\*\*\*\*)
- e) Método de autenticación (CHAP o PAP)
- f) Dirección IP del servidor de nombres de dominio (DNS: 150.214.3.9 y 150.214.90.11).

Ya que vamos a realizar labores de administración del sistema tendremos que entrar como superusuario (root) tanto en la configuración del módem como para la configuración de Internet.<sup>5</sup>

<sup>3</sup>Si no lo sabemos y tenemos Windows instalado, podemos usarlo para conocerlo.

<sup>4</sup>En general sólo necesitaremos los 4 primeros.

<sup>5</sup>Para indicar que el comando lo tiene que ejecutar el usuario root, usaremos #, y si un comando lo puede ejecutar un usuario normal, lo indicaremos anteponiendo a éste el carácter \$

### 4.2.1. Configuración del módem.

Lo primero que tenemos que conocer antes de iniciar el proceso de conexión a internet es saber si nuestro módem funcionará con Linux. Además, puede que necesitemos saber a qué puerto serie está conectado.



En Linux todo son ficheros (esta frase seguro que se repetirá en el curso), y los puertos serie también. Así, cada “fichero” `/dev/ttySx` se corresponde con el puerto de comunicaciones del DOS

Linux	MS-DOS
<code>ttyS0</code>	<code>COM1</code>
<code>ttyS1</code>	<code>COM2</code>
<code>ttyS2</code>	<code>COM3</code>
<code>ttyS3</code>	<code>COM4</code>

El mejor sitio para saber si nuestro módem funciona con Linux:

<http://www.idir.net/~gromitkc/latest.cgi>

Una página en la que encontrar información si tenemos problemas con el módem:

<http://www.escomposlinux.org/hardware>

En líneas generales podemos establecer que:

**Módem Externos:** No presentan ningún problema, se autodetectan.

**Módem Internos:** Si nuestro módem no es PCI no debería haber ningún problema.

**Winmódem:** La mayoría de los módem internos PCI no son módem completos y sólo son módem “software”. Han aparecido drivers para que algunos modelos de pseudomódem puedan funcionar bajo Linux. Para saber si nuestro módem es uno de estos lo mejor es mirar en las páginas

Linux Winmodem Support <http://linmodems.org/>

Winmódems no son módems [http://www.idir.net/~gromitkc/winmodem\\_es.html](http://www.idir.net/~gromitkc/winmodem_es.html)

Linmodem-HOWTO <http://www.tldp.org/HOWTO/Linmodem-HOWTO.html>

En general, y aunque estén soportados, no son fáciles de configurar y nuestra experiencia es que incluso los soportados dan bastantes problemas.

### 4.2.2. Configuración del interfaz ppp

La conexión a Internet en modo gráfico con esta versión de Red Hat es un “juego de niños”. Sólo hay que ejecutar (como root)



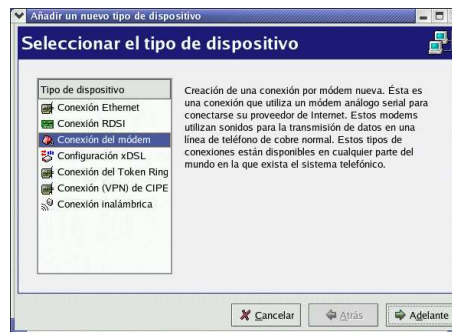
→Herramientas del sistema→Wizard de configuración de Internet<sup>6</sup>

<sup>6</sup>Equivale a ejecutar desde un terminal el comando:

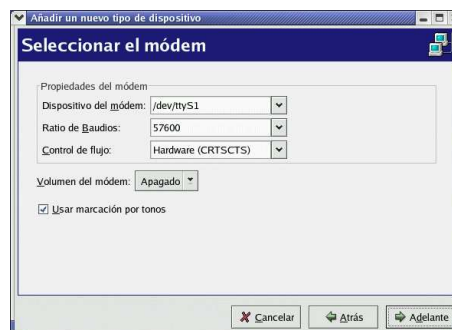
```
#internet-druid
```

o bien

```
#redhat-config-network-druid
```

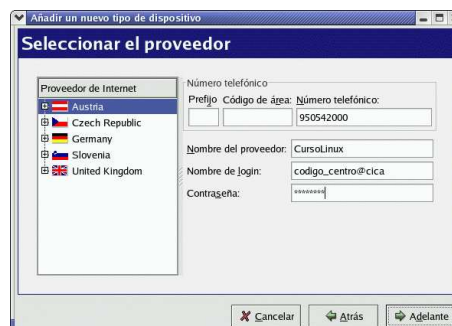


En la ventana que aparece seleccionamos **Conexión del módem** y, al pulsar sobre el botón **Adelante**, comenzará el proceso de autodetección del módem<sup>7</sup>. Si lo autodetecta correctamente aparecerá la pantalla:



Puede ocurrir que no detecte el módem, en este caso aparecerá un mensaje de advertencia. Si nos ocurre esto, podemos usar el programa `minicom` (Véase en la página 53) para, una vez seguros de que funciona bien y del puerto en el que está conectado, optar por introducirlo de forma manual.

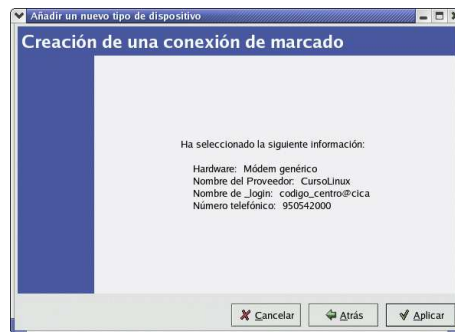
El paso siguiente es configurar la conexión de acceso a partir de los datos de nuestro servidor. En esta pantalla introduciremos el número de teléfono del nodo local al que llamar (por ejemplo 950542000), el nombre que le vamos a dar a esta conexión<sup>8</sup>, el nombre de usuario de nuestra conexión a Internet y la contraseña de acceso:



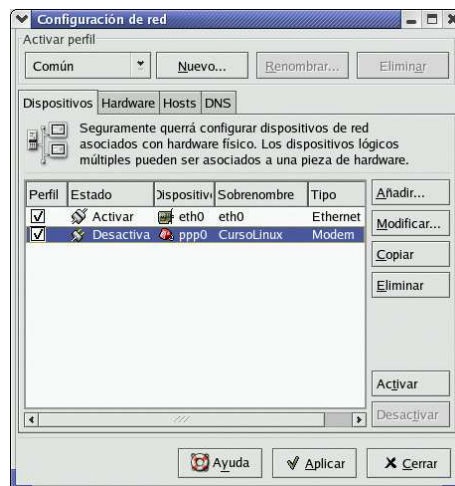
Si continuamos, aparecerá la ventana:

<sup>7</sup>El programa `wvdial` es el encargado de “chequear” los puertos.

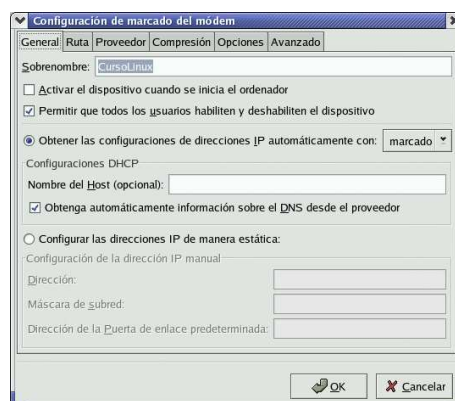
<sup>8</sup>En estos apuntes la llamaremos `CursoLinux`



solicitándonos confirmación sobre los datos introducidos y, por último, la que indica que ya hemos terminado el proceso



Mediante el botón **Añadir** podemos añadir cuantas cuentas de acceso a Internet tengamos a nuestra disposición. El resto son de uso común y sólo comentaremos un poco (por ahora) las posibilidades que se nos brindan al ejecutar **Modificar**

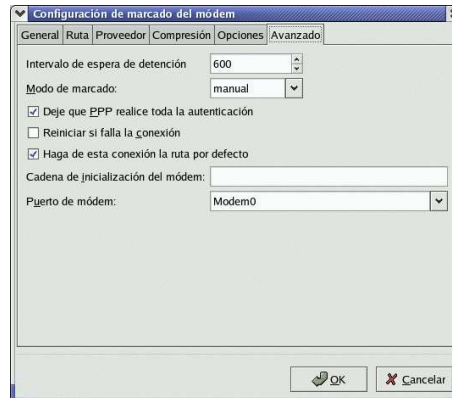


De esta ventana lo interesante es que permite que activemos la posibilidad de que todos los usuarios puedan conectar a internet (opción por defecto) y que debemos dejar activa la casilla de obtener automáticamente información sobre el DNS desde el proveedor<sup>9</sup>.

<sup>9</sup>Si optamos por hacerlo manualmente: véase 4.2.3 en la página 54



Desde la pestaña correspondiente a **Proveedor** podemos modificar los datos relativos a nombre de usuario, contraseña, etc. Por último, en la ventana a la que se accede en **Avanzado** hay que dejar activas las opciones que aparecen marcadas y pulsar **OK**



Ahora vamos a testear que nuestra conexión funciona como debe. En la ventana final de configuración hay que pulsar sobre **Activar** (se nos pedirá confirmación de guardar los cambios) y se inicia el proceso de activación del dispositivo de red *ppp0*. Si todo ha ido bien, en el campo **Estado** aparecerá la palabra **Activar**.

Listo, ya podemos comprobar con Mozilla (🦊) o el navegador que más nos guste que podemos navegar por la red. Para desconectar de internet sólo tenemos que **Desactivar** el dispositivo.

Ya hemos configurado la conexión a internet. A partir de ahora cuando queramos conectarnos usaremos el programa `redhat-control-network` o con menús:



→ **Herramientas del sistema** → **Control de dispositivos de red**

Desde esta ventana podemos Activar/Desactivar la conexión a internet así como cualquier otro dispositivo de red de nuestra máquina.



### 4.2.3. Pistas para detectar problemas

#### Puertos serie en Linux

Disponemos de un comando que nos permite configurar el puerto serie, se trata del comando `setserial`. Para conocer cómo trabajar con él podemos ejecutar (desde un terminal de órdenes):

```
$setserial --help
```

para obtener una ayuda (véase el capítulo 5) básica de los parámetros que admite o:

```
$man setserial
```

para obtener la ayuda completa sobre el programa<sup>10</sup>.

<sup>10</sup>Se sale de la ayuda con `q`.

Para conocer el estado de un puerto serie podemos ejecutar (como root)

```
#setserial -a /dev/ttySx
```

donde `ttySx` es el correspondiente al puerto de comunicaciones del DOS según el esquema

Linux	MS-DOS
ttyS0	COM1
ttyS1	COM2
ttyS2	COM3
ttyS3	COM4

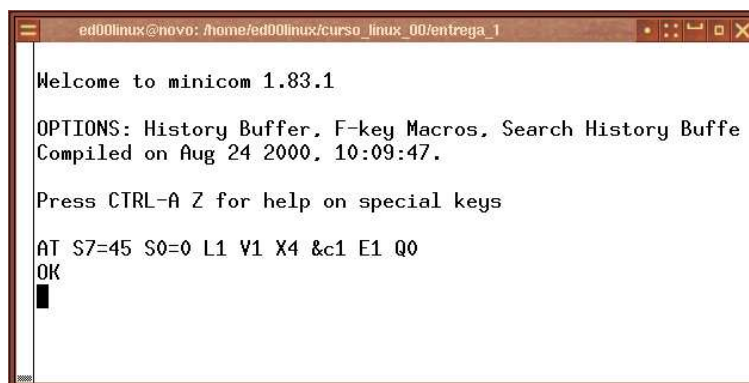
### Minicom

Disponemos de una utilidad que en caso de tener problemas con la configuración del módem nos puede ayudar a detectar en dónde puede estar el problema, se trata del programa `minicom`<sup>11</sup>. Con él podemos comprobar si el módem está bien conectado.

Para activarlo tenemos que ejecutar desde un terminal gráfico el comando:

```
# minicom
```

Si al cargar el programa aparece una pantalla similar a la que sigue, y siempre que nos aparezca el **OK** final, es que todo ha ido bien. Si no es así habrá que reconfigurar la conexión del módem y volver a comprobarlo.



Para acceder al menú de este programa hay que utilizar la combinación de teclas `CTRL+A` y después pulsar la letra `Z`. Por ejemplo, para salir del programa hay que pulsar `CTRL+A`, después `Z` y por último `Q`.

### Conocer IP asignada

Si deseamos comprobar si la conexión se produce con éxito y conocer la dirección IP que se nos ha asignado dinámicamente podemos escribir desde un terminal la orden<sup>12</sup>

```
#tail -f /var/log/messages
```

veremos entonces una serie de mensajes que nos muestran cual es el estado de la conexión, si ésta ha tenido éxito nos tienen que aparecer dos líneas del tipo:

```
local IP address xxx.xxx.xxx.xxx
remote IP address xxx.xxx.xxx.xxx
```

<sup>11</sup>Programa terminal de comunicaciones

<sup>12</sup>También podemos ver esto mismo con:

```
$ /sbin/ifconfig
```

donde esos números indican las direcciones IP asignadas dinámicamente a nuestra máquina y al servidor. Para cancelar el comando `tail` y dejar de visualizar las líneas que van saliendo hay que pulsar `CTRL+C`.


Una forma de saber si realmente hemos conectado bien, es hacer un *ping*. Este comando comprueba que llegamos a la máquina remota que queremos comprobar. Por ejemplo, `$ping 150.214.5.11`, para llegar al servidor de los cursos. El comando nos dirá si llegamos o devuelve error.

```
$ping 150.214.5.11
PING 150.214.5.11 (150.214.5.11) from 195.24.23.44 : 56(84) bytes of data.
64 bytes from 150.214.5.11: icmp_seq=1 ttl=64 time=0.040 ms
64 bytes from 150.214.5.11: icmp_seq=2 ttl=64 time=0.037 ms
64 bytes from 150.214.5.11: icmp_seq=3 ttl=64 time=0.035 ms
64 bytes from 150.214.5.11: icmp_seq=4 ttl=64 time=0.039 ms
--- 150.214.5.11 ping statistics ---
4 packets transmitted, 4 received, 0% loss, time 3006ms
rtt min/avg/max/mdev = 0.035/0.037/0.040/0.007 ms
Para terminar pulsamos [Ctrl]+[c]
```

### Conectamos pero no salimos fuera

Si pese a que conectamos no podemos visualizar páginas web, revisar:

- El fichero `/etc/resolv.conf`. Puede que pese a tener marcada la opción de que obtengamos la información sobre los DNS desde el proveedor esto no sea así. En este caso lo mejor es que configuremos esto de forma manual. Podemos conseguirlo de dos formas:

- Gráfica:  → **Configuración del sistema** → **Red**<sup>13</sup>. Pestaña **DNS** e introducimos las IPs de nuestros servidores de nombres:



- Texto: Utilizando un editor de textos escribiremos en el fichero:

```
domain cica.es
nameserver 150.214.3.9
nameserver 150.214.90.11
```

cambiando los datos de ejemplo anteriores por los de nuestro servidor de acceso.

<sup>13</sup>También podemos optar por pulsar en el escritorio sobre el icono de **Empezar aquí** → **Configuración del sistema** → **Red**.

- Si tenemos una tarjeta de red, revisar la salida del comando

```
#netstat -ar
```

si sale una línea del tipo

```
default 192.168.0.254
```

u otra IP local, es que hemos configurado como *Gateway* una máquina local. Debemos eliminarla ejecutando `neat` y después **Dispositivos(eth0)→Modificar** y eliminar la dirección IP de la puerta de enlace predeterminada.

#### 4.2.4. Conexión mediante Script

En Linux cualquier cosa que hagamos en modo gráfico se puede hacer también en modo texto (y con un mayor control por nuestra parte sobre lo que estamos haciendo).

Para aquellos que no hayan podido configurar la conexión a internet de forma gráfica o que prefieran hacerla en modo texto, existe posibilidad (quizá mejor) de realizar nosotros los ficheros de conexión. Para ello tendremos que crear o modificar los archivos:

- `/etc/ppp/options`
- `/etc/ppp/conecta`
- `/etc/ppp/chap-secrets (/etc/ppp/pap-secrets)`
- `/etc/ppp/peers/servidor`
- `/etc/resolv.conf`

A continuación facilitamos unos ficheros de ejemplo. Si se quieren mejorar estos Script, el propio programa `pppd` trae documentación y ejemplos que permiten automatizar y mejorar la conexión.

Para crear/modificar estos ficheros podemos usar cualquiera de los editores de GNOME/KDE (modo gráfico), el programa `mcedit` (Capítulo 6) o cualquiera de los editores que acompañan a cualquier distribución de Linux (`pico`, `vi`, ...)

##### `etc/ppp/options`

El contenido de este fichero es la serie de líneas que aparecen a continuación. Las líneas precedidas por `#` son comentarios que no es necesario escribir y cuya función es saber qué finalidad tiene el comando que sigue a esa línea. De algunos comandos se puede prescindir (por ejemplo: `noipx`, `debug`).

```
#activa el control de flujo por hardware
rtscts
#indicamos que vamos a usar un módem (línea telefónica)
modem
#si no recibe respuesta en una sesión debe esperar a recibir datos válidos
passive
#para que la conexión se realice a través de una asignación dinámica de
IP
noipdefault
#para que no aparezca la línea Unsupported protocol (0x031)received
noipx
#permite crear un archivo log de lo ocurrido en la conexión
debug
#Configura una ruta predeterminada en el sistema y la utiliza como puerta
de enlace.
defaultroute
```

```
#selecciona la negociación de los caracteres de control
asynmap a0000
#dispositivo a usar, también se puede poner /dev/modem si existe el enlace
/dev/ttySx
#no es la velocidad del módem, es la velocidad entre el módem y el
#puerto serie, si el ordenador es antiguo quizá haya que poner 57600
115200
```

### **/etc/ppp/conecta**

En este fichero hemos puesto una cadena de inicialización del módem estándar (AT& F). Puede que para vuestro módem haya que poner otra (en la documentación del módem viene):

```
ABORT "BUSY"
ABORT "NO DIALTONE"
ABORT "NO CARRIER"
ABORT "ERROR"
""
"AT& F" TIMEOUT 5 OK
"ATDTWteléfono" TIMEOUT 100 CONNECT
```

donde lo único que hay que sustituir es *teléfono* por el número que corresponda.

Este fichero tiene que ser ejecutable así que desde un terminal escribiremos

```
# chmod 755 /etc/ppp/conecta
```

### **/etc/ppp/chap-secrets (o pap-secrets)**

Tendrás que editar uno u otro dependiendo del método de autenticación. Primero prueba con CHAP y si no funciona inténtalo con PAP. Ambos ficheros son de la forma:

```
usuario@servidor      *      contraseña
```

donde lo único a reseñar es que los campos van separados con tabuladores.

Es conveniente proteger este archivo de “intrusos”, para esto ejecutamos desde un terminal:

```
# chmod 600 /etc/ppp/chap-secrets
```

### **/etc/ppp/peers/servidor**

Sólo contiene dos líneas, la segunda tendremos que adaptarla a nuestra conexión:

```
connect "/usr/sbin/chat -v -f /etc/ppp/conecta"
name usuario@servidor
```

### **/etc/resolv.conf**

Este fichero se puede generar a partir del panel de control, de igual manera que vimos al inicio del punto 4.2.2, o bien utilizando un editor de textos (gráfico como gedit, o en modo carácter como pico, emacs). Escribiremos en él:

```
domain servidor.es
nameserver xxx.xxx.xxx.xxx
nameserver xxx.xxx.xxx.xxx14
```

---

<sup>14</sup>Sustituir *servidor.es* por el nuestro. Si no tenemos DNS secundario, sobra la segunda línea

Una vez que tengamos todos los ficheros creados, para conectar escribiremos desde un terminal la orden  
pppd call servidor  
y para desconectar pondremos  
killall pppd

Si queremos monitorizar el estado de la conexión es mejor escribir  
pppd call servidor;tail -f /var/log/messages  
y para terminar, CTRL+C, y después  
killall pppd

### 4.3. RDSI

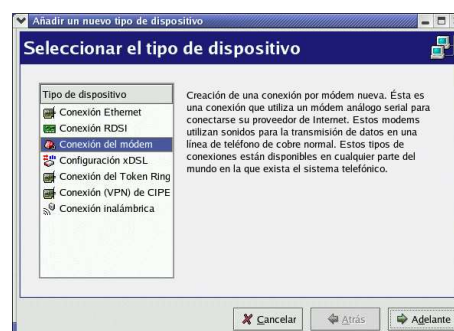
En este apartado vamos a intentar dar unas pinceladas de cómo conectar a Internet usando una línea RDSI y la tarjeta interna Novacom Micro<sup>15</sup>. Para el resto de tarjetas os remitimos al Cómo RDSI-Como, a los ficheros que se instalan con las fuentes del núcleo en

/usr/src/linux-2.4/Documentation/isdn

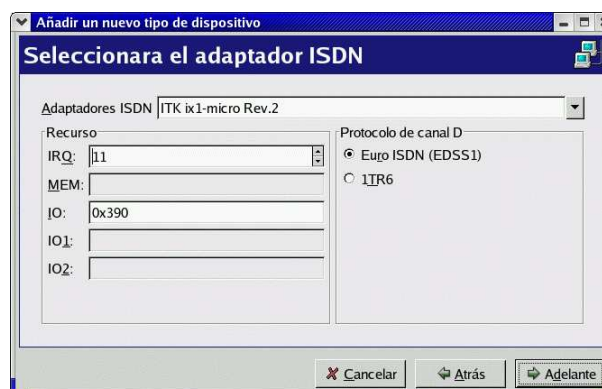
así como a las páginas antes comentadas sobre compatibilidad con el Hardware.

Desde un terminal gráfico ejecutamos el programa internet-druid (👉 → **Herramientas del sistema**→**Wizard de configuración de Internet**)

#internet-druid



Seleccionamos **Conexión RDSI**. Después elegiremos el modelo de tarjeta, en este ejemplo es una ITK, y le modificamos los valores que aparecen por defecto (IRQ 9 e IO 0xd80) a los adecuados para nuestra tarjeta (en este caso IRQ 11 e IO 0x390).



<sup>15</sup>Para el teléfono Novacom Micro mirar en: <http://roble.pntic.mec.es/~sgonzale/linux/novacom.html>

Cuando terminemos podemos modificar nuestra conexión ejecutando:

```
#neat
```

Las opciones y el modo de conectar son similares a las ya vistas en la conexión usando un módem y a ellas os remitimos (Páginas 52 a 52).


## 4.4. ADSL

No podemos contrastar que la utilidad `internet-druid` funcione correctamente ni podemos ejemplificar todas las posibilidades que existen actualmente en España para esta modalidad de conexión.

Sólo podemos ofrecer una guía para aquellas configuraciones que hemos contrastado que funcionan. Se trata de conexiones ADSL usando Telefónica como proveedor de acceso y módem router que no están basados en tecnología USB.

Vamos a suponer que disponemos de una tarjeta de red a la que tenemos conectado el módem. Además, lo más usual es que al instalar linux se detecte la tarjeta de red y que la tengamos ya configurada correctamente. Si no es así podemos revisar con las indicaciones que siguen en dónde está el error.

Si nuestro modelo es USB debemos mirar los enlaces e información que sobre este tema podemos encontrar en <http://www.escomposlinux.org/hardware>. También puede ser de ayuda la página <http://www.adsl4ever.com/>.

Para configurar nuestra conexión ADSL en los supuestos antes comentados usaremos  → **Configuración del sistema** → **Red** o bien desde un terminal gráfico:

```
#redhat-config-network
```

```
o
```

```
#neat
```

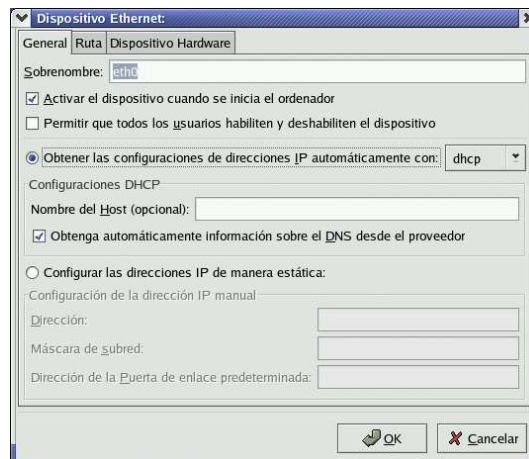


Pulsemos sobre **Modificar** y en la pestaña **General** configuraremos nuestra conexión ADSL en función de cómo tengamos el módem router configurado:

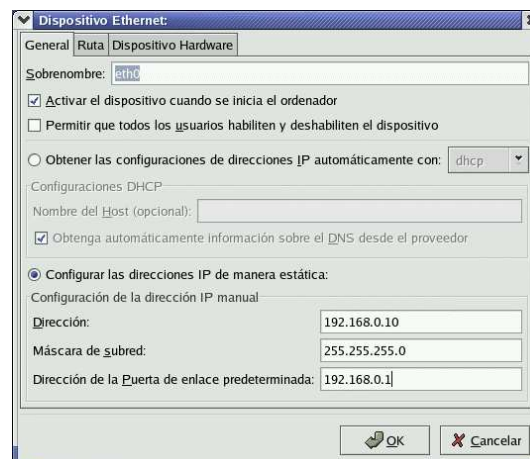
**Monopuesto** <sup>16</sup>sólo hay que decirle a la tarjeta de red que obtenga la configuración de direcciones IP automáticamente con DHCP y marcar la opción de obtener automáticamente información sobre el DNS desde el proveedor.

<sup>16</sup>Sin NAT (*Network Address Translation*) y con DHCP (*Dinamic Host Control Protocol*). Se trata de que la dirección pública del router la asignamos al interfaz de red. En este caso no podemos tener varios equipos conectados directamente al router.

Si disponemos de dos tarjetas de red en nuestra máquina mejor si ésta es la `eth0`. La que primero detecta el sistema. La otra Linux la verá como `eth1`.



**Multipuesto** <sup>17</sup>en este caso tendremos que poner una dirección IP a nuestro interfaz de red de la misma red que el router (este dato lo tenemos que conocer a través del proveedor de acceso). Así, por ejemplo, si nuestro módem router tiene la IP local 192.168.0.1 y como máscara de red 255.255.255.0 sólo le diremos al interfaz de red que use como Gateway la IP local del router y las IP de los servidores de nombres. Si optamos por poner de IP a nuestra máquina linux la dirección 192.168.0.10, quedaría:

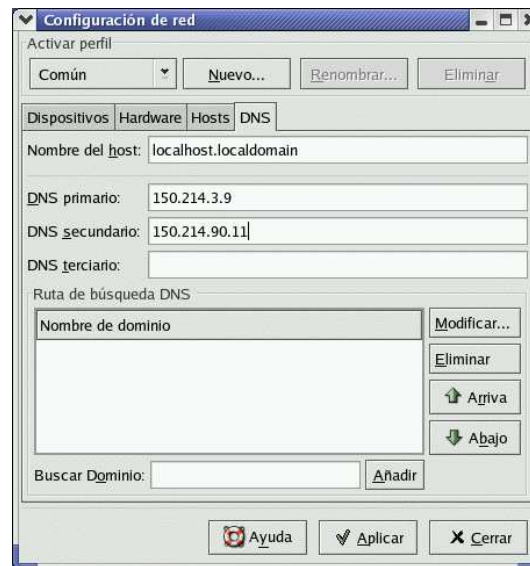


y para añadir los servidores de nombres (fichero `/etc/resolv.conf`), en la ventana principal de `neat` pulsaremos sobre DNS y

---

<sup>17</sup>Con NAT y sin DHCP.






En los dos casos, para que los cambios sean efectivos habrá que **Aplicar** los cambios y **Activar**<sup>18</sup> el interfaz de red en



Para saber si todo está bien podemos:

- Abrir Mozilla (  ) y comprobar si salimos fuera.
- Hacer un ping a una máquina remota<sup>19,20</sup>:

<sup>18</sup>Si ya estaba activo antes hay que **Desactivarlo**.

<sup>19</sup>La diferencia con el anterior comando ping es que ahora estamos intentando llegar a la máquina remota por su nombre DNS y no por su dirección IP. El mecanismo de resolución de nombres de DNS debe funcionar correctamente para que funcione.

<sup>20</sup>Para cancelar el comando:

[Ctrl] + [C]

```
$ping mileto.cica.es
PING mileto.cica.es (150.214.5.11) from 80.30.154.77 : 56(84) bytes
of data.
64 bytes from mileto.cica.es (150.214.5.11): icmp_seq=1 ttl=53 time=101
ms
64 bytes from mileto.cica.es (150.214.5.11): icmp_seq=2 ttl=53 time=97.3
ms
64 bytes from mileto.cica.es (150.214.5.11): icmp_seq=3 ttl=53 time=113
ms
64 bytes from mileto.cica.es (150.214.5.11): icmp_seq=4 ttl=53 time=93.8
ms
64 bytes from mileto.cica.es (150.214.5.11): icmp_seq=5 ttl=53 time=101
ms
--- mileto.cica.es ping statistics ---
5 packets transmitted, 5 received, 0% loss, time 4033ms
rtt min/avg/max/mdev = 93.849/101.676/113.561/6.657 ms
```

## Capítulo 5

# Información en línea

La documentación libre, como el software libre, es una cuestión de libertad, no de precio.

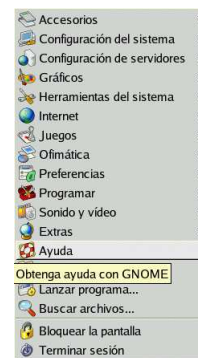
*The GNU Operating System and the Free Software Movement*, RICHARD STALLMAN

### 5.1. Introducción

La mayoría de las aplicaciones con las que trabajamos en Linux tienen suficiente información como para que podamos resolver cualquier duda que se nos plantee. Si tenemos GNOME/KDE en funcionamiento, disponemos de una magnífica ayuda que nos permite poder iniciarnos con las aplicaciones de estos entornos de escritorio.

En Linux, además de lo comentado anteriormente, la documentación se nos brinda de varias formas:

1. Páginas de manual o *manpages*.
2. Páginas de información o *info-pages*.
3. Documentación que se instala a la vez que la aplicación.



En casi todos los comandos de Linux tenemos a nuestra disposición una mini-ayuda en línea a la que se accede ejecutando

```
orden --help
```

#### ➡ Para practicar

1. A veces la ayuda disponible no cabe en una pantalla es conveniente usar una tubería (`|`) para que la salida del primer comando sea pasada al filtro `less`<sup>1</sup>:

a) Ejecuta

```
$ls --help
```

y comprueba que la información se nos sale de la pantalla <sup>2</sup>.

<sup>1</sup>`less` es un filtro que permite ver la información página a página con la ventaja añadida de que podemos retroceder en la información. También se puede usar `more` en vez de `less`, la diferencia está en que `more` permite ver la información página a página pero no echar hacia atrás.

<sup>2</sup>Bajo un `xterm` esto no representa ninguna dificultad ya que podemos usar la barra de desplazamiento vertical. En modo texto tampoco es un gran problema porque podemos acceder a la información no visible de la parte superior de la pantalla con la combinación de teclas

[Shift] + [AvPag]

si queremos “bajar” pulsaremos

[Shift] + [RePag]

- b) Si usamos  
\$ ls --help | less  
obtendremos

```
Modo de empleo: ls [OPCIÓN]... [FICHERO]...
Muestra información acerca de los FICHEROS (del directorio actual por defecto).
Ordena las entradas alfabéticamente si no se especifica ninguna de las
opciones -cftuSUX ni --sort.

Los argumentos obligatorios para las opciones largas son también obligatorios
para las opciones cortas
-a, --all          do not hide entries starting with .
-A, --almost-all do not list implied . and ..
--author          print the author of each file
-b, --escape      print octal escapes for nongraphic characters
--block-size=TAMAÑO utiliza bloques de TAMAÑO bytes
-B, --ignore-backups no muestra la entradas que terminan con ~
-c               con -lt: ordena por ctime y muestra ctime (fecha
                de última modificación del fichero)
                con -l: muestra ctime y ordena por nombre
:
```

que es más fácil de visualizar.

## 5.2. manpages

Además de la completa ayuda disponible con GNOME, veamos que todavía podemos afinar más.

En Linux tenemos una gran cantidad de información en línea sobre él mismo si usamos la orden `man`<sup>3</sup>. La sintaxis de esta orden es<sup>4</sup>:

```
man [-acdfFhkKtW] [-m system] [-p string] [-C con fig_file]
[-M path] [-P pager] [-S section_list] [section] name ...
```

El objetivo aquí no es explicar todas y cada una de las posibilidades de esta orden sino sólo aquello que nos permita en esta entrega continuar con buen pie el desarrollo del curso. En general la orden `man` casi siempre la usaremos como sigue:

### ■ man orden

al ejecutar este comando se nos mostrará una página de información sobre esa orden, para desplazarnos por la ayuda tan sólo tenemos que hacer uso de los cursores o de la barra espaciadora si lo que queremos es pasar a la página siguiente. Para buscar una palabra en el documento hay que usar:

/palabra

Si queremos continuar la búsqueda de esa palabra pulsaremos sobre la tecla **n**.

Para salir de la página hay que pulsar **q**. Por ejemplo, si ejecutamos:

```
$ man man
se obtendría
```

<sup>3</sup>Las manpages traducidas al castellano están en.

<http://ditec.um.es/~piernas/manpages-es/>

<sup>4</sup>Tal cual aparece si escribimos

```
$ man man
```

Para profundizar sobre el uso de `man` lo mejor es hacer uso de la orden anterior.

```
man(1) man(1)
NAME
    man - format and display the on-line manual pages
    manpath - determine user's search path for man pages
SYNOPSIS
    man [-acdfhkkW] [--path] [-m system] [-p string] [-C config_file]
    [-M pathlist] [-P pager] [-S section_list] [section] name ...
DESCRIPTION
    man formats and displays the on-line manual pages. If you specify sec-
tion, man only looks in that section of the manual. name is normally
    the name of the manual page, which is typically the name of a command,
    function, or file. However, if name contains a slash (/) then man
    interprets it as a file specification, so that you can do man ./foo.5
:
```

#### ■ man -k cadena

si no sabemos con seguridad qué estamos buscando podemos decirle al programa que busque la cadena objeto de nuestro interés. Al introducir el parámetro `-k` a la orden `man`, ésta busca en todos los archivos de ayuda todas las órdenes que contengan esa cadena y las muestra en pantalla. Después, usando la sintaxis anterior podemos visualizar la manpage deseada.

Por ejemplo, si queremos buscar sobre comandos relacionados con internet podemos escribir:

```
$ man -k internet5
```

el resultado sería

```
ftp (1) - Internet file transfer program
ftpd (8) - Internet File Transfer Protocol server
ftp [pftp] (1) - Internet file transfer program
IMAPd [imapd] (8c) - Internet Message Access Protocol server
inet_addr [inet] (3) - Internet address manipulation routines
inet_addr [inet] (3) - Rutinas de manipulación de direcciones de Internet
inet_aton [inet] (3) - Internet address manipulation routines
inet_aton [inet] (3) - Rutinas de manipulación de direcciones de Internet
inet_lnaof [inet] (3) - Internet address manipulation routines
inet_lnaof [inet] (3) - Rutinas de manipulación de direcciones de Internet
inet_makeaddr [inet] (3) - Internet address manipulation routines
inet_makeaddr [inet] (3) - Rutinas de manipulación de direcciones de Internet
inet_netof [inet] (3) - Internet address manipulation routines
inet_netof [inet] (3) - Rutinas de manipulación de direcciones de Internet
inet_network [inet] (3) - Internet address manipulation routines
inet_network [inet] (3) - Rutinas de manipulación de direcciones de Internet
inet_ntoa [inet] (3) - Internet address manipulation routines
:
```

Si ahora deseamos información sobre una orden concreta, como por ejemplo `pine`, escribiremos:

```
$man pine
```

#### ➡ Para practicar.

1. Visualizar la página man del comando `man`
2. A veces puede ser interesante imprimir una página man con el formato adecuado. Por ejemplo, veamos como imprimir la página man del comando `man`.

<sup>5</sup>Esta orden es la misma que si escribimos

```
$ apropos internet
```

- a) Desde un terminal gráfico ejecutemos:

```
$man -t man >man.ps
```

Donde:

**-t** con este parámetro, usamos el comando `/usr/bin/groff` con los parámetros adecuados para obtener un fichero PostScript listo para imprimir.

> canalizamos la salida a un fichero de nombre `man.ps`

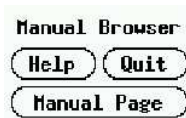
- b) Visualicemos/imprimamos el contenido del fichero generado ejecutando desde un terminal gráfico<sup>6</sup>:

```
$ggv man.ps
```

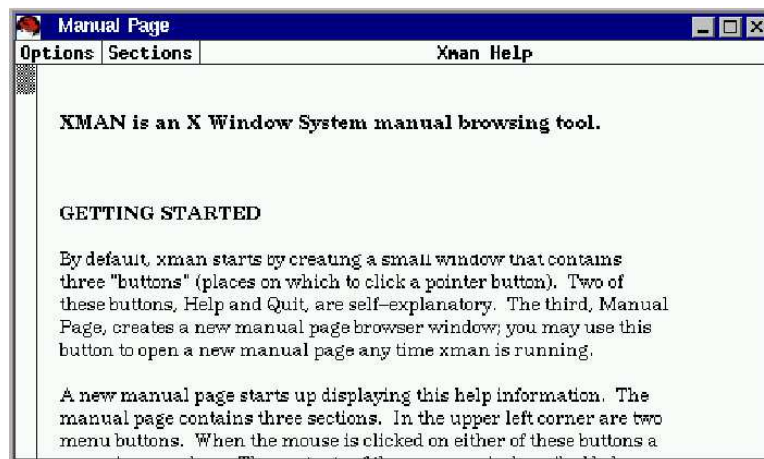
También disponemos de ayuda (además de la de GNOME) en modo gráfico, para acceder a ella tan solo hay que ejecutar desde una xterm la orden

```
$xman
```

se nos mostrará




y si pulsamos sobre **Manual Page** accederemos a



Su utilización es sencilla y la mejor forma de comprenderlo es poniéndose con él. El manual de ayuda está dividido en secciones:

- (1) User Commands
- (2) System Calls
- (3) Subroutines
- (4) Devices
- (5) File Formats
- (6) Games
- (7) Miscellaneous
- (8) Sys. Administration
- (n) New

<sup>6</sup>Si optamos por los menús:

 → Extras → Gráficos → PostScript Viewer

esta clasificación es válida también para las páginas man en modo texto, con esta clasificación en secciones podemos “complicar” un poco la sintaxis de la orden man:

```
man [sección] orden
```

1. Con esta ampliación sobre la sintaxis podemos acceder a información sobre órdenes con igual nombre pero que están en distintas secciones.

a) Por ejemplo si ejecutamos

```
$man 1 passwd
```

obtendremos el manual de la orden passwd (1 es **User Commands**)

```
PASSWD(1)                                User utilities                                PASSWD(1)

NAME
    passwd - update a user's authentication tokens(s)

SYNOPSIS
    passwd [-k] [-l] [-u [-f]] [-d] [-S] [username]

DESCRIPTION
    Passwd is used to update a user's authentication token(s).

    Passwd is configured to work through the Linux-PAM API. Essentially,
    it initializes itself as a "passwd" service with Linux-PAM and utilizes
    configured password modules to authenticate and then update a user's
    password.

:
```

b) mientras que si usamos

```
$man 5 passwd
```

accederemos al manual sobre el archivo de contraseñas (5 es **File Formats**)

```
PASSWD(5)                                Formatos de Fichero                                PASSWD(5)

NOMBRE
    passwd - fichero de contraseñas

DESCRIPCIÓN
    Passwd es un fichero de texto que contiene una lista de las cuentas del
    sistema, proporcionando para cada cuenta cierta información útil como
    el identificador (ID) de usuario, el ID de grupo, el directorio 'home',
    el intérprete de comandos, etc. Con frecuencia, también contiene la
    contraseña cifrada de cada cuenta. Este fichero debe tener permiso de
    lectura para todos (muchas utilidades, como ls(1), lo usan para tra-
    ducir el número de identificador de usuario (UID) al nombre del
    usuario), pero sólo el superusuario debe poder escribirlo.

    En los buenos viejos tiempos no había grandes problemas con estos per-

:
```

### 5.3. info-pages

Con el comando `info` podemos acceder a la ayuda en línea de cualquier comando de Linux. Igual que con la orden `man`, su sintaxis es bastante completa<sup>7</sup> y nos vamos a centrar en los dos modos más usuales:

---

<sup>7</sup>Basta ejecutar  
`man info`  
para comprobarlo



- `info` muestra una pantalla con los comandos de los que disponemos ayuda.
- `info comando` muestra información sobre el comando en cuestión.

## **5.4. /usr/share/doc (/usr/doc)**

En este directorio se sitúa documentación sobre muchos de los programas que instalemos READMEs, Howtos y manuales<sup>8</sup>. Antes de buscar información en Internet o en libros no está de más echar un vistazo por él para ver si nuestras dudas se pueden resolver con los ficheros que hay allí.

Con respecto a este tipo de información lo mejor es que os “divertáis” un rato mirando qué enorme cantidad de información tenéis ya a vuestra disposición.

---

<sup>8</sup>Desde la versión 7.0 los sistemas de directorios han cambiado para conseguir mayor compatibilidad con el FHS. En versiones anteriores la documentación/ayuda colgaba de `/usr` mientras que ahora lo hace de `/usr/share`



## Capítulo 6

# Apéndice: Midnight Commander

Este administrador de ficheros para Linux constituye una herramienta única para facilitarnos y, sobre todo, para acelerar todas las operaciones que diariamente debemos realizar con nuestro ordenador. Posiblemente se encuentre entre los programas más útiles que podemos encontrar para cualquier sistema operativo y más concretamente para Linux. (*Midnight Commander. Sólo Programadores Linux nº 7*. DAVID ESPADA GARCÍA)

### 6.1. Introducción

Si trabajamos en modo texto o desde una `xterm`, tenemos a nuestra disposición un programa (tipo Comandante Norton) que es de un valor inestimable para movernos por el sistema de ficheros de un sistema Linux, se trata de:

**mc**<sup>1</sup> Entorno visual para sistemas tipo Unix. *Midnight Commander* es un navegador de directorios/gestor de ficheros para sistemas operativos tipo Unix.



En este capítulo aparecen multitud de conceptos nuevos que, por ahora, no es necesario estudiar y que no son fundamentales para entender la funcionalidad básica del programa. Muchos de estos conceptos serán el objetivo del curso y se tratarán más ampliamente en temas posteriores. Unos posibles “contenidos mínimos” de este capítulo serían la sección 6.1.1 de Instalación así como las prácticas de autoevaluación de la sección 6.4.5 en la página 85

#### 6.1.1. Instalación

Si bien RedHat no instala este programa por defecto<sup>2</sup>, hemos optado por introducirlo en la primera entrega porque consideramos que es una utilidad que puede sernos de inestimable ayuda (fundamental si no conseguimos acceder inicialmente en modo gráfico).

Como su mayor interés radica en cuando se usa en modo texto, vamos a guiar sobre su instalación<sup>3</sup> suponiendo que no disponemos de entorno gráfico. Iniciaremos una sesión como root y, tras introducir el tercer CD de la distribución, ejecutaremos los comandos:

```
#mount /mnt/cdrom
```

para montar la unidad de CD y

```
#rpm -ivh /mnt/cdrom/RedHat/RPMS/mc-4.5.55-12.i386.rpm
```

para instalar el programa que se encuentra en la ruta especificada.

<sup>1</sup> Este documento se basa de forma casi literal en la página `man` del programa `mc`, lo único que se ha hecho es “resumir” y actualizar algunos aspectos de dicha página y añadir gráficos explicativos del programa. Para una mayor información sobre el programa os remitimos a dicha página de ayuda.

<sup>2</sup> Para saber si está instalado podemos ejecutar `$mc` y ver si se ejecuta el programa o aparece un error.

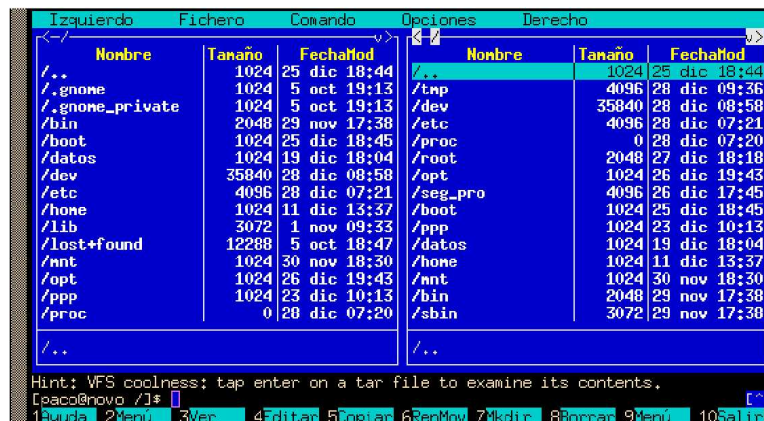
<sup>3</sup> Se puede hacer esto mismo en modo gráfico, pero la forma de hacerlo se dejará para entregas posteriores

### 6.1.2. Inicio de una sesión

Lo ejecutaremos con la orden:

```
$ mc
```

La pantalla de *Midnight Commander* está dividida en cuatro partes. La mayor parte del espacio de la pantalla se usa para los dos paneles de directorio. Por defecto, la segunda línea más inferior de la pantalla es la línea de comandos del shell, y la línea inferior muestra las etiquetas de las teclas de función.



La línea superior es la línea de la barra de menú. Podemos activar la barra de menú si pulsamos en la primera línea de la pantalla con el ratón o pulsamos la tecla **F9**.

El *Midnight Commander* provee una vista de dos directorios al mismo tiempo. Uno de los paneles es el panel actual (una barra de selección está en el panel actual). La mayoría de las operaciones tienen lugar en el panel actual. Algunas operaciones con ficheros como renombrar y copiar por defecto utilizan el directorio del panel de-seleccionado como destino.

Podemos ejecutar comandos del sistema desde el *Midnight Commander* simplemente escribiéndolos. Todo lo que escribamos aparecerá en la línea de comandos del shell y cuando pulsemos **Enter**, *Midnight Commander* ejecutará la línea de comandos.

Pulsando con el ratón sobre las teclas de función o usando las teclas **F1-F10**<sup>4</sup> tenemos posibilidad de ejecutar las operaciones más comunes.

### 6.1.3. Soporte de Ratón

*Midnight Commander* viene con soporte de ratón. Cuando pulsamos el botón izquierdo del ratón sobre un fichero, ese fichero es seleccionado; si lo hacemos con el botón derecho, el fichero es marcado (o desmarcado, dependiendo del estado previo).

Una doble pulsación sobre un fichero intentará ejecutar el comando si se trata de un programa ejecutable; y si la extensión del fichero tiene un programa asociado para la extensión del fichero, se ejecutará dicho programa.

Además, es posible ejecutar los comandos asignados a las etiquetas de las teclas de función pulsando con el ratón sobre ellas.

Si se pulsa con el botón del ratón en la línea divisoria superior del panel de directorio, se realiza un scroll (desplazamiento) de una página hacia atrás. Asimismo, una pulsación sobre la línea divisoria inferior produce un scroll de una página hacia delante. Este método de la línea divisoria funciona también con el Visor de Ayuda y el Árbol de directorios.

<sup>4</sup>Si ejecutamos `mc` desde un terminal de GNOME nos encontraremos con un problema a la hora de intentar usar la tecla **F10**, ya que se usa en los terminales gráficos para acceder a los menús. Si deseamos poder usarla desde `mc`, pulsaremos en **Editar** → **Combinaciones de teclas** del menú del terminal gráfico y desmarcaremos la casilla correspondiente a:

*Box* Deshabilitar la tecla de acceso a menús (F10 por omisión)

Si estamos ejecutando Commander con el soporte de ratón, podemos obtener el comportamiento por defecto del ratón (cortando y pegando texto) manteniendo pulsada la tecla *Shift*.

#### 6.1.4. Teclas

Algunos comandos en *Midnight Commander* implican el uso de las teclas Control (algunas veces representado por CTRL o CTL) y Meta (algunas veces denominado ALT o incluso Compose). En este manual<sup>5</sup> usaremos las siguientes abreviaturas:

**C-<chr>** significa mantener pulsada la tecla Control mientras pulsamos el carácter <chr>. Así C-f sería: manteniendo pulsada la tecla Control teclear f.

**M-<chr>** significa mantener pulsada la tecla Meta o Alt mientras pulsamos el carácter <chr>.

Para una descripción detallada de las combinaciones de teclas véase la página man del programa, aquí sólo vamos a comentar aquellas más “importantes” que impliquen acciones que no son realizables usando los menús del programa.

Merece la pena comentar que los cursores y las teclas Inicio, Fin, etc realizan las labores “usuales” a las que estamos habituados.

Comentemos algunas de ellas:

**Enter.** Si hay algún texto en la línea de comandos (la de la parte inferior de los paneles), entonces se ejecuta ese comando. Si no hay texto en la línea de comandos y la barra de selección está situada sobre un directorio, entonces *Midnight Commander* realiza un *chdir* al directorio seleccionado y recarga la información en el panel; si la selección es un fichero ejecutable entonces es ejecutado. Por último, si la extensión del fichero seleccionado coincide con una de las extensiones en el fichero de extensiones entonces se ejecuta el comando correspondiente.

**Tab** Cambia el panel actual. El panel activo deja de serlo y el no activo pasa a ser el nuevo panel activo. La barra de selección se mueve del antiguo panel al nuevo, desaparece de aquél y aparece en éste.

**Insertar** Para marcar ficheros (y/o directorios) como seleccionados podemos usar la tecla Insertar (Ins). Para deseleccionar ficheros, basta con repetir la operación sobre los ficheros y/o directorios antes marcados.

+ (más) Usado para seleccionar (marcar) un grupo de ficheros.

- (menos) Usaremos la tecla - para deseleccionar un grupo de ficheros.

Con respecto a las teclas útiles para evitar la excesiva escritura cuando se introducen comandos del shell, tenemos:

**C-Enter.** Copia el nombre de fichero seleccionado a la línea de comandos.

**M-Tab.** Realiza una Terminación automática (completion) del nombre de fichero, comando, variable, nombre de usuario y host.

**C-x p, C-x C-p.** La primera secuencia de teclas copia el nombre de la ruta de acceso actual a la línea de comandos, y la segunda copia la ruta del otro panel a la línea de comandos.

**C-q.** El comando cita (quote) puede ser utilizado para insertar caracteres que de otro modo serían interpretados por *Midnight Commander* (como el símbolo '+')

**M-p, M-n.** Usaremos esas teclas para navegar a través del histórico de comandos. M-p devuelve la última entrada, M-n devuelve la siguiente.

## 6.2. Barra de Menú

La barra de menú aparece cuando pulsamos F9 o pulsamos el botón del ratón sobre la primera fila de la pantalla. La barra de menú tiene cinco submenús: "Izquierdo", "Fichero", "Comando", "Opciones" y "Derecho".

Izquierdo	Fichero	Comando	Opciones	Derecho
-----------	---------	---------	----------	---------

<sup>5</sup>Sólo en la parte correspondiente al programa mc.

Los Menús Izquierdo y Derecho nos permiten modificar la apariencia de los paneles de directorio izquierdo y derecho.

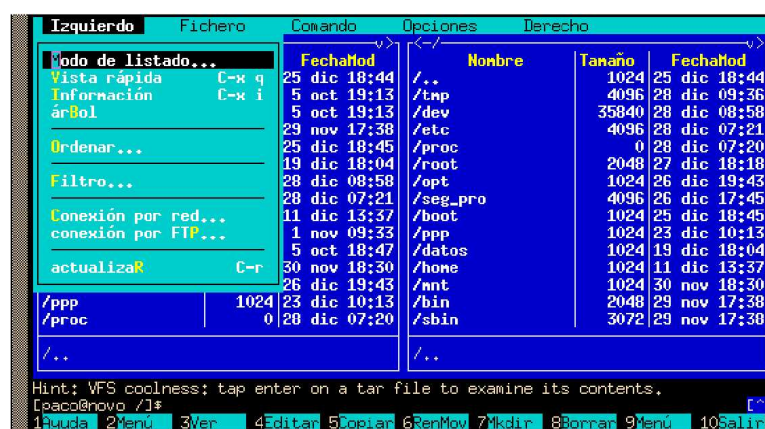
El Menú de Fichero lista las acciones que podemos realizar sobre el fichero actualmente seleccionado o sobre los ficheros marcados.

El Menú de Comandos lista las acciones más generales y que no guardan relación con la selección actual de ficheros.

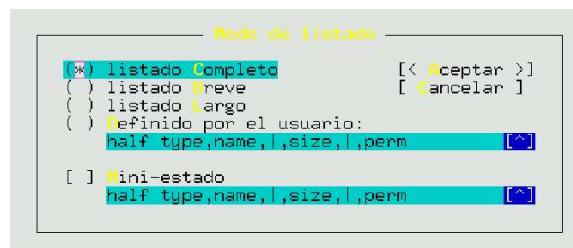
Una vez activo un menú podemos usar la letra marcada con mayúsculas y amarillo para acceder a ese comando del menú.

### 6.2.1. Menús Izquierdo y Derecho

La presentación de los paneles de directorio puede cambiarse desde los menús Izquierdo y Derecho.



**Modo de listado...** El modo de listado<sup>6</sup> se usa para visualizar ficheros y sus atributos. Hay cuatro modos diferentes: Completo, Breve, Largo y Personalizado



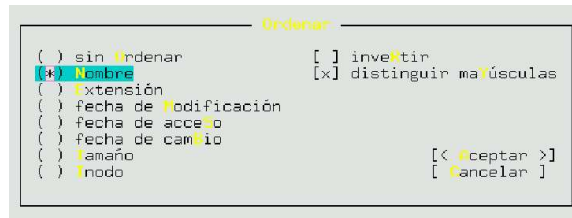
**Vista Rápida** En este modo, el panel cambia a un visor reducido que muestra el contenido del fichero actualmente seleccionado, si seleccionamos el panel (con la tecla tab o el ratón), tendremos acceso a los comandos usuales del visor.

**Información** La vista de información visualiza información relacionada con el fichero seleccionado actualmente y, si es posible, información sobre el sistema de ficheros actual.

**arBol** La vista Árbol muestra “casi” el árbol de directorios.

<sup>6</sup>Modo por defecto cuando entramos en el programa.

**Ordenar ...** Los ocho modos de ordenación son: sin ordenar, por nombre, por extensión, por fecha de modificación, por fecha de acceso, por la fecha de modificación de la información del i-nodo, por tamaño y por i-nodo. En el cuadro de diálogo del modo de ordenación podemos elegir el modo de ordenación así como especificar si deseamos que éste se realice en orden inverso chequeando la casilla Invertir.



Por defecto los directorios son ordenados antes que los ficheros pero esto puede ser cambiado desde el Menú de Opciones (opción Mezcla todos los ficheros).

**Filtro...** El comando de filtro nos permite seleccionar un patrón (por ejemplo \*.tar.gz ) con el cual los ficheros deben coincidir para ser mostrados. Indiferente al patrón de filtro, los directorios y enlaces a directorios se muestran siempre en el panel de directorios.

**Conexión por red...** Véase 6.4.3

**conexión por FTP...** Véase 6.4.1

**actualizaR** El comando actualizar recarga la lista de ficheros en el directorio.

## 6.2.2. Menú de Fichero

*Midnight Commander* utiliza las teclas de función F1 - F10 como atajos de teclado para los comandos que aparecen en el menú de Fichero.



El menú de Fichero posee los siguientes comandos (teclas rápidas de teclado entre paréntesis):

**meNú de usuario(F2)** Invoca el Menú de usuario. El menú de usuario otorga una manera fácil de tener usuarios con un menú y añadir asimismo características extras a *Midnight Commander*. Contiene acciones realizables con el fichero actualmente elegido y que pueden ser definidas de una forma sencilla.<sup>7</sup>

<sup>7</sup>Para profundizar sobre este tema se puede acudir a la ayuda del programa, donde se explica la función de cada una de las opciones posibles.

**Ver (F3)** Visualiza el fichero actualmente seleccionado. Por defecto invoca el Visor de Ficheros Interno pero si la opción "Usar visor interno" está desactivada, invoca un visor de ficheros externo especificado por la variable de entorno.

**Visor de Ficheros Interno** El visor de ficheros interno posee dos modos de pantalla: ASCII y hexadecimal. Para intercambiar entre modos, usaremos la tecla **F4**. Si tenemos el programa `gzip` instalado, se usará automáticamente para descomprimir los ficheros según se necesite.

El visor intentará usar el mejor método posible en nuestro sistema para mostrar la información. El visor interno de ficheros interpretará algunas secuencias de cadenas para activar los atributos de negrita y subrayado, para conseguir una apariencia mejor de nuestros ficheros.

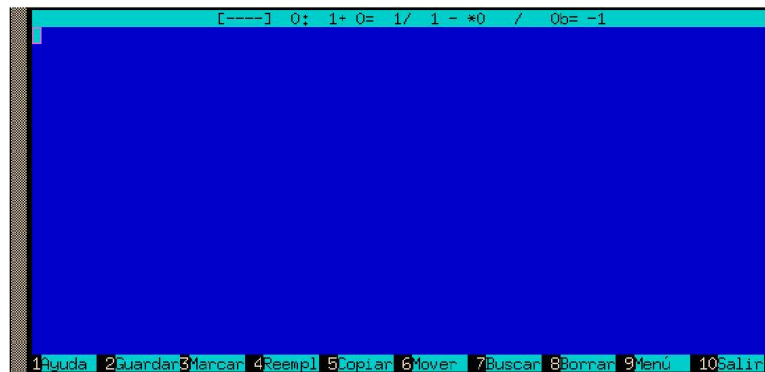
En modo hexadecimal, la función de búsqueda acepta texto entre comillas así como constantes hexadecimales.

**ver fichero ...** Permite visualizar un fichero que le pasemos como argumento. Por defecto toma el fichero seleccionado.

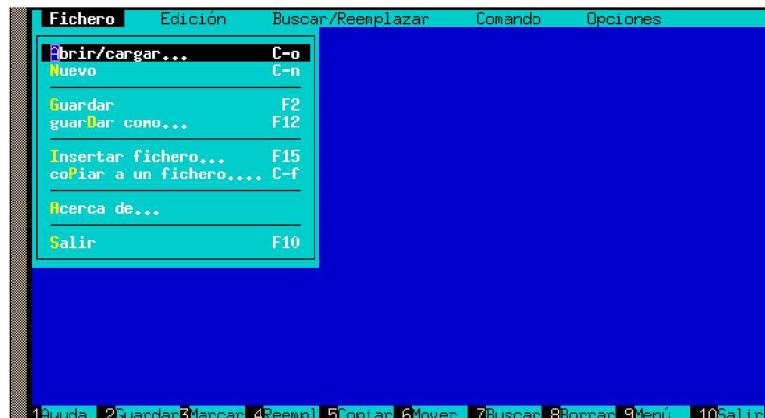
**ver fichero (con Filtro) ... (M-!)** Este comando pregunta por un comando y sus argumentos (el argumento por defecto es el nombre del fichero actualmente seleccionado), la salida de ese comando se muestra en el visor de ficheros interno.

**Editar (F4)** Invoca el editor `vi`, u otro especificado en la variable de entorno `EDITOR`, o en el Editor de Ficheros Interno si la opción `use_internal_edit` está activada.

### Editor de Ficheros Interno



El editor interno proporciona la mayoría de funcionalidades de los editores comunes de pantalla completa. Es invocado pulsando **F4** indicado por la variable `use_internal_edit` en el fichero de inicialización. Tiene un tamaño límite de fichero extensible de dieciséis megabytes y edita los ficheros binarios de manera impecable.



El editor es muy fácil de utilizar y no requiere de aprendizaje alguno. Para activar el menú superior basta con pulsar **F9** o “picar” con el ratón sobre la barra superior.

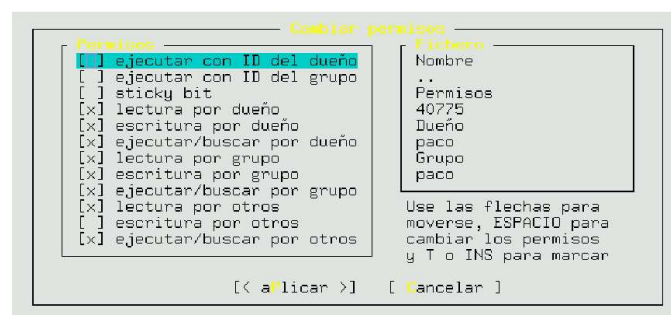
Para conocer la función de las teclas, basta consultar el menú emergente apropiado. Otras teclas son:

- Mayúsculas+cursores producen el resaltado de texto.
- Ctrl-Insert copia al fichero cooledit.clip.
- Mayúsculas-Insert pega desde cooledit.clip.
- Mayúsculas-Supr corta a cooledit.clip, y
- Ctrl-Supr elimina el texto resaltado.
- La tecla de terminación también realiza un Return con un sangrado automático

La selección con ratón también funciona.

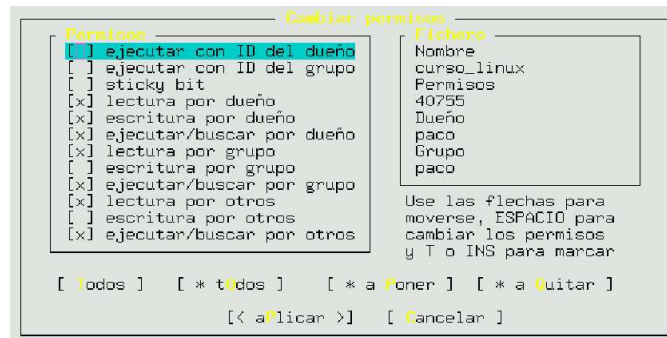
**Copiar (F5)** Sobreimpresiona una ventana de entrada con destino por defecto al directorio del panel no seleccionado y copia el fichero actualmente seleccionado (o los ficheros marcados, si hay al menos uno marcado) al directorio especificado por el usuario en la ventana.

**cambiar Permisos ... (C-x c)** Permite cambiar los permisos de los ficheros. La ventana de Chmod se usa para cambiar los bits de atributo en un grupo de ficheros y directorios. Puede ser invocada con la combinación de teclas C-x c. Si sólo tenemos seleccionado un archivo, la ventana que aparece es:



Si realizamos una selección múltiple de archivos la ventana de Chmod tiene dos partes - Permisos y Archivo





En la sección Archivo se muestran el nombre del fichero o directorio y sus permisos en formato numérico octal, así como su propietario y grupo.

En la sección de permisos hay un grupo de campos que corresponden a los bits de atributos del fichero. Conforme cambiamos los bits de atributo, podemos ver el valor octal cambiando en la sección Archivo.

Para aceptar los atributos, usaremos la tecla **Enter**. Cuando trabajamos con un grupo de ficheros o directorios, basta pulsar con el ratón sobre los bits que queremos activar o desactivar. Una vez seleccionados los bits que queremos cambiar, seleccionamos uno de los botones (Marca activa o marca desactiva).

Finalmente podemos usar:

**[Todos]**, que actuará sobre todos los ficheros marcados.

**[\* Todos]** actúa sólo sobre los atributos marcados de los ficheros seleccionados

**[\* a Poner]** Activa los bits marcados de los atributos de los ficheros seleccionados

**[\* a Quitar]** Borra los bits marcados de los atributos de los ficheros seleccionados

**[<aPlicar>]** Activa los atributos de un fichero

**[Cancelar]** cancela el comando Chmod.

**crear enLace (C-x l)** Crea un enlace al fichero actual.

**crear enlace Simbólico .... (C-x s)** Para aquellos que no conozcan qué son los enlaces: crear un enlace físico o duro (hard) a un fichero es algo parecido a copiar el fichero, salvo que el fichero original y el destino representan el mismo fichero físico, los mismos datos reales. Por ejemplo, si editamos uno de esos ficheros, todos los cambios que realicemos aparecerán en ambos ficheros. Hay quien llama a los enlaces alias o accesos directos.

Un enlace físico aparece como un fichero real. Después de crearlo, no hay modo de decir cuál es el original y cuál el enlace. Si borramos uno de ellos el otro aún seguirá intacto. Es muy difícil advertir que los ficheros representan la misma imagen. Usaremos estos enlaces cuando no necesitemos saberlo.

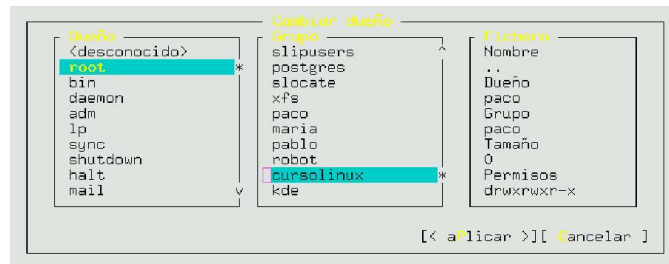
Un enlace simbólico es una referencia al nombre del fichero original. Si el fichero original se borra, el enlace simbólico queda sin utilidad. Es bastante fácil advertir que los ficheros representan la misma imagen.

*Midnight Commander* muestra un símbolo "@" delante del nombre del fichero si es un enlace simbólico a alguna parte (excepto a un directorio, caso en que muestra una tilde (~)). El fichero original al cual el enlace apunta es mostrado en la línea de estado si la opción Mini status está habilitada. Usaremos enlaces simbólicos cuando queramos evitar la confusión que pueden causar los enlaces físicos.

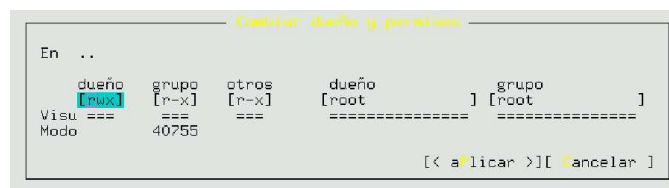
**ediTar enlace simbólico (C-x C-s)** Permite modificar un enlace simbólico.

**cambiar dueñO ... (C-s o)** Permite cambiar los permisos de los ficheros. El comando Chown permite cambiar el propietario/grupo de un fichero. La tecla rápida para este comando es C-x o.





**cCambiar dueño y permisos ...** Permite cambiar al fichero seleccionado el dueño y los permisos



**Renombrar/mover... (F6)** Sobreimpresiona una ventana de entrada que por defecto apunta al directorio en el panel no seleccionado y mueve el fichero actualmente seleccionado (o los ficheros marcados si hay al menos uno) al directorio especificado por el usuario en la ventana. Durante el proceso, podemos pulsar C-c ó ESC para anular la operación.

**crear Directorio (F7)** Sobreimpresiona una ventana de entrada y crea el directorio especificado.

**Borrar (F8)** Borra el fichero actualmente seleccionado o los ficheros marcados en el panel activo.

**caMbiar directorio... (M-c)** Usaremos el comando Cambiar de directorio si tenemos llena la línea de comandos y queremos hacer un cd a algún lugar.

**seleccionar Grupo ... (+)** Se usa para seleccionar (marcar) un grupo de ficheros. Para marcar directorios en vez de ficheros, la expresión debe empezar o terminar con '/'. .

**de-seleccionar grUpo (\)** Utilizado para deseleccionar un grupo de ficheros. Es la operación antagonista al comando Selecciona grupo.

**Invertir la selección (\*)** Si queremos invertir los ficheros seleccionados. Deselecciona los marcados y marca los no marcados.

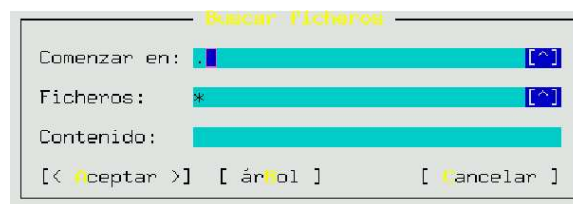
**saliR (F10)** Finaliza *Midnight Commander*.

### 6.2.3. Menú de Comando



**Árbol de Directorios** Muestra una ventana con estructura de árbol con los directorios.

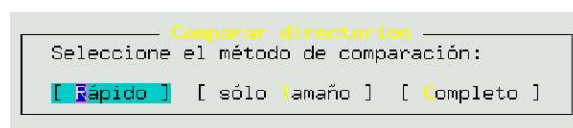
**Buscar ficheros... (M-?)** Permite buscar un fichero específico o ficheros con un contenido determinado.



**Intercambiar paneles (C-u)** Intercambia los contenidos de los dos paneles de directorios.

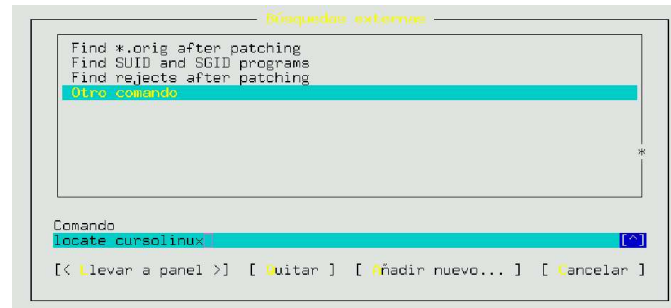
**Activar/desactivar paneles (C-o)** Muestra la salida del último comando del shell.

**Comparar directorios... (C-x d)** Compara los paneles de directorio uno con el otro. Hay tres métodos de comparación:



- El método rápido compara sólo el tamaño de fichero y la fecha.
- El método completo realiza una comparación completa octeto a octeto.
- El método de comparación de sólo tamaño sólo compara los tamaños de fichero y no chequea los contenidos o las fechas.

**búsquedas eXternas ... (C-x !)** Con este comando podemos ejecutar un programa “externo” y la salida de ese programa se visualiza en el panel actual.

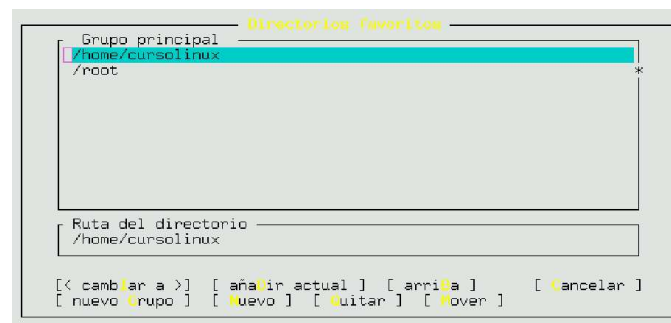


Por ejemplo, si utilizamos el comando  
`locate cursolinux`  
la salida del comando aparecerá en el panel marcado en ese momento. En este caso, el comando lo que hace es buscar ficheros/directorios cuyo nombre sea cursolinux.

#### mostrar Tamaños de los directorios

**Historia de comandos** Muestra una lista de los últimos comandos utilizados. El comando seleccionado se copia a la línea de comandos.

**directorios Favoritos (C-l)** Realiza el cambio desde el directorio actual a los directorios seleccionados por nosotros que usemos más a menudo de forma más rápida.



**directorios virtuales (VFS) ... (C-x a)** Nos permite acceder más rápido a nuestro directorio de usuario.

**Procesos en 2º plano... (C-x j)** Permite controlar el estado de cualquier proceso de *Midnight Commander* en segundo plano. Podemos parar, reiniciar y eliminar procesos en segundo plano desde aquí.

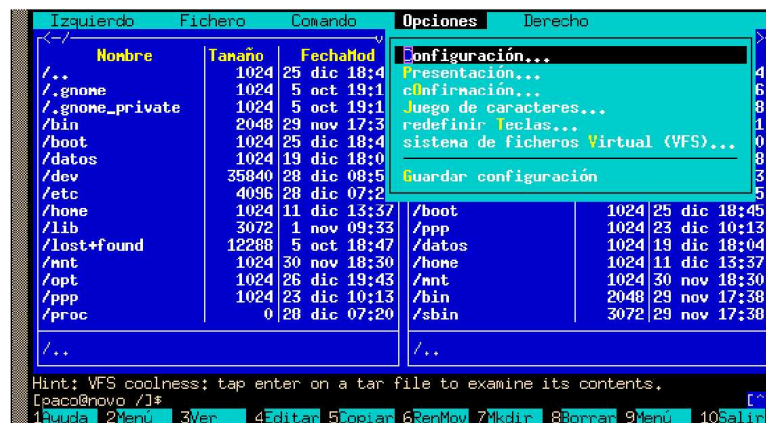
**Recuperar ficheros (ext2fs) ...** Se explica en 6.4.4

**Editar el fichero de extensiones ...** Nos permite especificar los programas a ejecutar para intentar ejecutar, ver, editar y realizar un montón de cosas sobre ficheros con ciertas extensiones (terminaciones de fichero). Por ejemplo, asociar la extensión de los ficheros de audio de SUN (.au) con el programa reproductor adecuado. Las extensiones de ficheros permiten que al pulsar la tecla **Intro** o hacer un doble clic sobre un fichero, intentar verlo, editarlo o arrastrar con el ratón otro fichero encima, se ejecute la aplicación seleccionada en el fichero de extensiones para la extensión en cuestión. En el fichero de extensiones debe existir una línea de entrada para cada extensión que queramos que el *Midnight Commander* interprete, esa entrada comienza en la columna primera; las líneas que siguen y que empiezan con un espacio en blanco o una tabulación, son las diferentes acciones que queremos definir para esa extensión.

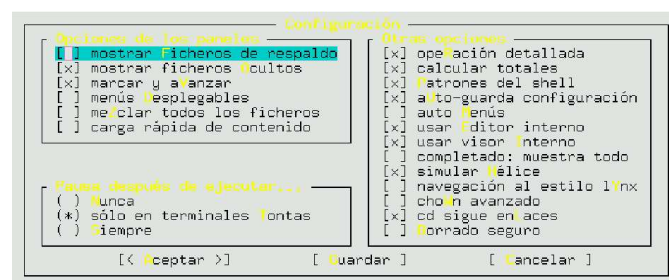
**editar fichero de Menú ...** El menú de usuario es un menú de acciones útiles que puede ser personalizado por el usuario. Cuando accedemos al menú de usuario, se utiliza si existe el fichero .mc.menu del directorio actual, pero sólo si es propiedad del usuario o del root y no es modificable por todos. Si no se encuentra ese fichero, se intenta con ~/.mc/menu, y si no, mc utiliza el menú por defecto para todo el sistema /usr/lib/mc/mc.menu.

## 6.2.4. Menú de Opciones

El comando Opciones muestra un diálogo desde el cual podemos cambiar la configuración de *Midnight Commander*.



**Configuración ...** Permite acceder a una ventana con la que podemos activar o desactivar algunas opciones para configurar el programa. Las opciones están activas si tienen un asterisco o "x" delante. Esas opciones están divididas en tres grupos: Opciones de los Paneles, Pausa después de ejecutar y Otras Opciones.



**Opciones del panel:**

- mostrar Ficheros de respaldo: por defecto, *Midnight Commander* no muestra ficheros terminados en '~' (copias de seguridad).
- mostrar ficheros Ocultos: *Midnight Commander* mostrará todos los ficheros que comienzan con un punto (como ls -a).
- marcar y avanzar: Por defecto, cuando marcamos un fichero (con C-t o la tecla Insert) la barra de selección se desplaza hacia abajo.
- Mostrar Mini-Estado. Si está activa, muestra una línea de información de estado en la parte inferior de los paneles sobre el ítem actualmente seleccionado.
- mezclar todos los ficheros: cuando esta opción está habilitada, los ficheros y directorios se muestran mezclados. Si la opción está desactivada, los directorios (y enlaces a los mismos) se muestran al principio de la lista, y el resto de ficheros a continuación.
- carga rápida de contenido: esta opción está desactivada por defecto. Si la activamos, *Midnight Commander* usará un truco para determinar si los contenidos del directorio han cambiado. El truco consiste en recargar el directorio sólo si el nodo-i del directorio ha cambiado; esto significa que las recargas suceden sólo cuando los ficheros son creados o borrados. Si lo que cambia es el nodo-i de un fichero en el directorio (cambia el tamaño del fichero, cambia el modo o propietario, etc) la pantalla no se actualiza. En esos casos, si tenemos la opción activada, deberemos releer el directorio manualmente (con C-r).

**Pausa después de ejecutar:** Tras ejecutar nuestros comandos, *Midnight Commander* puede quedarse en 2º plano, de tal modo que podamos examinar la salida del comando. Hay tres posibles selecciones para esta variable:

- Nunca: significa que no queremos ver la salida de nuestros comandos. Si estamos utilizando la consola Linux o un xterm, podremos ver la salida del comando pulsando C-o.
- sólo en terminales Tontas: obtendremos un mensaje de pausa y que no son capaces de mostrar la salida del último comando ejecutado (en realidad, cualquier terminal que no sea un xterm o una consola de Linux).
- Siempre: el programa realizará una pausa después de ejecutar todos nuestros comandos.

**Otras opciones:**

- opeRación detallada: hace que las operaciones de copia, renombrado y eliminación de ficheros sean detalladas (p.e., muestra un cuadro de diálogo para cada operación). Si tenemos un terminal lento, podríamos querer desactivar la operación detallada. Ésta es automáticamente desactivada si la velocidad de nuestro terminal es menor de 9600 bps.
- calcular totales: si tenemos activa esta opción el *Midnight Commander* suma el total del tamaño de los ficheros así como el número de ficheros antes de cualquier operación de copia, renombrado o borrado. Se obtendrá una barra de progreso más exacta en estos procesos a cambio de pérdida de velocidad. Esta opción no tiene efecto si no está activa opeRación detallada.
- Patrones del shell: por defecto los comandos de Selección, Deselección y Filtro usarán expresiones regulares del estilo del shell. Para realizar esto se hacen las siguientes conversiones: el '\*' es remplazado por cero o más caracteres; la '?' por exactamente un carácter y '.' por el punto literal.
- aUto-guarda configuración: si esta opción está activada, cuando salimos de *Midnight Commander* las opciones configurables de *Midnight Commander* se guardan en el fichero ~/.mc/ini.

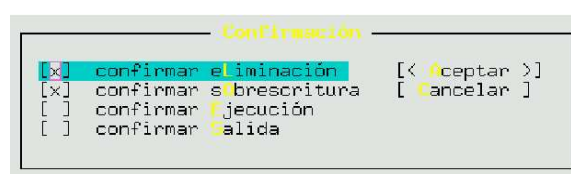
- auto Menús: Si está activada, el menú de usuario es invocado al arrancar. Útil para menús contruidos por personas ajenas a Unix.
- usar Editor interno: si está activada, el editor de ficheros incorporado es utilizado para editar ficheros. Si está desactivada, se usará el editor especificado por la variable de entorno EDITOR. Si no se especifica ninguno, se usará vi como editor de ficheros interno.
- usar visor Interno: si está activada, el visor de ficheros incorporados es utilizado para ver ficheros. Si la opción está desactivada, el visor especificado en la variable de entorno PAGER será el utilizado. Si no se especifica ninguno, se usará el comando view visor de ficheros interno.
- completado: muestra todo: por defecto el *Midnight Commander* muestra todas las posibles maneras de completarse si hay ambigüedad al darle a M-Tab dos veces, la primera vez completa todo lo posible y pita en caso de ambigüedad. Para poder ver todas las posibles maneras de completarse después de presionar M-Tab la primera vez hay que activar esta opción.
- simular Hélice: cuando está activa muestra un guión rotando en la esquina superior derecha para indicar que hay un trabajo en curso.
- navegación al estilo lYnx: si está activa podemos usar las flechas del teclado para para hacer chdir automáticamente siempre que la selección actual sea un subdirectorio y la línea de comandos esté vacía.
- Borrado seguro: opción activa por defecto, provocará que *Midnight Commander* pida confirmación cuando borremos ficheros.
- cd sigue los enLaces: si está seleccionada, hace que *Midnight Commander* siga la “cadena lógica” de directorios. Cuando está deseleccionada, *Midnight Commander* sigue la estructura real de directorios, así si hemos entrado en un directorio a través de un enlace y ejecutamos el comando cd .. , este comando nos trasladará al padre real del directorio actual y no al directorio donde se encontraba el enlace.

**Presentación ...** La ventana de presentación nos da la posibilidad de cambiar la presentación general de la pantalla.

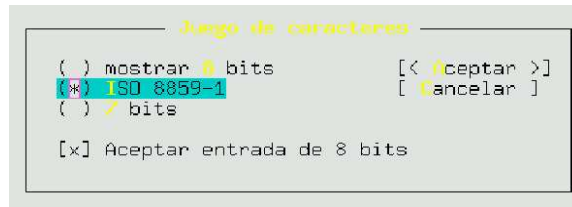


Por ejemplo, podemos especificar cuándo son visibles la Barra de Menú, la línea de comandos, la línea de sugerencias de xterm y la Barra de teclas de Función. Además, podemos dividir la pantalla en dos paneles verticales u horizontales, la división puede ser simétrica o bien podemos indicar una división asimétrica.

**cOnfirmación ...** Accedemos a un diálogo desde el cual podemos especificar qué acciones queremos que sean confirmadas antes de ser realizadas.



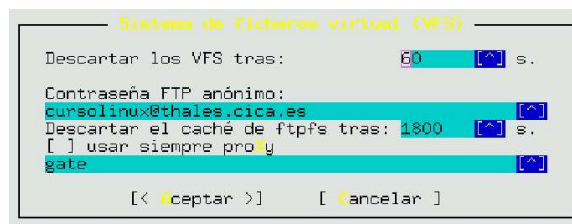
**Juego de caracteres ...** Desde él podemos seleccionar qué caracteres es capaz de visualizar nuestro terminal.



Si queremos escribir y visualizar correctamente en castellano (acentos y ñ) tendremos que tener activas las opciones del gráfico.

**redefinir Teclas ...** Este comando nos permite comprobar si nuestras teclas F1-F20, Inicio, Fin, etc. funcionan adecuadamente en nuestro terminal.

**sistema de ficheros Virtual (VFS)...** Muestra un diálogo desde el cual podemos especificar algunas opciones relacionadas con VFS<sup>8</sup> (Sistema de Archivos Virtual).



Esta opción nos proporciona el control sobre la caché de información del Sistema de Ficheros Virtual (VFS).

*Midnight Commander* guarda en memoria la información relacionada con alguno de los sistemas de ficheros virtuales para acelerar el acceso a los ficheros en el sistema de ficheros. Para acceder a los contenidos de ficheros comprimidos (por ejemplo, los ficheros tar comprimidos) este programa debe crear un archivo temporal descomprimido en el disco. Como la información en memoria y los archivos temporales ocupan recursos podríamos querer ajustar los parámetros de la información con caché para disminuir la utilización de memoria o aumentar la velocidad de acceso a los sistemas de ficheros más utilizados.

El sistema de ficheros Tar es bastante inteligente a la hora de manejar sus ficheros: sólo carga las entradas de los directorios y cuando necesita usar la información contenida en el fichero tar, va y la toma.

Los ficheros tar normalmente se guardan comprimidos, y debido a la naturaleza de esos ficheros (las entradas de directorio de los ficheros tar no están allí esperando a que las carguemos nosotros), el sistema de ficheros tar tiene dos posibilidades: cargar todo el fichero tar descomprimido en memoria o descomprimir el fichero en el disco en una localización temporal y acceder entonces al fichero descomprimido como a un fichero tar normal.

Ahora, dado que a todos nos encanta navegar por los ficheros, incluidos los tar, sobre el disco, es común que salgamos de un fichero tar y volvamos a entrar en él después. Puesto que la descompresión es lenta, *Midnight Commander* mantendrá en memoria la información durante una cantidad de tiempo limitado, después de alcanzado el momento, todos los recursos de memoria asociados con el sistema de ficheros serán liberados. El período por defecto es de un minuto.

El FTP File System mantiene el listado del directorio en la caché. El tiempo de finalización de la caché lo podemos configurar, un valor bajo para esta opción puede hacer más lenta cualquier operación en el FTP File System, porque cualquier operación va acompañada por una pregunta del servidor ftp.

<sup>8</sup>Véase 6.4



Además, podemos definir un proxy para hacer transferencias ftp y configurar *Midnight Commander* para que siempre lo use.

**Guardar configuración** Guarda los valores actuales de los menús Izquierdo, Derecho y Opciones. También se guarda un pequeño grupo de otros valores.



Si activamos la opción Auto-guarda configuración, MC guardará siempre la configuración actual al salir.

Existen también configuraciones que no pueden ser cambiadas desde los menús. Para cambiarlas tendremos que editar el fichero de configuración.

### 6.3. Ejecutando Comandos del Sistema Operativo

Podemos ejecutar comandos tecleándolos directamente en la línea de entrada de *Midnight Commander*, o seleccionando el programa que queremos ejecutar con la barra de selección en uno de los paneles y pulsando Enter.

Si pulsamos Enter sobre un fichero que no es ejecutable, *Midnight Commander* chequea la extensión del fichero seleccionado con las extensiones en el Fichero de Extensiones. Si se produce una coincidencia se ejecutará el código asociado con esa extensión.

#### 6.3.1. El Comando cd Interno

El comando cd es interpretado por *Midnight Commander*, no es pasado al shell de comandos para ser ejecutado.

**Substitución de Tilde** La tilde (~) será substituida por nuestro directorio de inicio, si añadimos un nombre de usuario tras la tilde, entonces será substituido por el directorio de entrada al sistema del usuario especificado.

**Directorio Anterior** Podemos saltar al directorio donde estábamos anteriormente de la mano del nombre de directorio especial '-' del siguiente modo:

```
cd -
```

### 6.4. Sistema de Ficheros Virtual

El selector del Sistema de Ficheros Virtual (SFV) permite a *Midnight Commander* manipular ficheros no localizados en el sistema de ficheros Unix.

Actualmente *Midnight Commander* está preparado para cinco SFV:

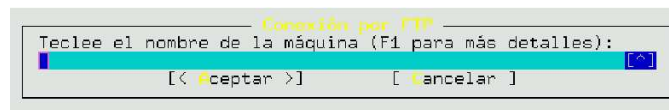
1. el sistema de ficheros local, utilizado para acceder al sistema Unix común;
2. el ftpfs, que manipula ficheros en sistemas remotos a través del protocolo FTP (File Transfer Protocol);
3. el tarfs, usado para manipular ficheros tar o tar comprimidos;
4. el undelfs, utilizado para recuperar ficheros borrados en sistemas de ficheros ext2 (el sistema de ficheros por defecto para sistemas Linux) y



5. finalmente el mcfs (*Midnight Commander* file system), un sistema de ficheros de red.

#### 6.4.1. Sistema de Ficheros FTP

El ftpfs permite manipular ficheros en máquinas remotas, para usarlo ahora, deberíamos intentar usar el comando del panel "Conexión FTP" (accesible desde los menús laterales iz. y der.)



en el campo que aparece escribiremos:

usuario:contraseña@máquina

cuando conectemos en el panel que teníamos seleccionado nos aparecerá la lista de ficheros de la máquina remota. Luego podremos navegar y copiar ficheros como si todo estuviera pasando en nuestro propio ordenador.

También podemos realizar directamente la conexión utilizando el comando `cd` con la ruta:

```
ftp://[!][usuario[:contraseña]@]maquina[:puerto][directorio- remoto]
```

los elementos, usuario, puerto y directorio-remoto son opcionales (Entre corchetes, []). Si especificamos el elemento usuario, entonces *Midnight Commander* intentará entrar en la máquina remota como ese usuario, en otro caso usará nuestro login. El elemento opcional contraseña, si está presente es la contraseña de acceso usada para autentificar la conexión.

##### Ejemplos:

```
cd ftp://ftp.nuclecu.unam.mx/linux/local
cd ftp://tsx-11.mit.edu/pub/linux/packages
cd ftp://!behind.firewall.edu/pub
cd ftp://guest@remote-host.com:40/pub
cd ftp://miguel:xxx@server/pub
cd ftp://ftp.um.es/pub
```

Para acceder a lugares tras un cortafuegos usaremos el prefijo `ftp://!` (es decir, con una exclamación tras la doble barra) para hacer que *Midnight Commander* utilice un proxy para realizar la transferencia ftp. Pondremos el proxy en el cuadro de diálogo Sistema de Ficheros Virtual.

#### 6.4.2. Sistema de Archivos Tar

El sistema de ficheros tar y los ficheros tar comprimidos pueden consultarse usando el visor interno, es decir, seleccionando el fichero y pulsando F3. *Midnight Commander* permite navegar por ficheros del tipo `.tgz`, `tar.gz`, `.Z`, `.rpm`, `.deb`, etc. Si accedemos a ficheros empaquetados, podemos navegar por los subdirectorios que pudieran contener, además podemos realizar algunas operaciones sencillas como copiar ficheros a y desde un directorio normal, o leer los contenidos de ficheros de texto.

Para ficheros tar también es posible usando el comando `chdir`. Para cambiar nuestro directorio a un fichero tar, cambiaremos nuestro directorio actual al fichero tar utilizando la sintaxis:

```
tar:NombreDeFichero.tar[Directorio-dentro_de-tar]
```

Normalmente basta con apuntar a un fichero tar y pulsar Return para entrar en el fichero tar.

#### 6.4.3. Sistema de Ficheros de Red

El sistema de ficheros de *Midnight Commander* es un sistema de ficheros de red básico que nos permite manipular ficheros en una máquina remota como si estuviesen accesibles localmente. Para utilizar esto, la máquina remota debe estar ejecutando el programa servidor `mcfserv`.

Para conectar a una máquina remota, sólo necesitamos hacer un `chdir` a un directorio especial cuyo nombre sigue el siguiente formato:

```
mc:[usuario@]máquina[:puerto][directorio-remoto]
```

Los elementos usuario, puerto y directorio-remoto son opcionales. Si especificamos el elemento usuario entonces *Midnight Commander* intentará acceder a la máquina como ese usuario, si no usará nuestro login.

El elemento puerto es utilizado cuando la máquina remota se ejecuta en un puerto especial (véase la página del manual de mcserv para mayor información sobre puertos); finalmente, si el elemento directorio remoto está presente, nuestro directorio actual en la máquina remota será éste.

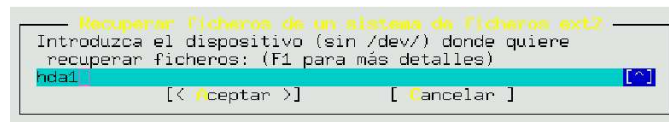
#### Ejemplos:

```
mc:ftp.nuclecu.unam.mx/linux/local
mc:Pepe@foo.edu:11321/privado
mc:Gabriel@Dersu.ArocaSoft.es:1235/SolidarityWare
```

### 6.4.4. Sistema de Archivos de Recuperación

La recuperación de ficheros borrados está disponible sólo en los sistemas de ficheros ext2. El sistema de ficheros recuperable es sólo un interfaz de la librería ext2fs con: restaurar todos los ficheros borrados en un ext2fs y proporcionar y extraer los ficheros seleccionados en una partición regular.

Si usamos el menú, al seleccionar esta opción veremos:



Tendremos que introducir el dispositivo donde queremos recuperar los ficheros. Esto le llevaría un tiempo a undelfs para cargar la información antes de empezar a navegar por los ficheros allí contenidos.

Para usar este sistema de ficheros desde la línea de comandos, tendremos que hacer un chdir en el nombre de fichero especial formado por el prefijo "undel:" y el nombre de fichero donde se encuentra el sistema de ficheros actual.

Por ejemplo, para recuperar ficheros borrados en la segunda partición del primer disco scsi en Linux, usaríamos:

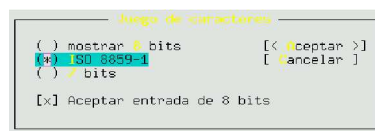
```
cd undel:/dev/sda2
```

### 6.4.5. ➡ Para practicar

No es necesario conocer a fondo todo lo expuesto para utilizar el programa. Si controlamos las cuestiones siguientes será suficiente por ahora:

1. Cambiar las opciones del programa mc para que permita ver y escribir acentos.

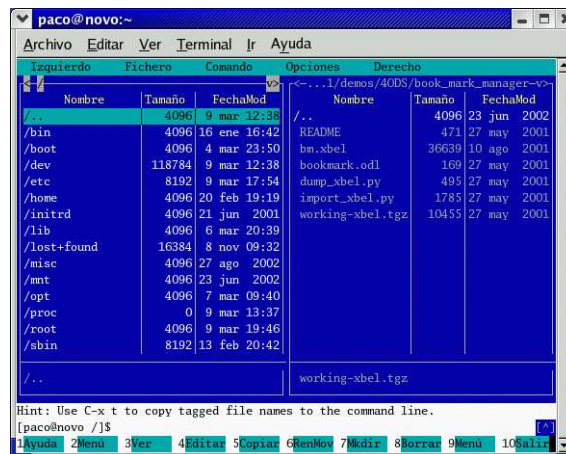
#### Opciones→Juego de caracteres



2. Editar con el programa mc (o con mcedit) el fichero /etc/issue y cambiar el mensaje de inicio en modo texto a:

```
Hola desde un Red Hat Linux Release 8.0 (Psyche)
Kernel \r on an \m
```

- Con el mc
    - #mc
    - Marcar el fichero en vídeo inverso y **Fichero**→**Editar** o **[F4]+**
  - Con el programa mcedit:
    - #mcedit /etc/issue
3. Entrando como root vamos a seleccionar el directorio `/etc/sysconfig` y copiarlo a nuestro directorio de root. Para hacer ésto en el panel izquierdo seleccionaremos el citado directorio mientras que en el derecho nos situaremos en el del root, luego sólo tenemos que pulsar **[F5]** y **Aceptar**
  4. Seleccionar los ficheros del directorio antes copiado (**[+]** o **Fichero**→**seleccionar Grupo**) y borrarlos con **[F8]** (manteniendo el directorio).
  5. Ver el contenido de un fichero tar.gz (parecido a ficheros zip): Marcar con vídeo inverso el fichero `/usr/share/doc/4Suite-0.11.1/demos/40DS/book_mark_manager/working-xbel.tgz`



y después **[Intro]**

## **Parte II**

# **Linux en modo gráfico. Configuración**

## Prefacio

Comienza aquí la segunda entrega del curso, en esta entrega pretendemos que se consiga configurar correctamente nuestro equipo y que seamos nosotros los que tengamos el control sobre nuestra máquina y no el “dichoso ordenador” sobre nosotros.

Algunas de las cuestiones que vamos a ver ahora ya han aparecido de pasada al tener que configurar el entorno gráfico. A lo largo de esta entrega vamos a intentar afianzar el control del entorno gráfico para, en capítulos posteriores, justificar y desarrollar el trabajo en modo consola.

La configuración es un tema delicado. Por esto, sería conveniente una primera lectura “rápida” de todos los capítulos de la entrega, ya que están interrelacionados y no siempre siguen un esquema lineal.

Además, algunos puntos del primer capítulo analizan casos particulares que puede que no todos necesitéis o incluso que ya hayáis configurado, pero es obligado tener una referencia de esos casos, que si bien son particularizaciones surgen con bastante frecuencia.

En un curso de estas características, la duración del mismo y la metodología de trabajo obligan a ir “adelantando contenidos” para conseguir que el trabajo sea lo más fructífero posible. Después, en entregas posteriores tendremos que retomar algunas de estas cuestiones para profundizar en ellas.



En toda esta entrega y sucesivas, tendremos que ir instalando paquetes de los CDs de la distribución. Siempre guiaremos su instalación de la forma más general posible: en el supuesto de que se tiene que montar el CD y que se instala el paquete en modo comando.

Si se trabaja en modo gráfico (será lo más usual), cuando se introduzca un CD se inicia un proceso de automontado, y, por tanto, no es necesario el comando `mount` (de hecho si se ejecuta dará error). Además, si se trabaja con Nautilus<sup>9</sup>, para instalar un paquete sólo hay que pulsar sobre él (dos veces) con el ratón, lo que implica que tampoco es necesario ejecutar el comando `rpm -ivh nombre_paquete`.

---

<sup>9</sup>Gestor de ficheros de GNOME

## Capítulo 7

# Configuración de las X window. Gestores de Ventanas.

Mientras que el corazón de Red Hat Linux es el kernel, para muchos usuarios, la cara del sistema operativo es el entorno gráfico proporcionado por el Sistema X Window, también llamado simplemente X. (*The Official Red Hat Linux Reference Guide*)

### 7.1. ¿Qué es un gestor de ventanas?

Comenzaremos este apartado con un poco de teoría para intentar resumir en unas cuantas líneas qué es el entorno gráfico X Window.

#### 7.1.1. Introducción:

El acercamiento al usuario final de la informática no habría sido posible sin interfaces gráficas de usuario (GUI)<sup>1</sup>. Aunque todos los aspectos de configuración de una máquina GNU/Linux pueden realizarse en modo texto es evidente que usando programas gráficos hay ocasiones en que las labores más complicadas de configuración se reducen a pulsar varias veces sobre el ratón. Además, programas de retoque fotográfico, tratamiento de textos u hojas de cálculo se convierten en un juego de niños usando este tipo de entornos.

Hoy en día es imposible pensar que un sistema operativo pueda “triunfar” sin GUIs que faciliten las labores del día a día y que permitan trabajar con todo tipo de programas.

En sistemas GNU/Linux es posible disponer de una GUI gracias a la labor iniciada a principios de los años 80 por el equipo responsable del proyecto Athena en el Instituto Tecnológico de Massachusetts (MIT). La idea era poder trabajar en modo gráfico con distintos equipos conectados en una red heterogénea.

En Enero de 1988 se creó por parte de las grandes empresas del sector informático el MIT X Consortium, una organización sin ánimo de lucro con el objetivo de asegurar la evolución del Sistema X Window como un producto abierto<sup>2</sup>. Mediante este sistema el MIT define el mecanismo por el que se particiona la pantalla en ventanas.

En Linux, podemos disponer de un sistema X gracias al proyecto XFree86<sup>3</sup>, un servidor X para PCs basado en el estándar X11 de X Window. X Window se basa en una gran cantidad de herramientas o librerías.

La RedHat 8.0 viene acompañada de la versión 4.2.0 de las XFree86. La versión 4 ha supuesto una “ruptura” con respecto a las versiones anteriores (la 3). Mientras que en la versión 3 se tenía que cargar

---

<sup>1</sup>Del inglés Graphical User Interface

<sup>2</sup>En la actualidad se trabaja en la versión 11 de las X, revisión número 6, es decir, X11R6

<sup>3</sup>Organización sin ánimo de lucro cuyo objetivo es proporcionar un servidor para distintos sistemas operativos compatible con la implementación oficial del X Window.

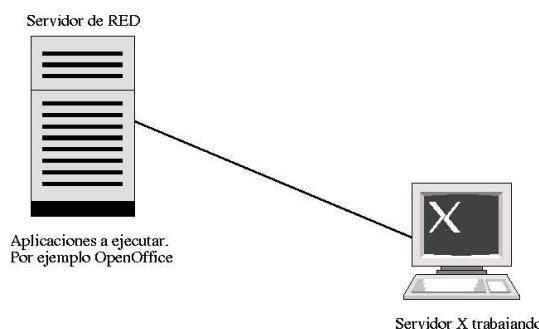
un servidor en función de la tarjeta gráfica de que disponíamos<sup>4</sup> en la versión 4 o posteriores hay un solo servidor para todas las tarjetas gráficas soportadas<sup>5</sup>. Así, en función de la tarjeta que tengamos se cargarán dinámicamente los módulos necesarios.

Como ya se ha dicho, la versión 4 se basa en un nuevo diseño modular que permite tener un solo fichero para el programa servidor X sea cual sea nuestra tarjeta (siempre que esté soportada). Esto supone una clara ventaja en cuanto a la rapidez con la que trabaja esta versión con respecto a las anteriores. Además dispondremos de:

- Soporte acelerado 3D por hardware<sup>6</sup>
- Posibilidad de conectar varias tarjetas gráficas y monitores a nuestra máquina.
- Soporte VESA DDC que facilita la configuración de nuestro monitor.
- Servidor de fuentes<sup>7</sup> integrado.

Con el sistema X podemos disponer de una GUI en nuestra máquina local y además nos permite compartir y ejecutar aplicaciones gráficas en red.<sup>8</sup> El modelo en que se basan las X es un modelo cliente/servidor: el servidor X es el encargado de controlar el teclado, ratón y monitor de la máquina en que se ejecuta, y responde a los comandos que de los clientes<sup>9</sup> X son enviados al servidor. El servidor X nos proporciona los recursos necesarios para la visualización. El cliente X realiza las operaciones y le dice al servidor qué tiene que mostrar.

El sistema X permite tener instaladas las aplicaciones en una sola máquina (cliente X) y con terminales dotados de entornos gráficos (servidor) podemos acceder a todas las aplicaciones disponibles en nuestro servidor de red. Es decir, podemos tener una aplicación (OpenOffice, por ejemplo) cargada sólo en el servidor de red (cliente) y ejecutar vía red el programa anterior usando sólo los recursos gráficos (servidor X) de la máquina en que vamos a visualizar la aplicación.



Este modelo de funcionamiento está asentado en tres pilares:

- El programa servidor gráfico para X.

<sup>4</sup>Por ejemplo, para una tarjeta SVGA el servidor era el fichero XFree86\_SVGA que forma parte de algún paquete de nombre XFree86-SVGA-3.x.x.i386.rpm.

<sup>5</sup>El único fichero para todas las tarjetas es el ejecutable XFree86 que forma parte del paquete XFree86-4.2.0-72.i386.rpm.

<sup>6</sup>Está listo sólo para unos pocos modelos de tarjetas, para saber cuáles son:

<http://dri.sourceforge.net/>

<sup>7</sup>Del inglés Fonts (tipos de letras)

<sup>8</sup>En este capítulo nos centraremos en cómo trabajar con él en una máquina local. Para aquellos que no puedan esperar merece la pena comentar:

- Es posible ejecutar programas Linux en una máquina con Windows, una utilidad que permite esto es Xwin32 y está disponible en Internet una demo que permite sólo un tiempo limitado de conexión:

<http://www.starnet.com/es>

- Cómo acceder a otra máquina Linux y ejecutar aplicaciones gráficas se puede ver en la página 109

<sup>9</sup>Clientes que no tienen que estar en la misma máquina y que tampoco tienen por qué correr bajo el mismo sistema operativo



- Un gestor de ventanas.
- Aplicaciones gráficas.

Explicemos un poco qué significado tiene cada uno de los pilares anteriores:

### **El programa servidor gráfico para X.**

Para explicar cómo funciona el servidor gráfico imaginemos una familia que quiere decorar su casa, para esto contrata a un operario X (servidor X) que es el que “sabe” cómo realizar las operaciones gráficas, pero por sí sólo no es capaz de hacer nada. Por otro lado tenemos a nuestra familia (programas clientes) que quieren decorar su casa (tareas gráficas) pero no saben cómo se hacen y le dicen al servidor X que se las haga él. Para poder decirle algo al servidor hay una premisa: tienen que entenderse, es decir, tienen que usar el mismo “lenguaje” (protocolo X).

El problema no termina ahí ya que nuestra familia sólo sabe decir al operario “qué” quiere que haga, pero es incapaz de decirle “cómo” quiere que lo haga. Por ejemplo, quiere que le pinte un rectángulo de otro color al borde de los marcos de las ventanas, pero son incapaces de transmitirle ni el color, ni el grosor del marco, ni nada de nada.<sup>10</sup>

Aparece entonces el tercero en discordia. Ya llegó el “decorador” (gestor de ventanas). Él se llevará el premio al final:

### **Gestores de ventanas.**

Nuestra familia está desesperada, sabe qué quiere hacer, ha encontrado el mejor operario de la región, el que mejor trabaja, sabe hacerlo todo, ellos tienen claro “qué” quieren hacer pero son incapaces de decirle cómo quieren que lo haga. De eso se encargará el “decorador” (gestor de ventanas).

Su función consiste en recibir la orden de la familia (cliente) y decide cómo se va a hacer esa orden, Es el responsable de la estética final con que va a quedar nuestra “casa” (escritorio).

El gestor de ventanas es el encargado de controlar la posición y apariencia de las ventanas en la pantalla. Él se encarga de controlar el borde, barra de desplazamiento, botones, etc, de las ventanas que visualizamos en nuestro monitor. Además, gracias a él podemos mover, cambiar el tamaño o cerrar una ventana que estemos visualizando en nuestra pantalla.

Si continuamos con el símil del marco de las ventanas, es el encargado de decir al operario que los marcos sean de un determinado color, grosor, etc. Ante estas aclaraciones nuestro eficiente operario no tarda en realizar su labor.

Gracias a estos gestores de ventanas tendremos una “casa” (escritorio) agradable para trabajar.

Dependiendo del “decorador” que contratemos tendremos resultados estéticos distintos.

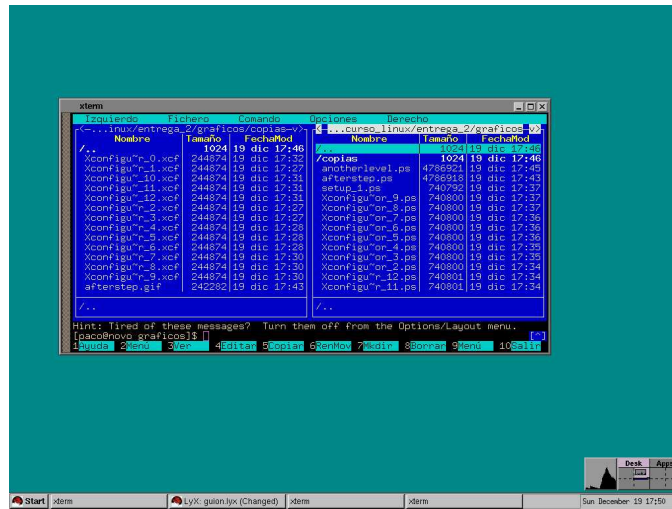
---

<sup>10</sup>Para entender el símil y, si accedemos al sistema en modo texto, sólo tenemos que ejecutar:

§ X

y comprobar la ventana gráfica que aparece. Si queremos salir del Servidor Gráfico X podemos hacerlo con la combinación de teclas [CTRL]+[ALT]+[←]





El gráfico anterior se corresponde con el gestor de ventanas fvwm2 usando la configuración AnotherLevel, mientras que el que sigue es AfterStep.



Afortunadamente, hay a nuestra disposición bastantes gestores de ventanas como para satisfacer a cualquier tipo de “familia”. Algunos gestores de ventanas son<sup>11</sup>:

**twm** *Tab Window Manager*, es un poco “arcaico”.

**wmaker** *WindowMaker*

**sawfish** Gestor de ventanas que usaba GNOME por defecto hasta esta versión de RedHat.

**metacity** *The Metacity Window Manager*. Gestor de ventanas que usa Gnome de forma predeterminada.

**mwm** *The Motif Window Manager*

**Icewm**

**fvwm2**

<sup>11</sup>Una relación de gestores de ventanas se puede consultar en <http://www.plig.org/xwinman>



## enlightenment

### AfterStep

...

Los 5 primeros acompañan a la distribución con la que estamos trabajando. Cada uno tiene sus ventajas e inconvenientes y después indicaremos cómo “probar” algunos de ellos.

Pero... nuestra familia aún no está contenta con el resultado.

### 7.1.2. ¿Es suficiente con los gestores de ventanas?: GNOME y KDE

No contentos con el resultado, además de gestores de ventanas y para simplificar el entorno gráfico surgieron dos entornos de escritorio: GNOME<sup>12</sup> (*GNU Network Object Model Environment*) y KDE (*K Desktop Environment*).

Los entornos de escritorio pretenden hacer más fácil la incorporación de usuarios inexpertos al mundo GNU/Linux. Un escritorio es un entorno de trabajo y un conjunto de aplicaciones de configuración y de programas de utilidad que permiten a los usuarios usar y configurar sus ordenadores de una forma sencilla usando una interfaz gráfica (*GUI*). Al usar un entorno de desarrollo común se permite que las aplicaciones (clientes X) desarrolladas para ellos puedan comunicarse entre sí<sup>13</sup>.

La facilidad de uso que se ha conseguido con los entornos de escritorio ha supuesto un aumento de la necesidad de memoria RAM del sistema: con menos de 64 Megas de RAM y si no se dispone de un microprocesador “aceptable” trabajar con cualquiera de los dos puede ser lento<sup>14</sup>.

## 7.2. Utilidad `redhat-config-xfree86`

Para determinar cuál es el servidor gráfico adecuado para nuestro equipo y si nuestra tarjeta gráfica está soportada deberíamos mirar en:

<http://hardware.redhat.com>

<http://www.xfree86.org/4.2.0/Status.html>

Como se ha comentado en 7.1.1, para trabajar en modo gráfico hay que utilizar tres pilares. En este apartado vamos a analizar las herramientas de que disponemos con esta distribución para configurar el primer pilar: el servidor gráfico X.



La mayoría de ficheros que conforman el servidor gráfico se sitúan en dos directorios dentro de nuestro sistema de ficheros. Se trata de los directorios:

- `/usr/X11R6`  $\mapsto$  binarios, librerías, documentación, ...
- `/etc/X11`  $\mapsto$  ficheros de configuración.

Para realizar la configuración del servidor X, disponemos de varias herramientas en el CD1 de la distribución:

- `xf86config`

Se instala por defecto ya que forma parte del paquete `XFree86-4.2.0-72`. No os lo recomendamos para su uso, ya que es en modo texto y no supone ninguna ventaja sobre el programa que vamos a usar después.

<sup>12</sup>Nos son los únicos, aunque sí los más extendidos.

<sup>13</sup>Por ejemplo, es posible la opción de “arrastrar y soltar” (*drag and drop*) de unas a otras.

<sup>14</sup>Red Hat recomienda 198 MB de RAM para trabajar holgadamente en modo gráfico. Sin embargo, con un Pentium 166 y 64 MB de RAM se puede trabajar aceptablemente bien en modo gráfico con la última versión de un sistema operativo multiusuario de red: ¡Busque, compare y si encuentra algo mejor ...! Avíseme.

#### ■ redhat-config-xfree86


Usaremos el programa de configuración del entorno gráfico (en versiones anteriores de Red Hat: Xconfigurator) que instala RedHat por defecto (se encuentra disponible en el paquete `redhat-config-xfree86`). Permite configurar de forma cómoda el sistema X y su interfaz es sencilla e intuitiva. Hacer notar que antes de utilizarlo deberíamos conocer todas las características de nuestro sistema gráfico.<sup>15</sup>

Para acceder a esta utilidad ejecutaremos:

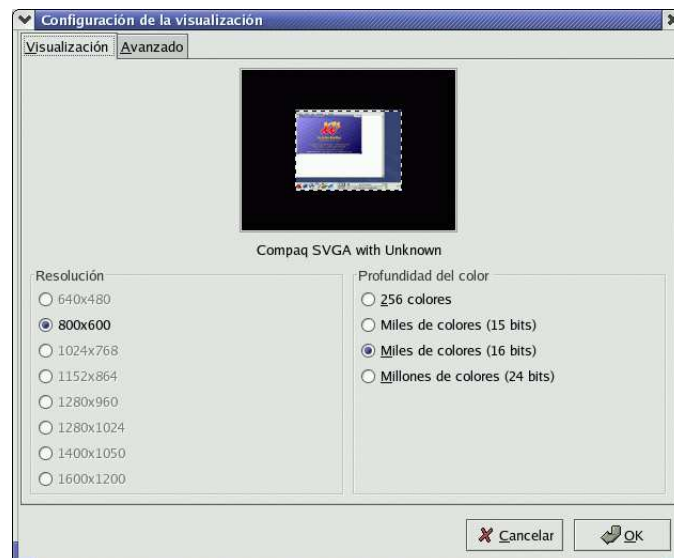
#### ■ En modo texto:

```
# redhat-config-xfree8616
```

#### ■ En modo gráfico podemos usar:

- Menús:  → **Configuración del sistema** → **Visualización**
- Escritorio: **Empezar aquí** → **Configuración del Sistema** → **Visualización**

Disponemos de dos ventanas principales de configuración:



**Visualización** permite seleccionar los modos de vídeo y la profundidad de color.

**Avanzado** para configurar el módulo para nuestra tarjeta gráfica, monitor y tamaño de la zona de visualización.

Inicialmente `redhat-config-xfree86` intenta determinar qué tarjeta de vídeo tenemos instalada (programa `ddcprobe`).

Si no tenemos claro cuál es nuestro modelo de tarjeta de vídeo, podemos intentarlo con: `ddcprobe`. Veamos la salida que se obtendría si utilizamos este programa en uno de los equipos usados para confeccionar estos apuntes:

```
# /usr/sbin/ddcprobe
```

<sup>15</sup>Esta información es la misma que la que tendríamos que tener antes de instalar. Se detallaba en la primera entrega.

<sup>16</sup>Es posible pasar parámetros al ejecutar el programa. Para obtener por pantalla el listado completo usar:

```
$redhat-config-xfree86 --help
```

Uno de los más "útiles" es el de obligar a que la configuración no se base en ficheros existentes en el sistema (si no configuramos bien el entorno gráfico permite comenzar desde cero):

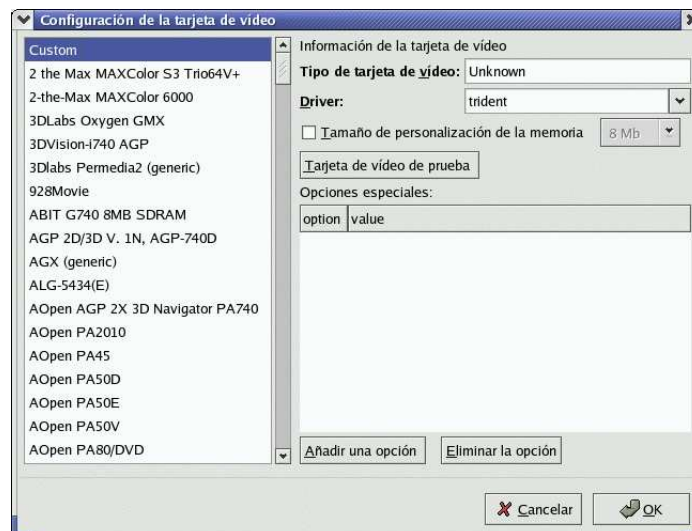
```
#redhat-config-xfree86 --reconfig
```

```
Videocard DDC probe results
Description: NVidia Corporation NV11 (GeForce2)
Board Memory (MB): 32
```

```
Monitor DDC probe results
ID: HTCb001
Name: Hitachi CM643
Horizontal Sync (kHz): 31-95
Vertical Sync (Hz) : 50-130
Width (mm): 330
Height (mm): 250
```

A la mayoría de las tarjetas actuales, las detectará automáticamente, y, por tanto, tendremos que saltarnos los pasos que siguen de selección de tarjeta, ya que el programa `redhat-config-xfree86` le asignará el servidor adecuado.

Si no la detecta, podemos optar por la ventana **Avanzado** y seleccionarla de la lista de tarjetas soportadas por esta versión de las XFree86:



Tendremos que elegir la que se adecúe a nuestro sistema. Si aún así tampoco está nuestra tarjeta, antes de seguir deberíamos comprobar en:

XFree86 <http://www.xfree86.org>

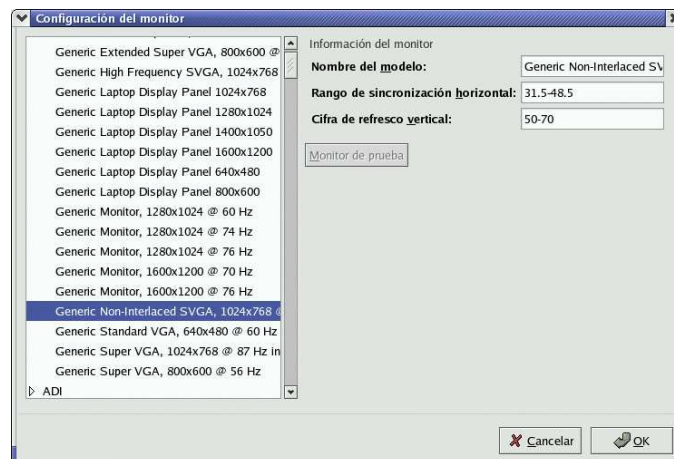
y ver si nuestra tarjeta está soportada por la versión de las XFree86 que estamos utilizando (en este caso 4.2).

Si la nuestra no aparece en la lista, tendremos que optar por seleccionar **Custom** y asignar directamente el módulo cargable del servidor X.



Después, podemos optar por personalizar el tamaño de la memoria de vídeo de la que dispone nuestra tarjeta gráfica mediante una lista desplegable

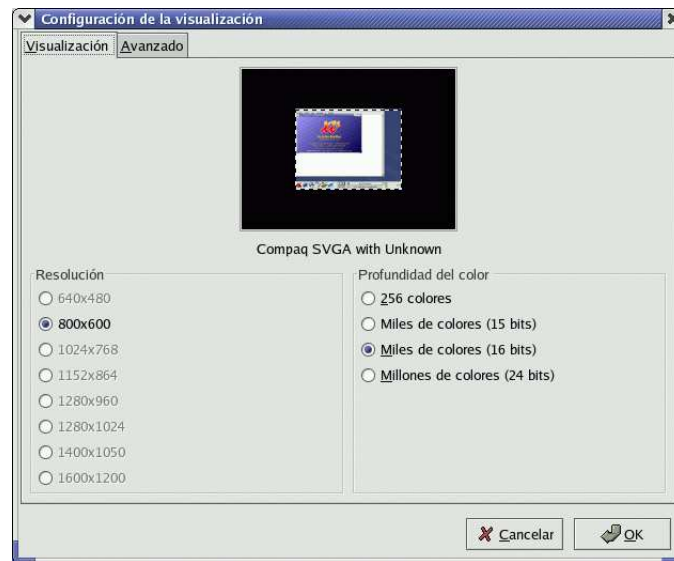
Desde la ventana **Avanzado** podemos configurar nuestro monitor. Si pulsamos sobre **Configurar** accedemos a una pantalla con una base de datos de bastantes modelos de monitores y sus características. Lo usual es que en el proceso de instalación ya se optara por el monitor correcto:



Si esto no es así, tendremos que seleccionar nosotros el monitor de la lista desplegable que nos aparece. En este punto hay que ser muy cuidadoso, no debemos elegir un modelo “parecido” al nuestro, debemos **elegir el nuestro**. Si no está en la lista no nos preocupemos, pulsemos sobre los campos: **Nombre del modelo**, **Rango de sincronismo horizontal** y **Cifra de refresco vertical** e introduciremos los valores adecuados para nuestro modelo.

Si no conocemos exactamente esos valores, podemos arriesgarnos a no elegir nuestro modelo. En este caso hay que ser precavidos e intentar aproximarnos a nuestro modelo al máximo, pero optando por uno de características inferiores. Una vez seleccionado uno (por ejemplo, SVGA 1024x768), se nos permitirá que seleccionemos el intervalo de trabajo de refresco vertical de nuestro monitor. Como antes, ante la duda es mejor ser prudentes y seleccionar valores que estemos seguros que son soportados por nuestro monitor.

Retomemos la ventana inicial del programa. Dependiendo de la capacidad gráfica de nuestro sistema, se nos permitirá que seleccionemos qué modos de vídeo y con qué profundidad de color vamos a trabajar. El gráfico mostrado es el que aparecería con una tarjeta de vídeo con 8 MB de vídeo. Cuanto mayor sea esta cantidad más modos de vídeo y mayor profundidad de color tendremos a nuestro alcance.



Es mejor comenzar con una configuración que no dé problemas, como por ejemplo 640x480 y 8 bits de profundidad de color y, una vez que nos garanticemos que funciona, en una segunda pasada aumentar el número de modos de vídeo y las profundidades de color. Dependiendo del tamaño de nuestro monitor interesa seleccionar más o menos modos de vídeo. Con un 17" el modo idóneo es 1024x768 mientras que con monitores de 14" ó 15" el máximo (depende del gusto de cada uno) debería ser de 800x600.

Por último y tras reiniciar el sistema gráfico podremos usar el servidor X con la nueva configuración almacenada.

Si trabajamos por defecto en modo texto<sup>17</sup>, con ejecutar

```
$ startx
```

tendremos a nuestra disposición toda la potencia (y a veces lentitud) de una GUI. Si algo no está bien configurado, nos avisará y se nos pedirá que reconfiguremos aquello que no funciona.

### 7.2.1. ¿Qué hemos hecho?: fichero /etc/X11/XF86Config.

Comencemos ya a ver algo de qué es Linux y la mejor forma de entender esto se resume en el título de este subapartado. No tendría sentido que estemos en "manos" de un "programita" que hace lo que quiere con nuestro sistema de vídeo y que me esconde lo que está haciendo. `redhat-config-xfree86` no es más que una utilidad muy cómoda para escribir en el fichero que gestiona parte del sistema gráfico. Ese fichero es `/etc/X11/XF86Config`<sup>18</sup>.

<sup>17</sup>Lo usual es que no sea así. Si ejecutamos este comando desde un terminal gráfico dará error. Ya ampliaremos sobre esto en temas posteriores, cuando veamos los distintos modo de arranque.

<sup>18</sup>Para ampliar sobre el tema se puede mirar en:

- La página man para
  - XFree86
  - XF86Config
  - La información del módulo que estemos usando en `/usr/X11R6/lib/X11/doc/`
- El HOWTO: *The Linux XFree86-Howto*
- *The Official Red Hat Linux Reference Guide*: Capítulo 6
- Los manuales/libros de Lucas:
  - *Linux Instalación y Primeros Pasos*, de MATT WELSH.
  - *Guía del enRootador de Linux*, de ERIC DUMAS.
  - *Linux Fácil*, una guía para introducirse en Linux.



Miremos este fichero en nuestro equipo

```
$less /etc/X11/XF86Config19
```

Comentemos brevemente cómo está organizado este fichero. El fichero `XF86Config` es un archivo de configuración del teclado, ratón, monitor, etc; está organizado en las secciones<sup>20</sup>:

### Sección `ServerLayout`

En ella definimos los dispositivos que componen nuestro display (es el nombre que recibe el conjunto de monitor, teclado y ratón). En el caso de disponer de varios, tendremos varias entradas como la que sigue<sup>21</sup>.

```
Section "ServerLayout"
    Identifier      "Anaconda Configured"
    Screen 0       "Screen0" 0 0
    InputDevice    "Mouse0" "CorePointer"
    InputDevice    "Mouse1" "SendCoreEvents"
    InputDevice    "Keyboard0" "CoreKeyboard"
EndSection
```

Con `Identifier` ponemos un nombre (único) a una sección de este tipo. La 2ª línea nos permite especificar las coordenadas absolutas que asignamos a la esquina superior izquierda de la pantalla (los dos últimos números, por defecto 0 0).

Con `InputDevice` especificamos los dispositivos de entrada; las opciones `Core*` indican aquellos que se usan por defecto.

### Sección `Files`

En ella podemos poner los path de las fuentes de nuestro sistema<sup>22</sup>, el path de la base de datos del color RGB o los módulos cargables del servidor X.

```
Section "Files"
    RgbPath "/usr/X11R6/lib/X11/rgb"
    FontPath "unix/:7100"
EndSection
```

Red Hat usa `xfs` (*X Font Server*, véase 7.6.4 en la página 116) como servidor de fuentes. Con `unix/:7100` indicamos al servidor X que se conecte al puerto 7100<sup>23</sup> de la máquina local para obtener la información que necesite sobre fuentes.

### Sección `ServerFlags`

Normalmente no aparece. Permite configurar algunas opciones globales del servidor X. Por ejemplo:

```
Section "InputDevice"
```

<sup>19</sup>Se sale pulsando la letra "q", de *quit* (salir).

<sup>20</sup>

- Las líneas de comentarios comienzan con el carácter #. En el fichero tendremos normalmente "dos tipos" de comentarios:

- Aclaraciones sobre qué significado tienen las líneas que siguen.
- Parámetros comentados que podremos activar si los descomentamos.

<sup>21</sup>En el caso de tener varias secciones como ésta, y no optar por una explícitamente cuando iniciamos el servidor X, pasándosela como argumento, se usa la primera que aparezca en este fichero.

<sup>22</sup>Hasta la versión 6.x de RedHat contenía un listado con los *pathnames* (nombres de directorios completos) de los archivos de fuentes y el archivo con la base de datos de los colores (archivo RGB). A partir de la versión 6.x, aunque mantiene el path completo del archivo con la base de datos de los colores, no es así con los directorios de las fuentes.

<sup>23</sup>Tiene que ver con las redes y el protocolo TCP/IP.



```
Option      "DontZap" "yes"
Option      "DontZoom" "no"
EndSection
```

Con DontZap activo (yes) se permite [CTRL]+[ALT]+[←] para finalizar el servidor X y con DontZoom a "no" conseguimos que no se acepten combinaciones especiales de teclas para cambiar la resolución de la pantalla (no podemos usar [CTRL]+[ALT]+[±]<sup>24</sup> para permutar de una resolución a otra).

### Sección Module

Se usa para especificar qué módulos del servidor X deben cargarse (/usr/X11R6/lib/modules). También permite especificar opciones a los módulos. Esta sección se ignora cuando el servidor X se construye en forma estática.

```
Section "Module"
    Load "dbe"          # Double-buffering
    Load "GLcore"       # OpenGL support
    Load "dri"          # Direct rendering infrastructure
    Load "glx"          # OpenGL X protocol interface
    Load "extmod"       # Misc. required extensions
    Load "record"      # X event recorder
    # Sólo son necesarias si no se usa xfs.
    Load "freetype"    # TrueType font handler
    Load "type1"       # Adobe Type 1 font handler
EndSection
```

### Sección InputDevice

Aquí se han incluido todas las secciones de los dispositivos de entrada. El fichero de configuración puede tener múltiples secciones InputDevice; normalmente habrá al menos dos: una para el teclado y otra para el ratón. Destacar que para disponer del teclado en castellano hemos tenido que añadir la opción "es"

```
Section "InputDevice"
    Identifier "Keyboard0"
    Driver     "keyboard"
    Option     "XkbRules" "xfree86"
    Option     "XkbModel" "pc102"
    Option     "XkbLayout" "es"
EndSection
```

La sección que sigue se corresponde con la configuración del ratón. En este ejemplo se trata de un ratón un tanto especial (en general será sólo PS/2 para los ratones estándar PS/2 o Microsoft para los ratones serie estándar). Además, se ha desactivado la opción de emular 3 botones, y con ZAxisMapping activamos la posibilidad de usar el botón central de este ratón para realizar desplazamientos en el eje Z. Pulsando hacia adelante desplazamos la página hacia arriba y pulsando en la parte de atrás del botón nos desplazamos hacia la página siguiente.

```
Section "InputDevice"
    Identifier "Mouse0"
    Driver     "mouse"
    Option     "Protocol" "NetMousePS/2"
    Option     "Device"   "/dev/mouse"
    Option     "Emulate3Buttons" "no"
```

---

<sup>24</sup>± del teclado numérico.



```

Option      "ZAxisMapping" "4 5"
EndSection

```

## Sección Monitor

Al menos debe haber una, para el monitor que se está usando (puede haber más si tenemos más de un monitor). En esta sección se incluyen las especificaciones del monitor, opciones específicas e información sobre los modos de vídeo. Además, éste es el lugar donde debemos incluir los valores de velocidad de sincronización horizontal y actualización vertical de nuestro modelo. Si no los ponemos, toma por defecto los valores de 28-33 y 43-72, respectivamente.

```

Section "Monitor"
    Identifier      "Monitor0"    VendorName      "HTC"
    ModelName       "b001"
    HorizSync       31.0 - 95.0
    VertRefresh     50.0 - 130.0
    # 640x480 @ 60 Hz, 31.5 kHz hsync
    Modeline "640x480"      25.175 640 664 760 800 480 491 493 525
    # 640x480 @ 75 Hz, 37.50 kHz hsync
    ModeLine  "640x480"      31.5 640 656 720 840 480 481 484 500 -HSync
-VSync
EndSection

```



Las líneas que con mayor frecuencia hay que modificar son las que se corresponden con la velocidad de sincronización horizontal y actualización vertical

```

HorizSync  rango_valores
VertRefresh  rango_valores

```

donde rango\_valores puede ser:

- Una serie de números separados por comas: en el caso de disponer de un monitor de frecuencia fija que admite varios valores.
- Un rango de valores separados por un guión: si nuestro monitor es multifrecuencia (los actuales lo son).

Las líneas `Modeline`<sup>25</sup> son generadas automáticamente por el programa `redhat-config-xfree86`. Aparecen para ajustar los modos de vídeo del monitor para unos determinados valores de refresco horizontal y vertical. En el caso de aparecer líneas de este tipo, el servidor X elegirá de entre todas las que representan la misma resolución, el mejor modo de vídeo admitido. Por ejemplo, si tuviésemos las dos líneas que aparecen anteriormente y si trabajamos con dos monitores de características:

### ■ FUJITSU FCM-246L

```

HorizSync 31.5, 35.2, 35.5, 37.8
VertRefresh 56, 60, 70, 72, 87

```

### ■ HITACHI CM643ET

```

HorizSync 31-95
VertRefresh 50-130

```

<sup>25</sup>No aparecen con todos los monitores. Para saber cómo se obtienen y qué significado tienen esos números, podéis consultar el libro publicado por editorial Anaya: *Red Hat Linux 5.2* de NABA BARKAKATI, página 382 y 383.



se daría que para una de resolución de 640x480 con el FUJITSU trabajaríamos con la primera línea:

```
# 640x480 @ 60 Hz, 31.5 kHz hsync
Modeline "640x480" 25.175 640 664 760 800 480 491 493 525
```

mientras que con el Hitachi se usaría la segunda:

```
# 640x480 @ 75 Hz, 37.50 kHz hsync
ModeLine "640x480" 31.5 640 656 720 840 480 481 484 500 -HSync
-VSync
```

## Sección Device

Es la sección relativa a la tarjeta de vídeo. La entrada `Driver` especifica el nombre del *driver* que controlará la tarjeta y es necesaria. La opción `BusID` es necesario especificarla cuando tengamos varios adaptadores de vídeo. Si sólo tenemos una tarjeta gráfica, es opcional.

```
Section "Device"
    Identifier "Card0"
    Driver      "ati"
    VendorName  "ATI"
    BoardName   "Mach64 GB"
    ChipSet     "ati"
    ChipId      0x4742
    ChipRev     0x5c
    #BusID      "PCI:1:0:0"
EndSection
```

## Sección Screen

Una pantalla representa la unión de un dispositivo gráfico (**sección Device**) y un monitor (**sección Monitor**). Podemos mantener varias secciones **Screen**, pero sólo usaremos aquella que se corresponda con el servidor gráfico que estemos usando. El servidor usado es el que aparece en la **sección Device** en el parámetro `Identifier`.

```
Section "Screen"
    Identifier "Screen0"
    Device     "Card0"
    Monitor    "Monitor0"
    DefaultDepth 16
    SubSection "Display"
        Depth 8
        Modes "800x600"
        Virtual 1024 768
    EndSubSection
    SubSection "Display"
        Depth 16
        Modes "1024x768" "800x600"
        ViewPort 0 0
    EndSubSection
EndSection
```

Con esta sección configurada según el ejemplo, entraríamos por defecto con una profundidad de color de 16 bpp y en un modo de vídeo de 1024x768. Después veremos que sin tener que reiniciar el sistema gráfico podemos pasar a una resolución de 800x600.

Iniciaríamos el servidor gráfico con 65.535 colores (16 bits de profundidad de color), si queremos modificar esto usaremos el parámetro

```
DefaultDepth x
```

antes de la primera subsección `Display`. Si deseamos que por defecto se inicie con 8 bpp (256 colores) pondríamos la línea

```
DefaultDepth 826
```

Para lo demás:

- `Depth x`: son los “bpp” (*bits por pixel*) que el servidor usará para generar los colores. El número `x` puede ser 8, 15, 16, 24 ó 32 dependiendo de la memoria de nuestra tarjeta gráfica.
- `Modes modo1 modo2 ...`: Lista de modos de vídeo que vamos a poder usar. El primero de la lista es el que se ejecuta por defecto. Para pasar de unos a otros lo podremos hacer usando **[CTRL]+[ALT]+[+]**.
- `ViewPort x y`: Posición inicial de la esquina superior izquierda de la pantalla inicial.
- `Virtual x y`: Permite disponer de una pantalla virtual mayor que el área de visualización. En este caso sólo tendríamos esa posibilidad cuando trabajemos con 800x600, en cuyo caso la pantalla será de 1024x768 aunque sólo veamos un área de 800x600.

## Sección DRI

Sólo se usa si nuestra tarjeta permite aceleración 3D por hardware<sup>27</sup> y está cargado el módulo `dri` en la sección `Module`.

### ➡ Para practicar: `redhat-config-xfree86`



Podemos editar un fichero de varias maneras, comentemos algunas:

#### En modo gráfico

- Usar el gestor de ficheros de GNOME. Se marca en vídeo inverso el fichero y tras pulsar sobre él con el botón derecho optamos por **Abrir con**.
- Desde un terminal:
 

```
#gedit
```
- Con los menús



→ **Accesorios** → **Editor de textos**

#### En modo alfanumérico: `mcedit`<sup>28</sup>.

1. Usando `redhat-config-xfree86` y modificando los ficheros de configuración del servidor X conseguir:
  - que el sistema arranque con 24 bpp (o con la máxima que permita nuestro equipo)
    - con sólo los modos de 800x600 y 1024x768
    - el modo por defecto ha de ser de 800x600
    - pantalla virtual de 1024x768

1. Podemos conocer la configuración del servidor x usado, ejecutando desde un terminal gráfico el comando:

```
$xset -q
```



**Vesa** Como se comentó al usar `redhat-config-xfree86`, hay tarjetas de vídeo que no están soportadas aún por la versión de XFree86 disponible. Para este caso hay un servidor X que funciona con la mayoría de tarjetas de última hornada. La condición indispensable es que nuestra tarjeta sea compatible con el estándar VESA 2.0. Si ésta es nuestra única opción, podemos consultar en <http://www.xfree86.org/4.2.0/vesa.4.html> la forma de configurarlo.

<sup>26</sup>Más adelante veremos cómo hacer esto para una determinada sesión gráfica

<sup>27</sup>Véase la página web antes comentada y el fichero `/usr/X11R6/lib/doc/README.DRI`.

<sup>28</sup>Es el editor interno del programa `mc`. También podéis usarlo cargando el programa `mc` y después, con el fichero seleccionado usar la tecla **F4**.

### 7.3. Otros gestores de ventanas

Hasta ahora, lo normal es que después de instalar Linux tan sólo se haya trabajado con los escritorios GNOME y KDE (este último, sólo en el caso de que se haya instalado). Pero ambos, aunque son potentísimos, presentan un grave problema: “son dos devoradores de recursos”. Esto no es problema para los equipos actuales, sin embargo para los “viejos 486/586” con “poca memoria RAM” sí que puede serlo. Con estos equipos es mejor trabajar con gestores de ventanas (tipo *WindowMaker*) y no con entornos de escritorio ya que necesitan menos recursos para funcionar adecuadamente en modo gráfico.


Antes ver cómo trabajar con algunos gestores de ventanas, veamos una utilidad interesante:

#### 7.3.1. Utilidad switchdesk.

Con él podemos cambiar el gestor de ventanas/escritorio a usar por defecto para ese usuario.

Podemos ejecutarlo tanto en modo texto como en modo gráfico. Si optamos por usar la línea de comandos escribiremos

```
$switchdesk
```

o bien ejecutar el programa desde  → **Extras** → **Configuración del Sistema** pulsando sobre **Desktop Switching Tool** y obtendremos<sup>29</sup>:



Para cambiar el “aspecto” de nuestro entorno gráfico tendremos que salir de él y volver a entrar. Por ahora es mejor dejarlo así hasta el próximo apartado. Pulsaremos sobre **OK** y dejaremos las cosas como están.<sup>30</sup>

#### ➡ Para practicar: El gestor de ventanas TWM

Uno de los clásicos dentro de los gestores de ventanas es el **twm**. Es un gestor de ventanas pensado para minimizar el consumo de memoria y no demasiado fácil de usar.

Ejecutemos `switchdesk` y tras seleccionar TWM, reiniciemos el sistema gráfico.

- ¿Qué se ha hecho? si editamos el fichero `.Xclients-default`, en vez de la línea

```
exec gnome-session31
```

<sup>29</sup>El número de opciones depende de los gestores de ventanas/escritorios que tengamos instalados

<sup>30</sup>Es importante que pulsemos sobre OK y que no cerremos la ventana pulsando sobre la esquina superior derecha. El sentido de hacerlo así es que de esta forma crearemos dos ficheros necesarios para seguir “bien” la entrega. Estos ficheros son:

```
.Xclients
```

```
.Xclients-default
```

Podemos ver su contenido, por ejemplo, con:

```
$cat ~/.Xclients-default
```

<sup>31</sup>o si es el KDE:

```
exec starkde
```

switchdesk ha escrito la línea:

```
exec /usr/X11R6/bin/twm
```

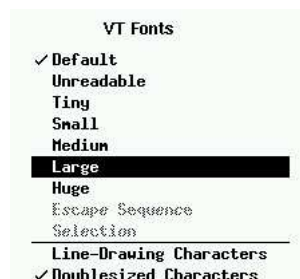
Al reiniciar el entorno gráfico, veremos una pantalla azul (demasiado limpita ¿no?). Vamos a intentar arreglar este desaguisado. Pulsemos con el botón izquierdo del ratón (sin soltarlo) sobre cualquier zona del escritorio y optemos por abrir un xterm:



y obtendremos nuestro terminal, que podemos ver a continuación:



- Veamos cómo conseguir que las fuentes sean un poco más grandes:
  - Manteniendo la tecla **[Ctrl]** pulsada, y el botón derecho del ratón en una zona de la ventana del terminal gráfico, aparecerá el menú



ya sólo tenemos que optar por el tamaño.

- Para dejar las cosas como estaban tan sólo tenemos que usar de nuevo:

```
$ switchdesk
```

y seleccionar GNOME. Después saldremos del gestor de ventanas usando el menú anterior y pulsando sobre **Exit**.

Si nuestro ordenador no es excesivamente lento, podemos usar otros gestores de ventanas más evolucionados y de más fácil configuración, tales como AfterStep o WindowMaker<sup>32</sup>. Estos gestores de ventanas también se pueden configurar *a pelo*, pero traen una especie de "panel de control" con el que se pueden configurar a base de botones en un entorno gráfico. Hay otros muchos gestores, como el Enlightenment<sup>33</sup>, Icewm, Bowman, etc. Podemos elegir aquel que mejor se adapte a nuestro gusto y posibilidades.

De los anteriores, y aunque no venga en los CDs hay uno que merece mención especial ya que necesita muy pocos recursos y es altamente configurable, se trata de IceWM<sup>34</sup>.

### ➡ Para practicar: Instalación de WindowMaker

Para disponer de WindowMaker hay que instalar los paquetes (y en este orden) del CD3, ficheros y librerías del gestor de ventanas<sup>35</sup>:

```
WindowMaker-libs-0.80.1-1.i386.rpm
```

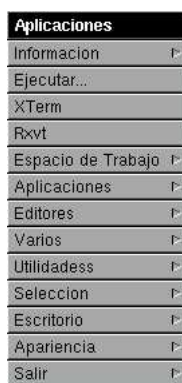
```
WindowMaker-0.80.1-1.i386.rpm
```

Veamos una forma de hacerlo en modo texto (en modo gráfico se puede hacer con Nautilus, el gestor de ficheros de GNOME):

1. Introduzcamos el tercer CD de la distribución en la unidad lectora y montémoslo:
 

```
$mount cdrom
```
2. Instalemos los paquetes que deseamos<sup>36</sup>:
 

```
#rpm -ivh /mnt/cdrom/RedHat/RPMS/WindowMaker*
```
3. Ejecutemos switchdesk y optemos por WindowMaker. Cuando volvamos a entrar como este usuario en modo gráfico, estaremos trabajando con WindowMaker. Para obtener el menú principal del programa, pulsar sobre el escritorio con el botón derecho del ratón.



<sup>32</sup>Este último viene en los CDs de Redhat

<sup>33</sup>En el CD2

<sup>34</sup>Quizá sea por hoy por hoy el mejor gestor de ventanas para máquinas con no demasiados recursos. No viene en los CDs de RedHat

<sup>35</sup>

- Si optamos por otro tipo de instalación en la primera entrega puede que tengamos problemas de dependencias con otros paquetes.
- Ya hay actualizaciones para Red Hat de ambos paquetes.

<sup>36</sup>En entregas posteriores explicaremos detenidamente qué significa esto. Se ha supuesto que hemos introducido el CD en el "primer lector del que disponemos". Si tenemos dos o más puede que haya que sustituir `cdrom` por `cdrom1`, `cdrom2`, ...

### ➡ Para practicar: Instalación de Icewm

El último y quizás el mejor actualmente para equipos sin demasiados recursos. No viene en los CDs así que hay que bajarlo desde internet:

- Están en la página del curso, sección downloads, o en
- [http://icewm.sourceforge.net/icewm-1\\_2\\_7.php](http://icewm.sourceforge.net/icewm-1_2_7.php)

Necesitamos los ficheros:

```
icewm-1.2.7-1.i386.rpm  
icewm-themes-1.2.7-1.i386.rpm  
icewm-l10n-1.2.7-1.i386.rpm  
icewm-menu-gnome1-1.2.7-1.i386.rpm
```

Una vez en nuestro equipo, tendremos que instalarlos. Para eso, una vez en el directorio que los contiene ejecutemos:

```
#rpm -ivh icewm*
```


Para ejecutarlo no podemos usar la utilidad `switchdesk`, hay que hacerlo a mano. Con un editor de textos de Linux (por ejemplo `gedit`<sup>37</sup>) modifiquemos el fichero `.Xclients-default`<sup>38</sup> de nuestro subdirectorio de usuario como sigue:

```
exec /usr/bin/icewm
```

El resto de líneas han de estar comentadas (por eso se inician anteponiéndoles el carácter #).

Listo, la próxima vez que entremos en modo gráfico trabajaremos con este gestor de ventanas.

## 7.4. Varias sesiones abiertas

 Desde este momento supondremos que entráis al ordenador en modo alfanumérico<sup>39</sup>. Para eso podemos ejecutar desde un terminal gráfico la orden

```
$su
```

Se nos pedirá la contraseña del root. Cuando se la escribamos, y si todo está bien, ejecutaremos el comando<sup>40</sup>:

```
#!/sbin/init 3
```

Ampliemos un poco sobre

```
$ switchdesk
```

Como ya hemos comentado, con él podemos cambiar el gestor de ventanas/escritorio a usar por defecto para ese usuario.

En el resto de este apartado, usaremos un terminal de texto para ejecutarlo y/o el menú de GNOME. Si desde un terminal gráfico ejecutamos el programa nos aparecerá<sup>41</sup>

<sup>37</sup> **Menú principal → Accesorios → Editor de textos**

<sup>38</sup> Hay que tener en cuenta que los ficheros que comienzan con punto son los “ficheros ocultos” de linux. Si estamos usando el gestor de ficheros de GNOME (Nautilus) no los veremos salvo que lo activemos desde el menú:

**Editar → Preferencias → Vistas icono y listas** y marcar en **Opciones de mostrar** el que veamos **Archivos Ocultos**.

<sup>39</sup> Si no es así, no funcionará lo aquí expuesto

<sup>40</sup> No os preocupéis si no se entiende. Ya se explicará en entregas posteriores.

<sup>41</sup> El número de opciones depende de los gestores de ventanas/escritorios que tengamos instalados



Para cambiar el “aspecto” de nuestro entorno gráfico tendremos que salir de él y volver a entrar. Merece la pena pararse un poco en la opción *Cambie tan solo las aplicaciones de la pantalla actual* y en el texto superior de la ventana: *La pantalla actual es novo:0*<sup>42</sup>

### ¿Qué es eso de *La pantalla actual es novo:0*?

Veamos qué significado tiene esto. Lo primero que hay que hacer notar es que *novo*<sup>43</sup> es el nombre de la máquina donde se ha capturado el gráfico, así que si se hubiese capturado en la máquina *gandalf*, en vez de *cursolinux* tendríamos: *La pantalla actual es gandalf:0*. Tras aclarar esto, no perdamos de vista lo importante y lo importante son los números que siguen al nombre de la máquina.

### ¿Qué pasa si pulso en: *Cambie tan solo las aplicaciones de la pantalla actual*?

Explicuemos esto con un ejemplo, supongamos que el usuario THALES que trabaja en la máquina CICA ha entrado en Linux EN MODO TEXTO y accede a una sesión gráfica de GNOME con la orden

```
$ startx -- :1
ejecuta
$ switchdesk44
```

selecciona TWM y marca la casilla *Cambie tan solo las aplicaciones de la pantalla actual*. Nuestro usuario THALES sale del entorno gráfico y vuelve a entrar, pero ahora, no pone qué terminal gráfico va a usar y sólo escribe

```
$ startx
```

Desalentado ve que sigue trabajando con GNOME y que no se ha producido ningún cambio. Ejecuta de nuevo

```
$ switchdesk45
```

¿pero qué está pasando?, ahora ve que aparece *cica:0* mientras que antes aparecía *cica:1*. Ahí está la respuesta: al marcar que apliquemos los cambios sólo para ese *display*<sup>46</sup> eso es lo que hemos hecho. Es decir, siempre que entremos en el display 0 tendremos GNOME como escritorio de trabajo mientras que si usamos la orden

```
$ startx -- :1
```

y trabajamos con el display 1, usaremos como gestor de ventanas TWM.

Es interesante comprobar los cambios que se producen en el subdirectorío del usuario. En este momento deberían existir al menos los ficheros<sup>47</sup>:

<sup>42</sup>Si se utiliza este programa desde KDE aparece *nombre\_maquina:0.0* en vez de *nombre\_maquina:0*.

<sup>43</sup>Si no hemos cambiado el nombre de la máquina debería aparecer: *localhost.localdomain*.

<sup>44</sup>Ahora vería *Current display is cica:1*

<sup>45</sup>Ahora vería *Current display is cica:0*

<sup>46</sup>Se le llama *display* al conjunto formado por un monitor, un ratón y un teclado.

<sup>47</sup>Los ficheros que empiezan por un punto son ficheros “ocultos”, ya hemos comentado cómo verlos desde el gestor de archivos de GNOME. También podemos usar la orden

```
$ ls -a
```

Desde el programa *mc* también es posible “verlos”



```
.Xclients
.Xclients-default
.Xclients-cica:1
```

De los tres ficheros, el primero no es modificado por el programa. El segundo sólo se modifica si no marcamos *Cambie tan solo las aplicaciones de la pantalla actual* y el tercero se ha creado/modificado al seleccionarlo.

Magnífico ¿verdad?, un usuario puede tener todos los gestores de ventanas que desee y cargar en cada momento aquel que más le interese.

Sigamos con el tema. Nuestro intrépido THALES es además el root del sistema, está trabajando en modo gráfico con el escritorio GNOME, como es muy precavido sólo entra como root cuando es absolutamente necesario y ahora lo es, para esto pulsa **[CTRL]+[ALT]+[F2]** y puede entrar como otro usuario, entra como root con la contraseña correspondiente y desea trabajar con su GUI favorita.

Ejecuta

```
# startx
```

y ¿qué pasa ahora?, no puede trabajar en modo gráfico, le aparece que ese entorno está ya en funcionamiento y se le queda bloqueada la pantalla, para salir pulsa **[CTRL]+[C]** y le aparece de nuevo el prompt (símbolo del sistema) de entrada de órdenes. Comprueba qué displays tiene activos<sup>48</sup> y ve los que están trabajando, ante esta situación escribe

```
startx -- :x
```

donde x es un número comprendido entre 0 y 9 que no esté ya en uso. En el supuesto de que tan sólo tenga uno en acción una posible alternativa sería

```
# startx -- :1
```

Ya está, tenemos dos terminales gráficos en acción.

Pero no termina aquí la cuestión, nuestro intrépido Linuxero tiene un problema, cuando está trabajando como root, además de querer trabajar en modo gráfico desea hacerlo con la máxima resolución posible. Necesita hacer algunas “cosillas” de retoque con el Gimp<sup>49</sup> y quiere tener el máximo de colores a su disposición. Sale del entorno gráfico y, como su tarjeta permite 32 bits de profundidad de color, escribe

```
# startx -- -bpp 32 :1
```

Además, normalmente trabaja con una resolución de 640x480 pero ahora le interesa cambiarla a 800x600. Para esto, una vez que está en el entorno gráfico, pulsa **[CTRL]+[ALT]+[+]** y puede ir permutando los modos de vídeo disponibles para esa profundidad de color. Si en un momento determinado quiere deshacer el cambio en el modo de vídeo pulsa **[CTRL]+[ALT]+[-]** y vuelve al modo previo.

Para pasar al terminal gráfico de Thales, tan sólo tiene que pulsar **[CTRL]+[ALT]+[F7]** y para volver al del root **[CTRL]+[ALT]+[F8]**.

Si abriese otra sesión gráfica, ¿cuál sería?. Está claro, **F9**<sup>50</sup>.

Ahora, nuestro amigo desea trabajar en una sesión en modo texto. Para esto, desde el terminal gráfico pulsa **[CTRL]+[ALT]+[F3]** y le aparece de nuevo

```
login:
```

tras introducir su nombre de usuario y su contraseña, la situación sería:

- Dos terminales gráficos a los que puede acceder usando **[CTRL]+[ALT]+[F7]** y **[CTRL]+[ALT]+[F8]**
- Un terminal de texto al que accede desde los terminales gráficos con **[CTRL]+[ALT]+[F3]**

<sup>48</sup>Una posible forma de verlo es usando

```
$ ls /tmp/.X11-unix/
```

el resultado de esta orden es

```
X0=      X1=      etc
```

donde cada número indica una sesión gráfica abierta.

<sup>49</sup>Programa de retoque fotográfico

<sup>50</sup>Podemos modificar la secuencia natural de asignación de las teclas de función usando el parámetro `vtI` donde `I` es un número comprendido entre 7 y 12. Por ejemplo, con:

```
$ startx -- :1 vt11
```

se abriría una sesión gráfica accesible usando la combinación de teclas: **[CTRL]+[ALT]+[F11]**.



Hay que hacer notar que si usamos sólo terminales de texto la forma de pasar de unos a otros es usando tan sólo **[ALT]+[Fi]**  $i=1..6$ . Es decir, podemos tener hasta 6 terminales de texto abiertos simultáneamente. ¿Cuántos terminales gráficos podemos tener? la respuesta también está clara, desde **F7** a **F12**, es decir, 6.

Merece la pena comentar un “problema” que se presenta si usamos la orden `switchdesk` y marcamos la opción *Cambie tan solo las aplicaciones de la pantalla actual* en vez de desde GNOME, desde el KDE o TWM, en ese caso, los ficheros que se generan en el subdirectorio del usuario son de la forma:

```
.Xclients-cica:x.051
```

donde `x` representa un número y cuando después ejecutamos

```
startx -- :x
```

no “encuentra” el fichero necesario, y , por tanto, los cambios no tienen efecto. Una forma de arreglarlo es renombrando esos ficheros de manera que sí funcione todo, la forma de hacerlo sería:

```
$ mv .Xclients-cica:x.0 .Xclients-cica:x
```

### ➡ Para practicar: Conseguir la situación

- Tres terminales gráficos abiertos con gestores de ventanas diferentes a los que se pueda acceder usando **[CTRL]+[ALT]+[Fi]** con  $i = 7..9$
- Un terminal de texto al que acceder desde los terminales gráficos con **[CTRL]+[ALT]+[F2]**

### Un par de añadidos para “nota”

Por último vamos a adelantarnos al tema de redes y para aquellos que tengan acceso a una red local es interesante probar la posibilidad que nos ofrece Linux de trabajar en modo gráfico con programas situados en otro equipo, para esto tendremos que:

- Usando telnet:

Desde un Xterm de la máquina local ejecutaremos

```
$ xhost +máquina_remota52
```

después haremos un telnet a la máquina remota y una vez conectados escribiremos

```
$ export DISPLAY=máquina_local:053
```

por último ya sólo tenemos que ejecutar el comando que deseemos, por ejemplo, podéis probar con

```
$ mozilla &54
```

- Usando ssh

más fácil todavía, sólo hay que autenticarse ante la máquina remota y ejecutar el programa gráfico que deseemos.

## 7.5. Ideas para configurar WindowMaker e Icewm

Estos dos gestores de ventanas son altamente configurables y se consigue hacerlo con facilidad.

<sup>51</sup>Recordar que estamos trabajando con la máquina CICA. Si no habéis cambiado el nombre de vuestra máquina aparecerá:

```
.Xclients-localhost.localdomain:x.0
```

<sup>52</sup>Donde `máquina_remota` es o bien la dirección IP de la máquina remota, o bien, el nombre de esa máquina

<sup>53</sup>Con respecto a `máquina_local` el mismo comentario que con `máquina_remota`. Con respecto a `:0` aplicar lo visto en apartados anteriores.

<sup>54</sup>El `&` se utiliza para ejecutar el comando en segundo plano y que no nos ocupe la ventana de terminal hasta su finalización.



### 7.5.1. WindowMaker

La forma más sencilla de configurar WindowMaker es en modo gráfico. Pulsando sobre el icono  accederemos a la ventana



que nos permite configurarlo sin dificultad y con los menús en castellano.

Los ficheros de configuración de este gestor de ventanas se guardan en el directorio `$HOME55/GNUstep`.

Ejemplifiquemos cómo añadir el programa `gEdit` al submenú **Aplicaciones** → **Editores** sin hacer uso de las utilidades anteriores. Conseguirlo es tan sencillo como editar el fichero `$HOME/GNUstep/Defaults/WMRootMenu` y añadir la línea

```
, ("gEdit", EXEC, "gedit")
en la parte correspondiente a Aplicaciones (cuidado con la coma)
("Editores,
....
("VI", EXEC, "xterm -e vi"),
("gEdit" EXEC, gedit)
),
...
```

Si abrimos ese submenú veremos que ya se ha añadido ese nuevo elemento.

### 7.5.2. Icewm

Es uno de los mejores gestores de ventanas que podemos encontrar en la actualidad, sobre todo si trabajamos con una máquina con no demasiados recursos. Podemos configurar casi todos los aspectos de este gestor de ventanas de forma gráfica, pero para eso hay que bajar alguna de las utilidades disponibles en internet. Para encontrarlos, sólo hay que entrar en <http://freshmeat.net> y buscar `icewm`.

Los ficheros de configuración de esta versión se localizan en `/usr/share/icewm`. Se trata de:

**keys** atajos de teclado

**menu** para configurar el menú principal

**preferences** fichero de configuración principal

**toolbar** para ver los iconos de las aplicaciones de la barra de tareas

**winoptions** configuración de los distintos elementos mostrados en pantalla

<sup>55</sup>`$HOME` se refiere al directorio personal de ese usuario. Por ejemplo, si el nombre de usuario es `paco`, este directorio sería `/home/paco`



Para no sobrecargar más el tema os remitimos a:

- Una guía en castellano de uso y configuración de este gestor de ventanas: IceWM para principiantes <http://www.icewm.org/files/es/guide/><sup>56</sup>, en ella se detalla qué significado y función tiene cada uno de los ficheros anteriores.
- ICEWM: El Window Manager <http://pinsa.escomposlinux.org/sromero/linux/icewm/>, artículo de SANTIAGO ROMERO sobre IceWM.

## 7.6. Fichero `$HOME/.Xresources`

El proceso de carga de un determinado gestor de ventanas es diferente si hemos iniciado un sistema con Red Hat Linux en modo gráfico o en modo texto<sup>57</sup>. Notar que en el inicio en modo gráfico se carga el sistema X antes de que el usuario se autentifique ante el sistema, mientras que en modo texto la autenticación es previa a la carga del gestor de ventanas (se inicia después con `startx`).

Analicemos qué ocurre en cada caso<sup>58</sup>:

### 7.6.1. Modo texto

Cuando, estando en nivel 3, se inicia un gestor de ventanas o escritorio con:

```
$startx59
```

se inicia un proceso que se puede resumir en la lectura y ejecución de los ficheros<sup>60</sup>:

1. el fichero `$HOME/.xinitrc` es un script que se transmite a `xinit` al ejecutar el comando `startx`. Inicia algunas opciones globales sobre el entorno gráfico para definir qué clientes X se deben ejecutar. Si no existe, utiliza el script por defecto del sistema: `/etc/X11/xinit/xinitrc`.
2. `xinitrc` busca entonces los ficheros: `$HOME/.Xresources`<sup>61</sup>, `$HOME/.Xmodmap`, y `$HOME/.Xkbmap`<sup>62</sup>, y si no están, utiliza los generales del sistema del directorio `/etc/X11`
3. Después, `xinitrc` ejecuta los scripts del subdirectorio `/etc/X11/xinit/xinitrc.d`.<sup>63</sup>
4. Por último, ejecuta el fichero `$HOME/.Xclients` (si no existe, `/etc/X11/init/Xclients`). Es el script encargado de controlar qué gestor de ventanas o escritorio se va a usar. Llama al fichero `$HOME/.Xclients-$HOSTNAME$DISPLAY` si le pasamos al comando `startx` el número de display a usar, o bien, al gestor de ventanas por defecto (`$HOME/.Xclients-default`)

### 7.6.2. Modo gráfico

En este caso, son los propios ficheros de configuración de inicio del sistema los que se encargan de gestionar qué programa gráfico de configuración vamos a usar (por defecto `gdm`) determinado por el fichero `/etc/X11/prefdms`. Una vez cargada la pantalla gráfica de autenticación (configurada en función de `/etc/X11/xdm/Xsetup_0`) se inicia el script `/etc/X11/xdm/GiveConsole` para que el usuario que quiere acceder sea el “dueño” de esa consola<sup>64</sup>. El papel que jugaba `xinitrc` (véase 7.6.1) en el caso

<sup>56</sup>En la web del curso también estará.

<sup>57</sup>Niveles de ejecución 5 ó 3. Ya los estudiaremos después.

<sup>58</sup>Para ampliar podéis mirar el Capítulo 6 de *The Official Red Hat Linux Reference Guide*

<sup>59</sup>front-end del programa `xinit`

<sup>60</sup>No aparecen todos los ficheros implicados.

<sup>61</sup>En versiones anteriores este fichero se llamaba `$HOME/.Xdefaults`. De hecho si en vez del fichero `$HOME/.Xresources` creamos un fichero de nombre `$HOME/.Xdefaults` todo funciona igual. Por otro lado, si están los dos, se carga el que estamos estudiando.

<sup>62</sup>Estos dos últimos se usan por `xmodmap` para configurar el teclado

<sup>63</sup>El fichero `xinput` que permite seleccionar el idioma, programa gráfico de inicio, etc

<sup>64</sup>Mientras esté utilizándola, será el que pueda leer y escribir en ella.

anterior lo toma ahora `/etc/X11/xdm/Xsession` con casi las mismas funciones que las comentadas anteriormente<sup>65</sup>.

Cuando finalizamos el trabajo como ese usuario, se ejecuta `/etc/X11/xdm/TakeConsole` para que sea de nuevo el root el dueño y señor de esa consola, reiniciándose el gestor gráfico de autenticación para iniciar en su caso una sesión de usuario nueva.

### 7.6.3. \$HOME/.Xresources



**Programa gv** Para ejemplificar este apartado, vamos a usar un visor de ficheros PostScript que no se carga por defecto, se trata del programa `gv` (paquete `gv-3.5.8-18.i386.rpm` del CD3). Por tanto antes de seguir hay que montar el CD3 e instalar el paquete:

```
#mount /mnt/cdrom
#rpm -ihv /mnt/cdrom/RedHat/RPMS/gv-3.5.8-18.i386.rpm
```

Las aplicaciones X permiten bastantes opciones cuando se cargan y de nosotros depende en gran medida el aspecto que queremos que tengan. Con el fichero `$HOME/.Xresources` podemos modificar los parámetros y valores iniciales de las aplicaciones instaladas en nuestra máquina. Con este fichero podemos hacer que esos parámetros<sup>66</sup> se pasen al programa a ejecutar para cualquier escritorio o gestor de ventanas.

A los parámetros que podemos pasar a una aplicación para modificar su apariencia o comportamiento, se les llama *recursos*. Por ejemplo, recursos pueden ser: el tamaño inicial con que se carga un programa, los colores de primer plano o de fondo, las fuentes de letras, etc. Los usuarios pueden especificar recursos en diferentes lugares. Al conjunto de todas estas especificaciones se denomina *base de datos de recursos*. Los recursos de una aplicación se buscan según un esquema predefinido. A continuación se listan algunos de los lugares en que se buscan esos recursos. El listado no está completo (en cuanto a todos los lugares donde se busca) pero sí que se ha mantenido el orden de búsqueda<sup>67</sup>:

1. En el directorio `/usr/lib/X11/app-defaults`. Aquí hay una serie de archivos que permiten modificar los recursos de esas aplicaciones. En general, es mejor no tocar estos archivos y realizar las modificaciones para cada cliente en los que se exponen a continuación.
2. El fichero `.Xresources` en el directorio de usuario.
3. El fichero `.Xresources-host` en el directorio de usuario.
4. Recursos definidos en la línea del comando.

El archivo `.Xresources` es un archivo de texto en donde se definen los recursos y sus valores (uno por línea). Las líneas de este archivo son de la forma:

```
aplicacion.variable: valor
donde:
```

**aplicación** nombre de la aplicación.

**variable** recurso a especificar

**valor** valor asignado a ese recurso

<sup>65</sup>Notar que podemos optar por el gestor de ventanas para esta sesión desde la pantalla gráfica de autenticación. Si no lo hacemos, se seguirá el esquema de 4 en 7.6.1.

<sup>66</sup>Otra forma consiste en pasarle los parámetros que deseamos cargar directamente, es decir, desde una xterm (ventana de terminal utilizada en modo gráfico) escribiríamos:

```
$ gv -geometry 620x450
```

<sup>67</sup>De nuevo este tema daría por sí solo para un curso, nosotros solamente vamos a poner lo que por ahora está en nuestras manos poder modificar.

Inicialmente sólo el root tiene un fichero como éste en su directorio de usuario<sup>68</sup>. Forma parte del paquete `rootfiles-7.2-4.noarch.rpm`. Lo mejor sería copiar este fichero al \$HOME de un usuario<sup>69</sup> y trabajar con este último manteniendo el del root sin tocar:

```
# cp /root/.Xresources /home/usuario
# chown usuario /home/usuario/.Xresources
# chgrp usuario /home/usuario/.Xresources70
```

Los recursos están organizados en clases. Esta jerarquía de objetos refleja la jerarquía de ventanas de la aplicación. El nombre de una clase (empieza por mayúscula) indica la categoría general de una aplicación, mientras que cada recurso individual se le conoce como instancia (empiezan por una letra minúscula).

No sólo podemos separar los componentes con el carácter “.”, sino que podemos separarlos usando \* para indicar que entre ellos puede haber cualquier número de componentes. También se puede usar el carácter ? que indica un componente con cualquier nombre.

Para explicar este galimatías podéis echar un vistazo a vuestro fichero `/root/.Xresources`. Además, hay que mirar la documentación de cada aplicación para conocer qué recursos utiliza.

Algunos de los recursos más usuales son:

Descripción	Clase	Instancia	Opciones en línea de comandos
Color del fondo	Background	background	-bg
Color del borde	BorderColor	borderColor	-bd
Ancho del borde	BorderWidth	borderWidth	-bw
Color del primer plano	Foreground	foreground	-fg
Fuente	Font	font	-fn
Tamaño de la ventana y posición	Geometry	geometry	-geometry
Título de la ventana	Title	title	-title



Algunas aplicaciones que permiten configurar la apariencia de nuestro entorno gráfico son<sup>71</sup>:

**xset** permite controlar la repetición de teclas, el pitido del teclado, etc. (Ya ha aparecido `$xset -q`)

**xsetroot** permite controlar la apariencia de la ventana principal.

Pone el fondo del escritorio en azul (desde GNOME no funciona)

```
$xsetroot -solid blue
```

**xmodmap** permite configurar los “mapas” del teclado y del ratón.

Parar conocer cómo se están asignado los botones del ratón.

```
$xmodmap -pp
```

Veamos algunos ejemplos de cómo lograr que algunas aplicaciones se carguen a nuestro gusto.

En todos los ejemplos que siguen, las opciones se están escribiendo en el supuesto de que las ponemos en el archivo `.Xresources`. Lo mejor, para practicar, es escribirlas en la línea de comandos (desde una `xterm`) y comprobar los cambios antes de modificar dicho archivo. Para cada ejemplo se ha creado una nota a pie de página de cómo se debería escribir desde una `xterm`.

1. Supongamos que deseamos que el visor `gv` se cargue siempre con el tamaño definido anteriormente, independientemente del gestor de ventanas/escritorio que usemos. En ese caso añadiríamos al fichero `.Xresources` la línea:

<sup>68</sup>Si deseamos crear un fichero “genérico” para todos los usuarios podemos modificar este fichero a nuestro gusto y copiarlo al subdirectorio `/etc/skel`. Así cada vez que creamos un usuario tendrá su fichero de recursos.

<sup>69</sup>El que hayamos creado en nuestro sistema.

<sup>70</sup>Por ahora, sólo debéis preocuparos de que hemos copiado este fichero del directorio del root al de un usuario, y hemos hecho que el fichero pertenezca a este usuario.

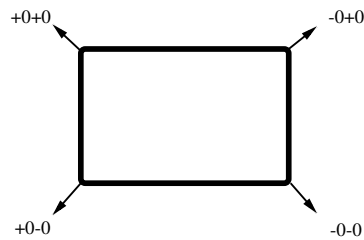
<sup>71</sup>Para conocer sus posibilidades os remitimos a las páginas `man` de cada uno de ellas

```
gv*geometry: 620x450+100+10072
```

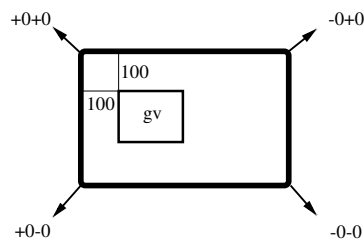
Si modificamos el archivo `.Xresources` y queremos que sean efectivos los cambios sin reiniciar el gestor de ventanas escribiremos desde un terminal la orden:

```
$ xrbdb .Xresources
```

Con ese valor para *geometry* indicamos que la ventana inicial del visor *gv* sea de 620x450 pixels. Los otros dos números indican en qué posición se cargará la aplicación. Cada una de las esquinas de nuestro monitor tienen de coordenadas:



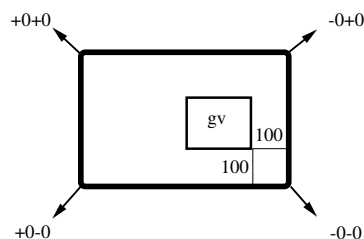
A partir de ese sistema de referencia, con el valor anterior conseguiríamos que nuestro programa se cargase 100 pixels hacia la derecha y hacia abajo desde el punto +0+0,



si por el contrario escribimos:

```
gv*geometry: 620x450-100-10073
```

conseguiríamos el mismo efecto pero ahora desde el punto -0-0 y teniendo en cuenta que en este caso se desplazaría la ventana hacia arriba y hacia la izquierda:



Por ultimo, los desplazamientos no tienen por qué ser iguales, así, si escribimos<sup>74</sup>:

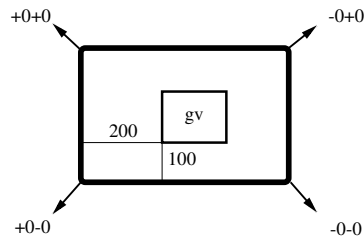
```
gv*geometry: 620x450+200-10075
```

<sup>72</sup>\$ gv -geometry 620x450+100+100 &

<sup>73</sup>\$ gv -geometry 620x450-100-100 &

<sup>74</sup>Más aún, probad con - -100- -100, esto obliga a que la aplicación se salga de la pantalla.

<sup>75</sup>\$ gv -geometry 620x450+200-100 &



2. Sin embargo, con las Xterm<sup>76</sup> no tenemos que poner el ancho y el alto en pixels sino que se deben especificar las filas y columnas que queremos que tenga nuestro terminal, por ejemplo:

```
xterm*geometry: 80x2577
```

nos permite especificar que todas las xterm que arranquemos tengan 80 caracteres por 25 líneas de pantalla. Si además queremos que la fuente que nos cargue por defecto sea otra podemos escribir<sup>78</sup>:

```
xterm*font: 10x2079
```

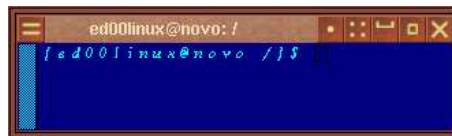
Tendremos que tener cuidado con modificar ambos valores a la vez (la geometría y las fuentes) ya que los dos inciden sobre el tamaño de la ventana.

3. Modificación de colores<sup>80</sup> y fuentes.

Si ejecutamos desde una xterm el comando:

```
$xterm -geometry 30x5 -fn '-b&h-lucidabright*' -bg navy -fg cyan &
```

podemos conseguir una xterm tan poco práctica como:



## 7.6.4. Fuentes en Linux

### Nomenclatura

En el ejemplo anterior nos ha aparecido el nombre de una fuente, a saber la fuente `-b&h-lucidabright*`. En Linux el nombre de una posible fuente es de la forma:

```
-adobe-courier-bold-r-normal--14-100-100-100-m-90-iso8859-1
```

El nombre de cada fuente consta de 12 campos. El significado de cada campo es:

- Suministrador de la fuente (adobe, b&h, etc).
- Familia (courier, lucida, times, etc)
- Peso (bold, medium, etc)

<sup>76</sup>Con las xterm hay más formas de acceder a menús que permiten su modificación. Una consiste en mantener pulsada la tecla **Ctrl** y pulsar con el ratón sobre esa xterm. Con ambos botones nos aparecen menús distintos que permiten, desde cambiar el tipo de fuente, a salir de ellas.

<sup>77</sup>\$ xterm -geometry 80x25 &

<sup>78</sup>Hay que comentar/modificar la línea

```
xterm*font: fixed
```

<sup>79</sup>\$ xterm -fn 10x20 &

<sup>80</sup>Para saber los nombres de los colores y no tener que usar códigos hexadecimales podéis consultar el archivo `/usr/X11R6/lib/X11/rgb.txt`.





- Inclinación (r=roman, itálica, etc)
- Ancho del conjunto (normal, semicondensada, etc)
- Altura de los caracteres (en pixels).
- Tamaño de la fuente (en decenas de puntos).
- Resolución horizontal (en puntos por pulgada).
- Resolución vertical (en puntos por pulgada).
- Espaciado (monoespaciado, proporcional, etc).
- Anchura media (en decenas de pixels).
- Conjunto de caracteres base (iso8859-1,etc) .

Se pueden usar comodines en la especificación de una fuente. El comodín en este caso tiene la interpretación habitual, es decir, cualquier cadena de caracteres. Lo que ocurre entonces es que, de todas las fuentes que encajan en esa definición, el sistema asigna la primera que encuentra.

Tenemos a nuestra disposición varias utilidades para ver las fuentes de nuestro sistema, una de ellas es

- `xfonstsel` : el clásico, es una aplicación que permite ver cómo queda una fuente al modificar dinámicamente las distintas partes del nombre. Usando esta utilidad es como mejor se puede entender la forma en que se nombran las fuentes en Linux

### Servidor de fuentes: xfs

Red Hat usa `xfs` como programa servidor de fuentes del sistema X.

Ya vimos que en los ficheros de configuración del entorno gráfico aparecía una línea del tipo:

```
Section "Files"
    FontPath "unix/:7100"
EndSection
```

con esta entrada informamos al servidor gráfico que se conecte al puerto 7100 (en ese puerto está escuchando `xfs`) para cualquier tema relacionado con fuentes del sistema.

La configuración del demonio `xfs` se realiza mediante el fichero `/etc/X11/fs/config`:

```
#
# Default font server configuration file for Red Hat Linux
#

# número máximo de clientes conectados en este servidor
client-limit = 10

# Si es on, cuando lleguemos a 10 conexiones, se clonará otro proceso xfs
clone-self = on

# servidores alternativos (separados por comas) que podrán ser utilizados en
el caso de que el servidor actual no esté disponible
#alternate-servers = foo:7101,bar:7102

# fuentes a utilizar. Si añadimos :unscaled conseguimos que las fuentes no escalables
se carguen antes que el resto
#
catalogue = /usr/X11R6/lib/X11/fonts/misc:unscaled,
            /usr/X11R6/lib/X11/fonts/75dpi:unscaled,
            /usr/X11R6/lib/X11/fonts/100dpi:unscaled,
            /usr/X11R6/lib/X11/fonts/misc,
```



```
/usr/X11R6/lib/X11/fonts/Type1,  
/usr/X11R6/lib/X11/fonts/Speedo,  
/usr/X11R6/lib/X11/fonts/cyrillic,  
/usr/X11R6/lib/X11/fonts/CID,  
/usr/X11R6/lib/X11/fonts/local,  
/usr/X11R6/lib/X11/fonts/latin2/Type1,  
/usr/share/fonts/default/TrueType,  
/usr/share/fonts/default/Type1,  
/usr/share/AbiSuite/fonts,  
/usr/share/fonts/ja/TrueType  
  
# tamaño por defecto de 12 puntos  
default-point-size = 120  
  
# resoluciones soportadas por el servidor: 100 x 100 y 75 x 75  
default-resolutions = 75,75,100,100  
  
# carga glyphs en fuentes de 16 bit (idiomas asiáticos)  
deferglyphs = 16  
  
# en on usará el registro de logs del sistema  
use-syslog = on  
  
# protocolo prohibido de escuchar  
no-listen = tcp  
Para trabajar con fuentes disponemos del comando chkfontpath veamos algunos ejemplos sobre su  
uso:
```

#### ➡ Para practicar:

- Listar los directorios (o rutas) de las fuentes instaladas:  

```
$ /usr/sbin/chkfontpath
```
- Eliminemos una entrada<sup>81</sup>:  

```
#!/usr/sbin/chkfontpath -r /usr/X11R6/lib/X11/fonts/cyrillic
```
- Si queremos añadir una entrada<sup>82</sup>:  

```
#!/usr/sbin/chkfontpath -a /usr/X11R6/lib/X11/fonts/cyrillic
```
- Para que los cambios sean efectivos:  

```
#!/sbin/service xfs reload
```

<sup>81</sup>Si bien aparece en el listado anterior, no debería estar instalada en el sistema.

<sup>82</sup>Si el directorio no existe, no lo permite. Éste debería ser el caso en el ejemplo

## Capítulo 8

# Moviéndose por Linux: GNOME (y KDE)

A mí, el amable responsable de las PUF, me gusta definir GNOME como todo aquello que se espera disponer en un entorno moderno de programación. En este aspecto, es aproximadamente equivalente a CDE, win32, OpenStep, o KDE. La gran diferencia es que, a diferencia de los ejemplos mencionados anteriormente, cada componente individual de GNOME es software "libre" o de "código abierto". (*GNOME, preguntas de uso frecuente, PUF (FAQ)*)<sup>1</sup> TODD GRAHAM LEWIS Y DAVID "GLEEF" ZOLL)

En Linux disponemos de varios entornos de escritorio, pero sin duda los dos más "famosos" (y mejores) son:

- GNOME (GNU Network Object Model Environment): la primera versión estable es de Marzo del 99.
- KDE<sup>2</sup> (K Desktop Environment): la primera versión estable es de julio del 98.

Como ya hemos comentado, los entornos de escritorio aparecieron para intentar hacer más fácil la incorporación de usuarios inexpertos al mundo GNU/Linux. Los dos escritorios que vienen con Red Hat 8.0 presentan una serie de características comunes:

- \* Disponen de un panel (barra de herramientas) altamente configurable.
- \* Programas gráficos que permiten gestionar el sistema de ficheros (Nautilus y Konqueror).
- \* Programas de utilidad que facilitan la labor diaria: programas de ofimática, programas que facilitan la conexión a Internet, gestores de correo, navegadores de internet, etc.
- \* Disponen de librerías que permiten la comunicación entre aplicaciones ("drag and drop")

Para ampliar sobre ambos escritorios:

- Web Oficial de GNOME <http://www.gnome.org/>
  - La Web en castellano de GNOME <http://www.es.gnome.org><sup>3</sup>
- La Web del KDE <http://www.kde.org>

En la guía: *Official Red Hat Linux Getting Started Guide*, tenéis casi todo lo necesario para manejar correctamente GNOME y el KDE.

---

<sup>1</sup><http://www.linuxlots.com/~barreiro/spanish/gnome-es/faq/>

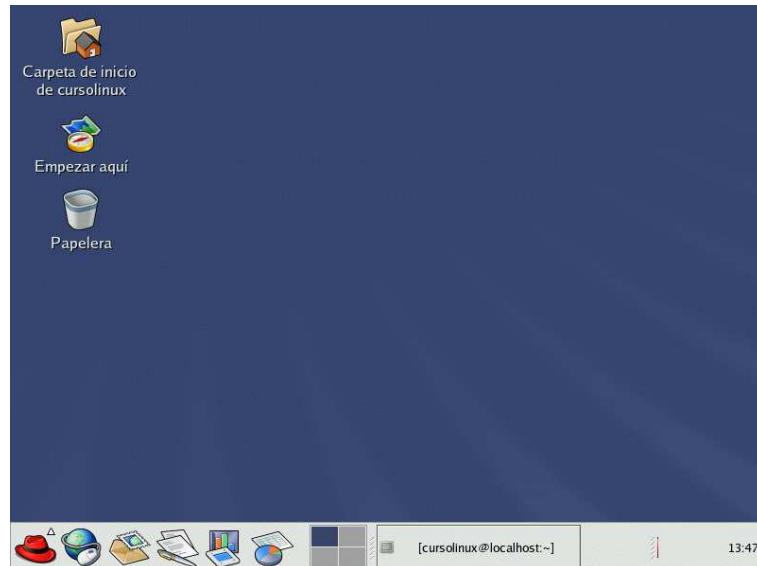
<sup>2</sup>Si se ha instalado el sistema tal cual aparece en la primera entrega no se dispondrá de este escritorio ni de sus aplicaciones. En entregas posteriores ejemplificaremos cómo poder disponer de él sin tener que instalar de nuevo todo el sistema.

En cualquier caso y aún sabiendo que no "debería" estar instalado, hemos preferido comentarlo ya que su uso es similar el de GNOME y no presenta mayor dificultad.


<sup>3</sup>Desde esta Web se puede enlazar a <http://www.linuxlots.com/~barreiro/spanish/gnome-es/usua/>, en ella tenéis la *Guía de usuario de GNOME* de DAVID C. MASON, DAVID A. WHEELER y traducidas por MANUEL DE VEGA Y EQUIPO LINUX LANDIA.

## 8.1. GNOME

### 8.1.1. El escritorio



El escritorio de GNOME es el lugar donde se encuentran los enlaces a aplicaciones, las ventanas de documentos, listas de ficheros (carpetas), etc. En la parte inferior del escritorio tenemos el panel de GNOME.



Para obtener ayuda del entorno:  → **Ayuda.**

El uso de las ventanas (maximizar, minimizar, cerrar, etc) es similar a cuando trabajamos con otros sistemas operativos (de cuyo nombre no quiero acordarme). Comencemos a trabajar un poco con GNOME:

### 8.1.2. Nautilus

Ya veremos en sesiones posteriores que acceder a disqueteras o cdrom no es inmediato en Linux. Pero, desde GNOME sí que lo parece. Si introducimos en la unidad un disco compacto, se abrirá automáticamente el administrador de ficheros de esta versión de GNOME: *Nautilus*. Se trata de un gestor de ficheros que además es un navegador Web (y más cosas). Con él podemos movernos por el árbol de directorios de nuestra máquina y realizar todas las operaciones usuales en el manejo de ficheros<sup>4</sup>.

Podemos acceder a él de varias formas:

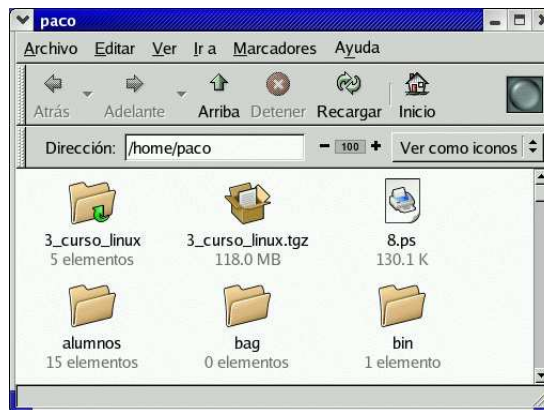
- Porque se abre solo -:-)
- Pulsando sobre cualquiera de los tres iconos que aparecen en el escritorio (**Carpeta de inicio**, **Empezar aquí** o **Papelera**)
- Siguiendo la cadena de menús  → **Lanzar programa** y tras escribir **Nautilus** pulsar sobre el botón **Lanzar**.
- Si desde una *xterm* (  → **Herramientas del sistema** → **Terminal**) ejecutamos: `nautilus`

➡ **Para practicar:** Editar usando Nautilus el fichero `~/ .bash_profile`

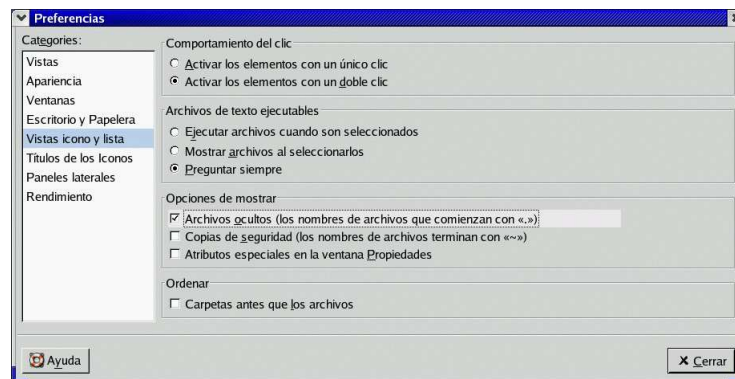
1. Si no tenemos Nautilus abierto pulsaremos sobre el icono  **Directorio Personal**

---

<sup>4</sup>Siempre que tengamos permiso para ello



2. Como es un archivo que comienza por punto (“archivo oculto”), le tendremos que decir al gestor de ventanas que nos lo muestre, para eso pulsaremos en **Editar** → **Preferencias** y optaremos por



Cuando accedamos de nuevo a nuestro directorio de usuario, veremos el fichero `.bash_profile` (ya veremos después para qué sirve) y le vamos a modificar la línea

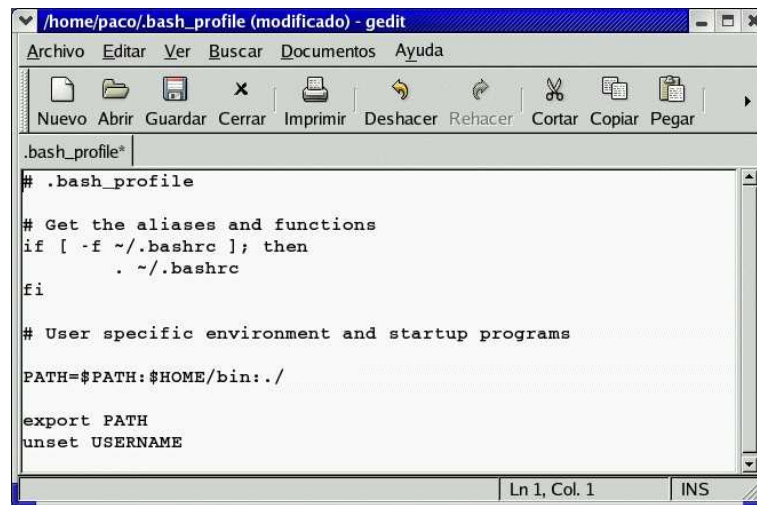
```
PATH=$PATH:$HOME/bin5
```

Para eso editemos el fichero. Sólo tenemos que pulsar sobre él con el botón derecho<sup>6</sup> del ratón y seleccionar **Abrir con gedit**, después añadamos a la línea anterior `.`/<sup>7</sup>

<sup>5</sup>path por defecto de ese usuario

<sup>6</sup>Si pulsamos sobre él dos veces podremos visualizar el fichero pero no modificarlo.

<sup>7</sup>Con esto conseguimos que el directorio en uso forme parte del path por defecto y si deseamos ejecutar un programa que está en él, no tenemos que poner la ruta completa.




### 8.1.3. El panel

Uno de los aspectos en los que destaca GNOME es en que casi todo se puede adecuar a nuestro antojo. Y uno de los aspectos más configurables es el Panel.



Elementos del Panel:

#### Menú principal de GNOME: Algunos programas de GNOME

 Tan solo vamos a enumerar algunas de las utilidades disponibles en GNOME. Están organizadas de igual manera a como salen en los menús de GNOME. Además, entre paréntesis tenéis el nombre del comando por si queréis ejecutarlos desde un xterm:

#### Accesorios

- Calculadora (gnome-calculator)
- Diccionario (gnome-dictionary)
- Editor de textos (gedit)
- File Roller (file-roller): Para manejar archivos comprimidos
- Mapa de caracteres (gnome-character-map): Inserta caracteres especiales en los documentos
- Piloto/Herramienta Handspring (gpilotd-control-applet): Configurar la conexión para el PDA del usuario

#### Configuración del sistema

- Autenticación (usr/bin/authconfig-gtk): Controlar el modo en el que el sistema verifica a los usuarios que intentan conectarse
- Contraseña de root (/usr/bin/redhat-config-rootpassword): Configura la contraseña de root para el sistema

- Detección de la tarjeta de sonido (/usr/bin/redhat-config-soundcard): Autodetección y configuración de la tarjeta de sonido
- Fecha y hora (/usr/bin/redhat-config-date): Cambiar la fecha y la hora del sistema
- Gestor de usuarios (/usr/bin/redhat-config-users): Para añadir usuarios
- Idioma (/usr/bin/redhat-config-language): Cambia el idioma/lenguaje por defecto
- IMail Transport Agent Switcher (/usr/bin/redhat-switchmail): Cambia el agente de transporte de correo electrónico
- Pantalla de registro (gdmsetup): Configurar la pantalla de registro
- Paquetes (/usr/bin/redhat-config-packages): Gestionar los paquetes instalados en el sistema
- Printing (/usr/bin/printconf-gui) : Configuración de impresoras
- Ratón (/usr/bin/redhat-config-mouse): Configurar el ratón
- Red (/usr/bin/neat): Configuración de dispositivos de red y conexiones
- Security Level (/usr/bin/redhat-config-securitylevel): Configurar el nivel de seguridad del sistema
- Teclado (/usr/bin/redhat-config-keyboard): Herramienta de configuración del teclado
- Visualización (/usr/bin/redhat-config-xfree86): Configurar la resolución y los colores

#### Configuración de servidores

- Servicios (/usr/bin/redhat-config-services): Configura los servicios que se iniciarán cuando el sistema arranca.

#### Gráficos

- Digital Camera Tool (/usr/bin/gtkam)
- EL GIMP (gimp): Programa de manipulación de imágenes GNU
- GQview (gqview): Visualizador rápido de imágenes
- PDF Viewer (xpdf): Visor de archivos en formato pdf
- Scanning (xsane): Utilidad para adquirir imágenes con el Scanner

#### Herramientas del sistema

- Configuración de kickstart (/usr/sbin/redhat-config-kickstart): Generador de ficheros de kickstart (arranque rápido)
- Control del dispositivo de red (/usr/bin/neat-control): Controlar y monitorizar los dispositivos de red
- Formateador de disquetes (gfloppy)
- Gestión del disco (usermount): Montar y desmontar sistemas de ficheros
- Monitor del sistema (gnome-system-monitor): Ver el proceso actual y monitorizar el estado del sistema
- Navegador de hardware (hwbrowser): Muestra información sobre el hardware del ordenador
- Red Hat Network (/usr/bin/up2date). Determina los paquetes del sistema que necesitan actualización
- Registro del sistema (/usr/bin/redhat-logviewer): Fichero de registro del sistema
- Terminal (gnome-terminal): Línea de comandos
- Traceroute (xmttr): Para conocer por qué máquinas pasamos para llegar vía red a una máquina en concreto.
- Wizard de configuración de internet (/usr/bin/internet-druid): Configura la conexión de internet

#### Internet

- Correo electrónico (evolution): Correo electrónico y calendario Evolution
- Instant Messenger (gaim)
- Navegador de Web (mozilla)

## Juegos

## Ofimática

- Configuración de la impresora<sup>8</sup> OpenOffice.org (oopadmin)
- Diagramas (dia): Crear diagramas
- Dibujos de OpenOffice.org (oodraw): Dibujar diagramas y figuras
- Escritor de OpenOffice.org (oowriter): Procesador de textos
- Impresión de OpenOffice.org (ooimpress): Para crear presentaciones
- OpenOffice.org Calc (oocalc): Hoja de cálculo
- OpenOffice.org Math (oomath): Editor de fórmulas
- Project Management (mrproject)

## Preferencias

- Centro de control: Preferencias personales y configuraciones
- Accesibilidad (gnome-accessibility-keyboard-properties)
- Combinación de teclas (gnome-keybinding-properties): Asigna combinaciones de teclas a los comandos
- Contraseña (userpasswd): Para cambiar la contraseña de registro
- Datos personales (userinfo): Cambio de la información personal
- Foco de la ventana (metacity-properties): Cambia el modo en que el foco se mueve desde una ventana a otra
- Foto de inicio de sesión (gdmphotosetup)
- Menus & Toolbars (gnome-ui-properties): Para personalizar la apariencia de las barras de herramientas y de menú
- Network Proxy (gnome-network-preferences)
- Pilot/Herramientas Handspring (gpilotd-control-applet): Para la PDA Pilot
- Propiedades del CD (gnome-cd-properties). Configura el manejo de dispositivos CD
- Propiedades salvapantallas (xscreensaver-demo-crapplet): Cambiar las propiedades del salvapantallas
- Ratón (gnome-mouse-properties)
- Sonido (gnome-sound-properties): Activar el sonido y asociar sonidos con eventos
- Tapiz (gnome-background-properties): Personalizar el fondo del escritorio
- Teclado (gnome-keyboard-properties)
- Tema (gnome-theme-properties): Seleccionar temas para las distintas partes del escritorio
- Tipografía (gnome-font-properties): Seleccionar las fuentes para el escritorio
- Tipos de archivo y programas (gnome-file-types-properties): Asociar aplicaciones con tipos de archivos

## Programar

- Emacs (emacs)

## Sonido y video

---

<sup>8</sup>Con esta utilidad podremos generar pdfs directamente desde el OpenOffice.



- Control de volumen (`gnome-volume-control`)
- Grabador de sonido (`gnome-sound-recorder`): Grabar clips de sonido
- Lector de CD (`gnome-cd`): Reproducir CDs audio
- Reproductor de audio (`xmms`): Reproduce Ogg Vorbis y otros ficheros audio

## Extras

### Configuración del sistema

- Desktop Switching Tool (`switchdesk`)
- Mail Transport Agent Switcher (`/usr/bin/redhat-switchmail`)
- Printer System Switcher (`/usr/bin/redhat-switch-printer`)

### Gráficos

- Ghostview (`gv`)
- ImageMagic (`/usr/X11R6/bin/display`): Visor de ImageMagic y editor
- PostScript Viewer (`ggv`): Para ver archivos postscript
- Visor de imágenes EOG (`eog`): Visualizar distintos tipos de imágenes

### Herramientas del sistema

- CDWriter (`/usr/bin/gtoaster`): Para “tostar” CDs
- Editor de configuración (`gconf-editor`): Edita directamente la base de datos de configuración
- Kernel Tuning (`redhat-config-proc`)

### Internet

- X-Chat (`xchat`)
- gFTP (`gftp`): Cliente FTP multihilo
- Mozilla Mail (`/usr/bin/mozilla-mail`): Leer el correo con Mozilla

### Juegos

#### Ofimática

- Modificación OpenOffice.org (`/usr/lib/openoffice/program/setup.bin`): Modificar la instalación de OpenOffice.org

#### Preferencias

- Aplicaciones predeterminadas (`gnome-default-applications-properties`): Para seleccionar las aplicaciones por defecto
- CDDb (`cddb-slave2-properties`): Propiedades CDDb (CD Database)
- Desktop Switching Tool (`switchdesk`): Para alternar varios entornos de escritorio o gestores de ventanas
- Panel (`gnome-panel-preferences`): Selecciona las preferencias para los paneles.
- Sessions (`gnome-session-properties`): Configurar las sesiones

#### Programar

- Herramienta de informe de errores (`bug-buddy`): Informe de un error en GNOME
- Memory Profiler (`menprof`): Chequea un programa para analizar su uso de la memoria

#### Sonido y video

- grip (`/usr/bin/grip`)

**Ayuda** (`yelp`): Todo (bueno, bastantes cosas, pero todo, todo, no) lo que quiso saber sobre GNOME y no se atrevió a preguntar.



**Carpeta de inicio** (nautilus): Ver la carpeta de inicio en el administrador de archivos Nautilus







**Lanzar programa:** Ejecutar un comando

**Buscar archivos** (gnome-search-tool): Busca archivos, carpetas y documentos en el ordenador

**Bloquear la pantalla** (xscreensaver): permite bloquear la pantalla hasta que introduzcamos nuestra contraseña de usuario

**Terminar sesión:** Salir de la sesión para entrar como un usuario diferente o apagar el ordenador

### Resto de elementos del Panel

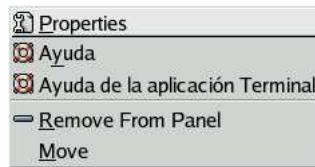
-  Navegador web Mozilla
-  Gestor de correo y calendario Ximian Evolución.
-  OpenOffice writer
-  OpenOffice Impress
-  OpenOffice Calc
-  Paginador de espacios de trabajo y ... reloj.

### ➡ Para practicar

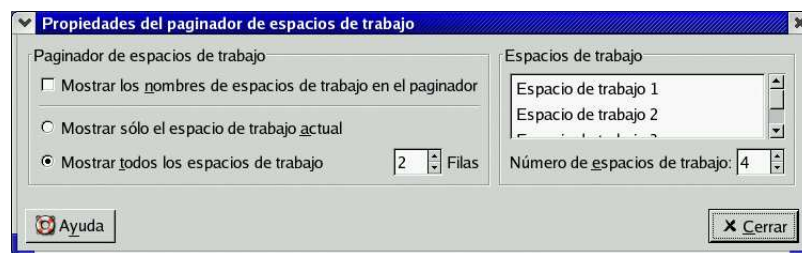
1. Añadir un lanzador al panel para abrir un terminal: pulsemos con el botón derecho del ratón en una zona libre del panel de Gnome y



seleccionemos en el menú emergente: **Lanzador desde Menú** → **Herramientas del sistema** → **Terminal**. Se añadirá un icono que nos va a permitir abrir un terminal de introducción de órdenes. Podemos modificar esta lanzadera pulsando sobre ella con el botón derecho del ratón.

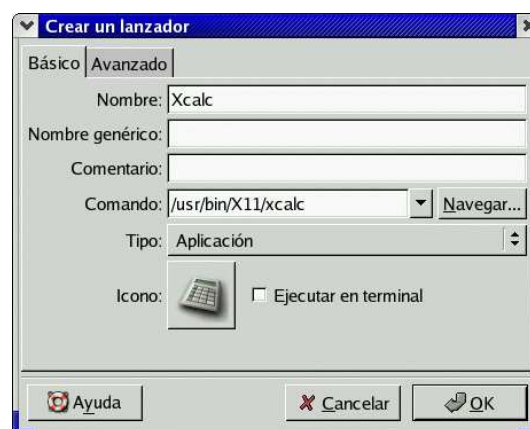


2. Cambiar la fuente por defecto de un terminal de texto: Cuando usamos resoluciones elevadas, el texto de los terminales puede que se vea muy pequeño. Vamos a poner una fuente de 20 puntos como fuente por defecto. Para eso abramos una `xterm` y en el menú principal pulsemos en **Editar**→**Perfil Actual**, después deshabilitaremos la opción de **Usar la misma fuente que otras aplicaciones** y seleccionaremos la fuente que se ajuste bien a la resolución de pantalla con la que estamos trabajando.
3. Escritorios múltiples: GNOME viene configurado por defecto para trabajar con 4 escritorios virtuales.
  - a) Abrir los programas: Mozilla, Nautilus, un terminal y Gedit y ponerlos en los 4 escritorios virtuales (en ese orden).<sup>9</sup>
  - b) Vamos a conseguir que sean 8 los escritorios. Pulsemos con el botón derecho del ratón sobre la zona de escritorios múltiples. Optemos por **Preferencias** en el menú que aparece y nos debe salir




Una vez aquí hacer que sean 8 es fácil ¿no?.

4. Comprobar el efecto de la combinación de teclas **[Alt]+[F1]**
5. Vamos a añadir al panel un icono que nos va a permitir ejecutar una calculadora. Pulsemos sobre el panel con el botón derecho del ratón y después, en el menú emergente: **Panel**→**Añadir al Panel**→**Lanzador**



<sup>9</sup>Podemos ver qué aplicaciones tenemos abiertas con la combinación de teclas **[Alt] + [Tab]**

Para obtener el icono (en principio está vacío) sólo hay que pulsar en el botón de Icono y podremos seleccionarlo del directorio<sup>10</sup> `/usr/share/pixmaps`.


6. Añadir un applet que nos marque cuándo tenemos correo-e: **Add to Panel**→**Red**→ **Monitor de buzón entrada**.
7. Apagado y reinicio del sistema: Una de las cosas que podemos hacer desde el entorno gráfico es reiniciar (reboot) o terminar con la sesión de trabajo (halt). Al pulsar sobre el icono , aparecerá el menú que nos permite terminar la sesión:



podremos salir del entorno gráfico (Logout), reiniciar el equipo (Restart the computer o Reboot) o salir de Linux apagando la máquina (Shut Down o Halt).

Podemos añadir un botón de salida al panel mediante la opción Log Out Button

#### 8.1.4. Personalizar GNOME

Desde  →**Preferencias** o con los submenús que aparecen tras →**Extras**→**Preferencias** podemos configurar casi todo lo que se nos ocurra para adecuar GNOME a nuestro gusto:

- El fondo.
- Cambiar el aspecto del escritorio usando diferentes temas.
- Etc.

Se trata de probar.

---

<sup>10</sup>Se puede optar por otros directorios.

## Capítulo 9

# Configuración: Impresora, Sonido, etc.

El software, sin el hardware, no es más que palabras en una página. Sin embargo, no sólo es eso. Me gusta la idea de que todo comenzó como piedras y arena y ahora podemos enviar hombres a la luna y ver dentro de los átomos. (*LINUX. Recursos para el usuario*. JAMES MOHR)

### 9.1. Configuración personalizada: impresora, tarjeta de sonido.

En esta sección pretendemos ahondar un poco sobre algunos de los aspectos más comunes de configuración de nuestros equipos. Para cuestiones puntuales o hardware específico, no queda más remedio que buscar en lugares donde podamos encontrar información para un modelo en concreto.

Dos páginas imprescindibles:

- Web de los grupos de noticias es.comp.os.linux.\* <http://www.escomposlinux.org>
- Hardware soportado por la RedHat <http://hardware.redhat.com>

Los periféricos que vamos a tratar van a ser los dos que hoy por hoy, son más comunes en nuestros equipos:

- Impresoras
- Tarjetas de sonido

Se supondrá en todas las secciones en que tratemos temas relativos a configuración de nuestra máquina que estamos trabajando como root.

La herramienta de autodetección del hardware de nuestra máquina se denomina kudzu y, salvo que lo deshabilitemos, se ejecuta siempre en el arranque. Para los periféricos más usuales, lo normal es que los autodetecte y configure en el arranque.<sup>1</sup>

### 9.2. Impresoras

De igual manera que con los módem internos, en el mercado hay impresoras que funcionan perfectamente con Linux, otras con las que se puede trabajar aunque den un poco la lata y otras con las que no hay manera<sup>2</sup>. Para comprobar dónde se encuadra la nuestra antes de ponernos manos a la obra deberíamos consultar en:

- <http://www.linuxprinting.org/howto>

---

<sup>1</sup>Si deseamos iniciar este programa ya desde Linux podemos optar por escribir:

```
#/sbin/service kudzu start
```

<sup>2</sup>En general cualquier impresora que funcione bajo MSDOS funcionará en Linux. Si nuestra impresora sólo imprime bajo Windows: ¡mal asunto!.

- Hardware-HOWTO
- RedHat Hardware Compatibility List

Un documento muy completo y amplio que abarca aspectos más técnicos sobre la impresión es el *HOWTO Configuración-Impresión-Como*. En este documento, además de tratar la configuración de las impresoras, se comenta cómo se imprime y con qué imprimir distintos formatos de ficheros.

Antes de ver la parte “fácil” de la configuración gráfica, demos un repaso a los elementos de los que consta el sistema de impresión en Linux.

- El demonio<sup>3</sup> de impresión `lpd`, que se inicia al arrancar el sistema y es el encargado de recibir los trabajos de impresión.
- El comando `lpr` que sirve para mandar trabajos de impresión al demonio `lpd`
- Filtros, que convierten los documentos que vamos a imprimir a un determinado lenguaje que sea capaz de entender la impresora.
- El fichero `/etc/printcap` contiene la configuración de las impresoras que se definan en el sistema.
- El directorio `/var/spool/lpd`, que es donde se almacenan temporalmente los trabajos de impresión hasta que les llegue su turno de ser impresos.

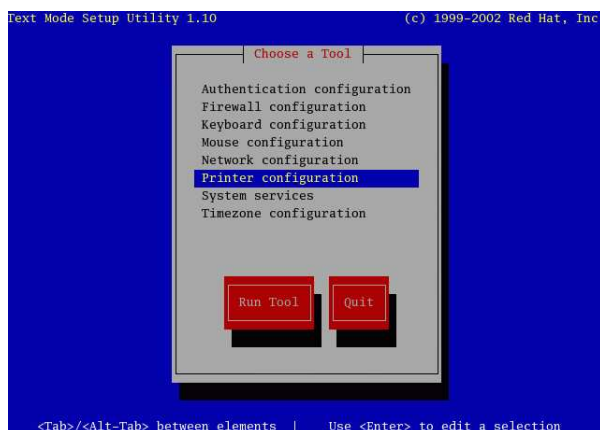
Siguiendo un orden cronológico, para imprimir un trabajo lo haremos con `lpr`<sup>4</sup>, que lo manda a un directorio de spool según la configuración que está en `/etc/printcap`. Del directorio de spool lo coge el demonio `lpd` que lo enviará a la impresora física correspondiente, pasándole el filtro adecuado. Si no lo puede mandar inmediatamente a la impresora, lo dejará en el directorio de spool en espera de que llegue su turno o la impresora esté preparada.

Estos conceptos sirven para situarnos y saber qué es lo que pasa por debajo, pero el configurar nuestra impresora no nos debe resultar muy complicado con la ayuda de

`#redhat-config-printer`<sup>5</sup>.

Además de la comentada en la línea superior, disponemos de varias opciones de iniciar esta utilidad de configuración.

- Ejecutar<sup>6</sup>  
`#/usr/sbin/setup`




<sup>3</sup>Se llama demonios (daemon) a los procesos que se ejecutan en segundo plano encargados de realizar operaciones importantes para el sistema. Suelen comenzar su ejecución al arrancar el sistema y permanecen a la escucha para realizar un servicio determinado.

<sup>4</sup>O los comandos o iconos gráficos que a su vez llaman a éste

<sup>5</sup>Si usamos `printconfig` conseguimos el mismo efecto. Si estamos en modo gráfico se inicia el programa de configuración en modo gráfico (`redhat-config-printer-gui`) en caso contrario se inicia `redhat-config-printer-tui`


<sup>6</sup>Notar que desde este menú de texto se accede a múltiples herramientas de configuración de nuestro Red Hat.

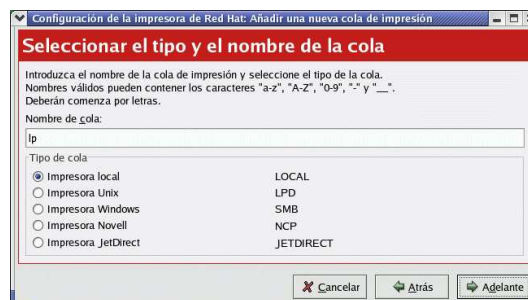
y optar por **Printer configuration**

-  → **Configuración del Sistema** → **Printing**
- **Empezar aquí** → **Configuración del Sistema** → **Printing**

En cualquier caso, si optamos por configurar la impresora usando las X window obtendremos



Para añadir impresoras usaremos el botón **Nuevo** . Después de confirmar que vamos a crear una cola de impresión pulsando sobre **Adelante**, aparecerá una ventana que nos presenta la posibilidad de elegir el tipo de impresora que vamos a instalar:



Las posibilidades que nos aparecen son:

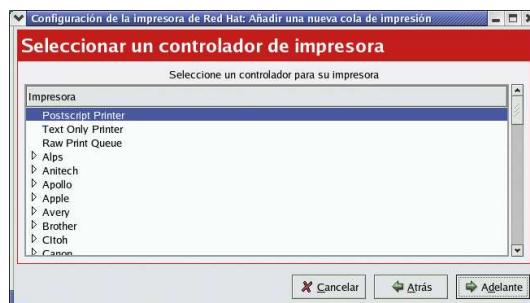
- **Nombre de cola:** le he puesto `lp` (*local printer*) pero al gusto. La cola de impresión que se llame `lp`, será la cola por defecto.
- **Impresora Local:** para una impresora conectada a un puerto paralelo (o serie) del ordenador.
- **Impresora Unix:** impresoras conectadas a otra máquina Linux/Unix a la que estamos conectados en Red. En la máquina remota debe estar funcionando el demonio `lpd`.
- **Impresora Windows:** para usar los servicios de impresión de otra máquina con Windows que hace de servidor de impresión. Para ello hace uso de SAMBA que es un conjunto de utilidades que integran a Linux en una red de Windows.
- **Impresora Novell:** Para usar una impresora de una red Novell.
- **Impresora JetDirect:** Para impresoras que tienen su propia conexión a red<sup>7</sup>.

<sup>7</sup>Dentro de las impresoras con tarjeta de red integrada, nos encontramos con la posibilidad de conexión directa a puerto (que es a la que nos referimos aquí) y la posibilidad de tener un demonio `lpd` ejecutándose en la impresora (en este caso, sería como la segunda opción Impresora Unix).

Si aceptamos en la opción por defecto (impresora local<sup>8</sup>) el programa deberá autodetectarnos el puerto paralelo de la impresora<sup>9</sup>. Si todo se ha realizado de forma correcta veremos una ventana similar a:



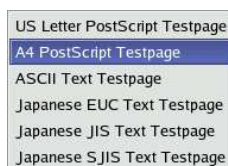
Si pulsamos sobre **Adelante** comienza el proceso de configuración de nuestra impresora. Mediante esta ventana seleccionaremos el filtro que mejor se adecúe a ella y los parámetros que mejor se ajusten a nuestras necesidades.



Seleccionaremos el filtro adecuado para nuestro modelo y tendremos:



Una vez elegido el filtro adecuado, y sin cerrar la ventana gráfica inicial del programa, probaremos nuestra impresora usando el menú **Probar** y seleccionando **Imprimir una página de prueba Postscript A4**:

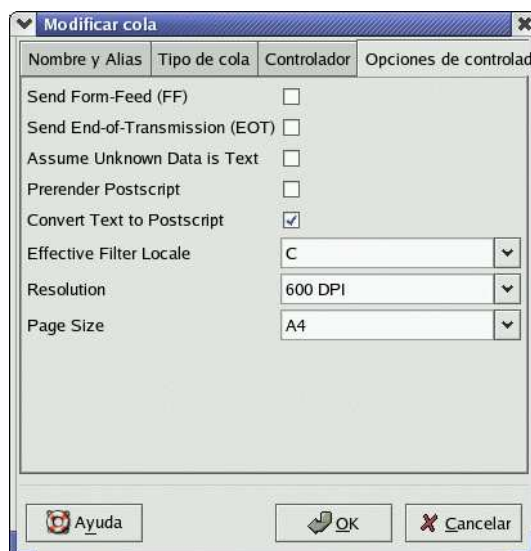


<sup>8</sup>Para el resto de posibilidades mirar el capítulo 7 de *The Official Red Hat Linux Getting Started Guide* y sobre todo el 26 de *The Official Red Hat Linux Customization Guide* en <http://europe.redhat.com/documentation/>

<sup>9</sup>Nuestro puerto paralelo que se llamaba LPT1 en DOS, en Linux se denomina /dev/lp0. Si se trata de LPT2 en linux será /dev/lp1, ...



Antes de mandar la página a la impresora nos pedirá confirmación para guardar los cambios y reiniciar el demonio de impresión. Si no hemos elegido el modelo exacto, lo más probable es que tengamos que modificar los parámetros de la impresora, eso se consigue usando el botón **Modificar**. La pantalla más interesante se corresponde con:



- **Enviar FF y/o Enviar EOT**: si marcamos estas casillas, obligamos a que la impresora tire la página al finalizar el trabajo de impresión. Pueden dar lugar a que aparezcan páginas en blanco; en este caso deberíamos desactivarlas. Sin embargo, es necesario marcarlo si nuestra impresora se queda en el buffer la última página y esta opción le obliga a “escupirla”. Para probarlas marcaríamos en pimer lugar sólo la opción **Enviar FF**, si no funciona, sólo la segunda (**Enviar EOT**) y si tampoco, las dos.
- Si marcamos **Asume que los datos desconocidos son texto** el controlador de impresión imprimirá como texto cualquier dato que no pueda reconocer. Sólo deberíamos marcarla si tenemos problemas con algunos caracteres que no se imprimen bien.
- Optaremos por la anterior y **Convertir texto a Postscript** si tenemos problemas al imprimir algunos datos que no reconoce la impresora, esos datos se asumirá que son texto y después, el texto se pasará a PostScript.
- Si **Rehacemos PostScript**, dispondremos de fuentes PostScript no estándar, al marcarla se pueden solucionar problemas con determinadas fuentes “extrañas”. El proceso de impresión será más lento.
- Si no imprimimos en Japonés, en **Localización de filtro efectivo** dejaremos C
- El resto de campos dependen del modelo, en mi caso cambié:
  - *Resolution* que la puse a 600 dpi, y
  - Paper Size, el tamaño de papel, que puse en A4<sup>10</sup>.

Si pulsamos sobre **OK** volveremos a la ventana inicial y tendremos que guardar los cambios realizados y, si deseamos mandar una página de prueba, reiniciar el demonio de impresión.

Merece la pena comentar que para una misma impresora podríamos tener definidas varias configuraciones. Esto se consigue añadiendo la misma impresora con otro nombre y con un filtro diferente. Así, por ejemplo si disponemos de una impresora de inyección color podemos optar porque la primera esté definida para imprimir en color y la segunda en blanco y negro.

<sup>10</sup>Por defecto aparecerá el formato letter

### 9.2.1. /etc/printcap

En él se configuran las impresoras de nuestro sistema. No debemos editarlo a mano ya que los cambios que realicemos en él no se mantendrán, esto se debe a que el demonio de impresión crea un fichero /etc/printcap cada vez que se reinicia. Si queremos añadir alguna impresora a mano debemos usar el fichero /etc/printcap.local.

Tras salir de la utilidad `redhat-config-printer` y reiniciar el demonio `lpd`, el fichero que hemos creado es el fichero `printcap` del directorio /etc; para una impresora HP5L (con filtro *lj4*) es<sup>11</sup>:

```
# /etc/printcap
#
# DO NOT EDIT! MANUAL CHANGES WILL BE LOST!
# This file is autogenerated by printconf-backend during lpd init.
#
# Hand edited changes can be put in /etc/printcap.local, and will be included.

lp:\
    :ml=0:\
    :mx=0:\
    :sd=/var/spool/lpd/lp:\
    :af=/var/spool/lpd/lp/lp.acct:\
    :sh:\
    :lp=/dev/lp0:\
    :lpd_bounce=true:\
    :if=/usr/share/printconf/util/mf_wrapper:

#####
## Everything below here is included verbatim from /etc/printcap.local      ##
#####
# printcap.local
#
# This file is included by printconf's generated printcap,
# and can be used to specify custom hand edited printers.
```

El fichero /etc/printcap<sup>12</sup> es donde tenemos almacenada la información relativa a cada una de las impresoras conectadas al sistema (tanto locales como remotas). Por cada impresora conectada al sistema tendría que aparecer un grupo de líneas como el que aparece arriba. La primera línea es el nombre de la cola de impresión. El resto de líneas que aparecen para cada impresora son una serie de campos que siempre se inician y terminan con “:” (salvo el que contiene el nombre de la impresora, que solamente termina con los dos puntos), después aparecen nombres de campos (sd, mx, sh, etc) que pueden tomar distintos valores. Todos los campos se identifican usando dos letras y hay más de 30 campos diferentes que asignándoles distintos valores (tipo cadena, booleano<sup>13</sup> o numérico) nos permiten controlar nuestra impresora. De esos campos los que aparecen con más asiduidad son los siguientes<sup>14</sup>:

---

<sup>11</sup>El carácter “\” que aparece al final de cada línea del fichero es el carácter de continuación. Permite poner varias líneas “físicas”, pero su efecto será como si sólo fuera una línea “lógica”.

<sup>12</sup>Para ampliar el tema véase la ayuda de `printcap`:

\$ `man printcap`

<sup>13</sup>Los valores posibles son: true, false. Si no se pone nada es false.

<sup>14</sup>Si nuestra impresora es local no nos aparecerán ni rm, ni rp.

Campo	Tipo	Descripción	Ejemplo
lp	Cadena	Dispositivo de impresión	/dev/lp1
sd	Cadena	Nombre del directorio de spool de esta impresora	/var/spool/lpd/lp
lf	Cadena	Fichero donde almacenar el registro de errores	Ha de ser un fichero existente
if	Cadena	Nombre del filtro de entrada	/var/spool/lpd/lp/filter
rm	Cadena	Nombre del host de la impresora remota	192.168.1.1
rp	Cadena	Nombre de la impresora remota	lp
sh	Booleano	Suprime las páginas de cabecera para separar los trabajos	
sf	Booleano	Suprime las páginas en blanco al final de los trabajos	
mx	N Numérico	Tamaño máximo del trabajo de impresión.	0
af	Cadena	Nombre del archivo de cuenta o servidor	/var/spool/lpd/lp/lp.acct
lpd_bounce	Booleano	Fuerza a lpd a filtrar los trabajos	true
ml	N Numérico	Caracteres mínimos imprimibles para chequeo de impresión	0

Del fichero printcap de ejemplo y de la tabla anterior se puede deducir que existen tres “tipos” de campos:

1. :nombre\_campo=cadena:
2. :nombre\_campo#numero:
3. Booleanos que toman el valor *true* (verdadero) cuando aparecen y *false* como valor por defecto en el caso de que no aparezcan en el fichero. Así, si nos fijamos en el fichero `printcap` de ejemplo, como aparece el campo `sh` esto indicaría que toma el valor verdadero, mientras que como `sf` no está, toma el valor falso.

### 9.2.2. Órdenes básicas de impresión.

Para ampliar y conocer todas las órdenes lo mejor es mirar en las man de cada una de ellas. Aquí sólo vamos a poner ejemplos de su uso más frecuente.

#### Demonio lpd:

Es el proceso utilizado por Linux para gestionar los trabajos de impresión. Cuando este proceso no se está ejecutando, no se puede imprimir (los archivos permanecen en los directorios de gestión de colas hasta que lo iniciemos). Normalmente se inicia al arrancar el sistema. Si quisiéramos arrancarlo manualmente ejecutaríamos

```
#/sbin/service lpd start
para pararlo
#/sbin/service lpd stop
para saber su estado
$/sbin/service lpd status
```

#### Orden lpr

Envía un trabajo a la impresora o pone en cola un trabajo de impresión<sup>15</sup>. Si no hemos especificado ningún archivo, `lpr` lee el contenido de la entrada estándar. Su sintaxis es:

```
$ lpr [opciones] [archivo]
```

Por ejemplo, para imprimir el fichero de texto `curlinux.txt` escribiríamos:

```
$ lpr curlinux.txt
```

En el caso de que tengamos dos impresoras en el sistema, la primera “`hpcolor`” y la segunda “`hpmono`” podemos seleccionar la impresora a usar para imprimir mediante:

```
$ lpr -P impresora fichero_a_imprimir
```

<sup>15</sup>En el caso de que `lpd` no esté arrancado.

así, si deseamos imprimir un fichero en color escribiríamos:

```
$ lpr -P hpcolor cursolinux.txt
```

mientras que si deseamos imprimirlo en blanco y negro escribiríamos:

```
lpr -P hpmono cursolinux.txt
```

Si no decimos nada, es decir si escribimos:

```
$ lpr cursolinux.txt
```

escogerá aquella que se llame lp (impresora por defecto) entre uno de sus nombres.

#### Orden lpq:

Muestra el contenido del directorio de gestión de colas. También nos indica el identificador de trabajo y el orden que ocupa el trabajo en la cola.

Por defecto, muestra la cola de impresión de la impresora lp, pero si deseamos ver el estado de la impresora hpcolor escribiríamos:

```
$ lpq -P hpcolor
```

#### Orden lprm:

Elimina un trabajo de la cola. Se puede especificar el trabajo concreto que queremos eliminar:

```
$ lprm 3
```

cancelar todos los trabajos de un usuario

```
$ lprm nombre_usuario
```

cancelar todos los trabajos de impresión (si introducimos la orden como root)

```
# lprm all
```

#### Orden lpstat:

Con esta orden se puede comprobar el estado actual del servicio de impresión.

### 9.3. Tarjeta de sonido

Para mirar si está soportada, primero deberíamos mirar en las páginas antes comentadas. La utilidad que se instala por defecto para detectar la tarjeta de sonido en modo gráfico:

```
#redhat-config-soundcard
```

aunque, por ejemplo, una SoundBlaster ISA PNP no fue detectada por ella. Esto nos ha llevado a tener que instalar la utilidad que se instalaba en versiones anteriores para configurar la tarjeta de sonido y que no acompañará versiones futuras de Red Hat, se trata del paquete del CD3 de `sndconfig-0.69-1.i386.rpm`.

Pasemos a instalarlo:

```
#mount /mnt/cdrom
```

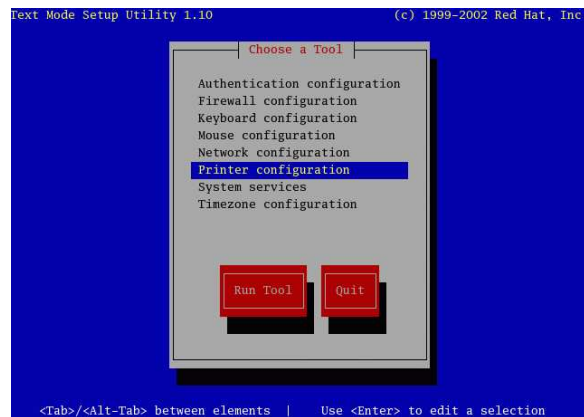
```
#rpm -i /mnt/cdrom/RedHat/RPMS/sndconfig-0.69-1.i386.rpm
```

Una vez instalado el paquete, y para configurar una tarjeta<sup>16</sup> soportada usaremos

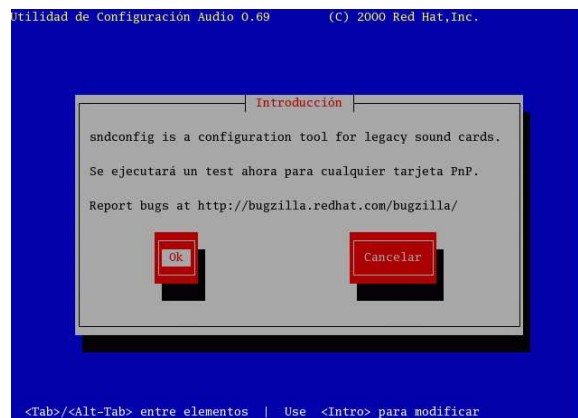
```
# /usr/sbin/setup
```

---

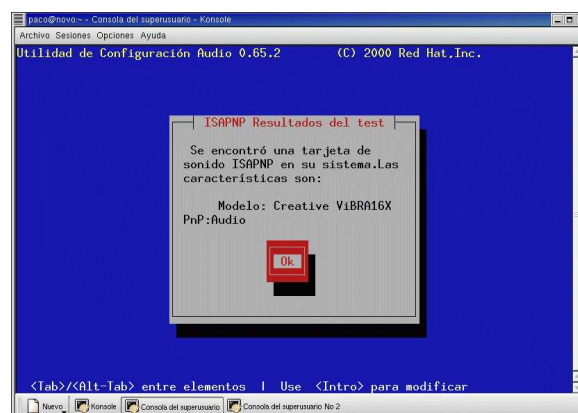
<sup>16</sup>Puede que para algún modelo de tarjeta sea necesario recompilar el kernel para que funcione.



y elijeremos **Sound card configuration**, o bien ejecutamos desde la línea de comandos  
`# /usr/sbin/sndconfig`  
 este programa intentará autodetectar nuestra tarjeta



si lo consigue, veremos una pantalla similar a la que sigue en la que el programa nos informará de la tarjeta de sonido conectada a nuestro equipo



Una vez pasado este trámite el programa `sndconfig` testeará con ejemplos que la tarjeta de sonido está bien configurada. En general, si el programa ha detectado correctamente la tarjeta no debe surgir ningún

contratiempo y tan sólo con aceptar (**OK**) en las pantallas que van apareciendo tendremos la posibilidad de oír nuestra música favorita en nuestro equipo.

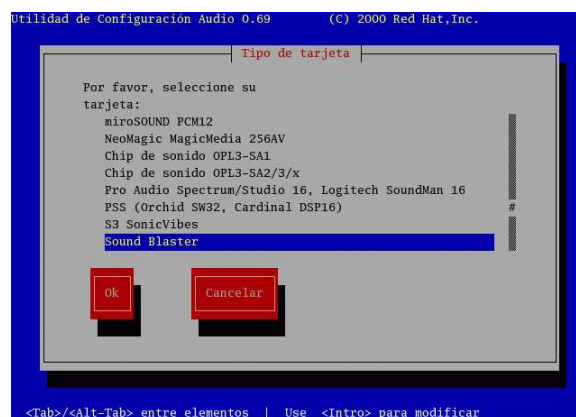
Si queremos comprobar algunos de los cambios que se han producido en el sistema podemos mirar en los ficheros que se listan a continuación. Las líneas que aparecen debajo del primer fichero se corresponden con las líneas que han aparecido tras configurar una SoundBlaster 16 pnp:

- `/etc/modules.conf`  
alias sound-slot-0 sb  
post-install sound-slot-0 /bin/aumix-minimal -f /etc/.aumixrc -L >/dev/null 2>&1 || :  
pre-remove sound-slot-0 /bin/aumix-minimal -f /etc/.aumixrc -S >/dev/null 2>&1 || :  
options sound dmabuf=1  
alias synth0 opl3  
options opl3 io=0x388  
options sb io=0x220 irq=5 dma=0 dma16=1 mpu\_io=0x330  
options sb isapnp=1
- `/etc/sysconfig/soundcard`: en este fichero aparecerá el tipo de tarjeta conectada al sistema.

Si nuestra tarjeta no es pnp ni pci, es decir, si el programa no la autodetecta y está en la lista anterior tendremos que configurarla a mano, en ese caso, la pantalla que obtendremos será:

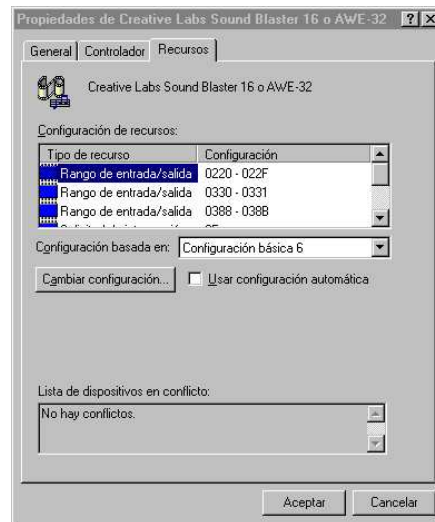


si aceptamos se nos mostrará la lista de tarjetas soportadas por el programa, donde elegiremos la nuestra.



Dependiendo de la tarjeta tendremos que configurar distintos valores, por ejemplo para una Sound-Blaster 16 “antigua” tendríamos que conocer las direcciones de memoria que usa, interrupción, etc.

Esa información tendremos que obtenerla del manual de la tarjeta<sup>17</sup> o bien, si tenemos instalado Windows podemos mirar en el Panel de Control de Windows y tras seleccionar la tarjeta de sonido mirar en la pestaña de recursos. De nuevo para esta tarjeta saldría



Es decir:

IO	0220-022F	I/O PORT
IO	0330-0331	MPU I/O
IO	0388-038B	
IRQ	5	IRQ
DMA	0	DMA 1
DMA	1	DMA 2

Con estos datos seleccionaremos correctamente en:



y listo. Tras aceptar, comprobaremos que todo funciona correctamente oyendo los sonidos de prueba que trae el programa.

<sup>17</sup>O bien, si es una tarjeta antigua, mirar cómo están situados los jumpers (o switches) de la tarjeta

Para verificar que el controlador de sonido está configurado<sup>18</sup> de manera adecuada podemos usar (aunque dependiendo del tipo de tarjeta esto puede no funcionar):

```
# cat /usr/share/sounds/error.wav >/dev/dsp
```

### 9.3.1. Más sonido

- Para conocer aplicaciones de sonido y formatos de sonido tenéis *Reproducción-De-Sonido COMO*. Este documento “contiene una lista de los diversos formatos de sonido y de las aplicaciones que pueden ser usadas para reproducirlos. Además, lista algunos trucos y consejos referentes al uso de estas aplicaciones. También describe algunas aplicaciones relacionadas con el sonido pero no directamente con su reproducción”
- Con respecto a poder ampliar el tema sobre la configuración de sonido el *Linux Sound HOWTO*<sup>19</sup> pretende ser “una guía de referencia rápida que cubre todo lo que necesitas saber para instalar y configurar el soporte de sonido bajo Linux. Se contestan las preguntas más frecuentes sobre el sonido en Linux, además de incluir referencias a otras fuentes de información sobre múltiples características relativas a la generación de sonido y música por ordenador.”
- En los apuntes sobre GNOME, en la sección “Algunos programas de GNOME” aparecen comentados los programas sobre sonido de GNOME.
- Aparte de estos, merece la pena comentar:
  - Reproductores de ficheros de sonido
    - En modo texto:
      - ◇ cdp : reproductor de cd.
      - ◇ aumix: mesa de mezclas.
      - ◇ sox: para procesar ficheros de sonido, permite convertir de un formato a otro.
    - Otros:
      - cdparanoia: extractor de CD
      - BladeEnc: conversor MP3
      - Grip: Front End para BladeEnc
      - cdrecord: para grabar CD
- En la página Sound and MIDI Software For Linux <http://linux-sound.org/> hay información sobre casi todo el software musical existente para Linux.

### 9.3.2. Más periféricos

De los periféricos restantes se quedan algunos en el tintero, ya que debido al amplio rango de posibilidades que presentan es preferible tratarlos de forma “individualizada”. De nuevo, como referencia obligada para todo el hardware:

<http://www.escomposlinux.org/hardware/index.php>

<sup>18</sup>Si no lo está, en el arranque del sistema también nos lo indicará

<sup>19</sup>También está traducido en Lucas



## Scanner

Si son SCSI no presentan problema, para configurarlos se puede consultar en:

<http://www.mostang.com/sane/>

Para saber si el nuestro se ha detectado podemos usar el programa

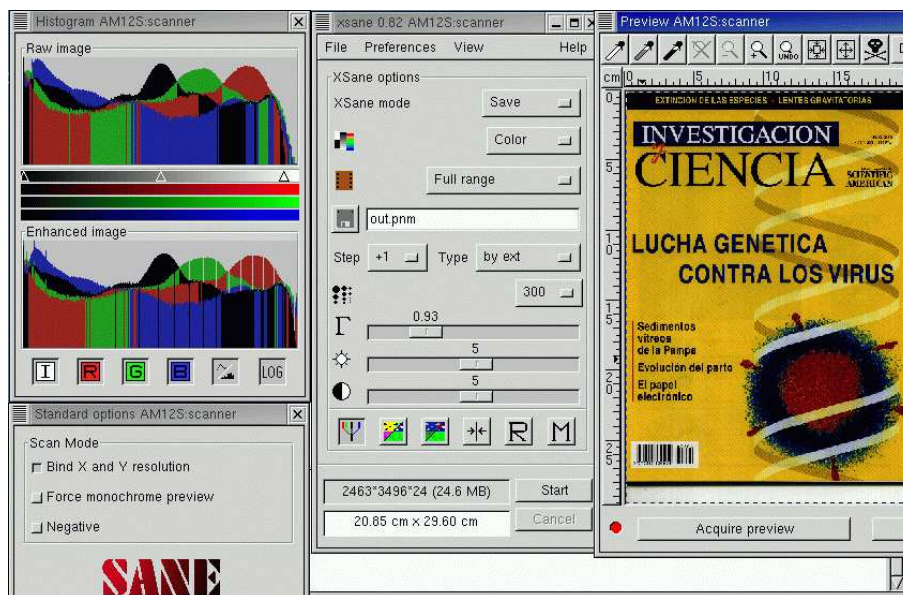
```
#xsane
```

si tras aceptar en la página de licencia nos aparece



tendremos que buscar nuestro modelo en la URL anterior y ver si se puede configurar con Linux.

Si sí lo detecta, un ejemplo:



## Grabadoras

Si son SCSI no hay problema, si son IDE se tienen que reconfigurar como SCSI (RedHat lo hace de forma automática).

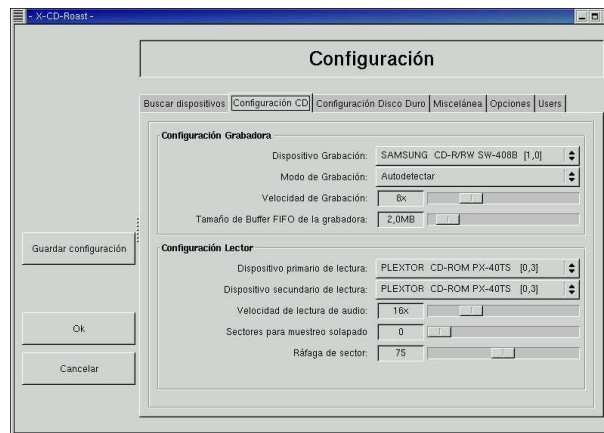
Si las detecta como CD, pero no permite grabar con ellas se puede probar a crear una imagen `initrd` y modificar los parámetros que se pasan al kernel en tiempo de arranque (véase la página man de `initrd` y del fichero `/boot/grub/grub.conf`. En el tema de inicio del sistema retomaremos este fichero). En general se detectan automáticamente y para trabajar con ellas sólo tenemos que configurar y ejecutar el programa

```
#gtoaster
```

Como el “hábito pesa mucho”, seguimos trabajando con el que se instalaba anteriormente por defecto y que está en el CD3. Si introducimos el CD, lo montamos e instalamos:

```
#mount /mnt/cdrom
```

```
#rpm -ivh /mnt/cdrom/RedHat/RPMS/xcdroast-0.98a9-18.i386.rpm
```



En caso de duda, mirar el capítulo 3 de The Official Red Hat Linux Getting Started Guide <http://europe.redhat.com/documentation/rhl8.0/rhl-gsg-es-8.0/>. Se explica detalladamente el funcionamiento de este programa.

#### Dispositivos conectados al puerto paralelo (scanner no).

Información sobre este aspecto se puede mirar en <http://gear.torque.net/parport/>

## Capítulo 10

# Paquete Ofimático: OpenOffice

Una trabajadora social que usa... veamos: un procesador de textos, correo electrónico y quizá una hoja de cálculo para hacer pequeños presupuestos. Yo creo que a estas alturas del partido esta persona tiene ya una solución completa en el Software Libre. Tenemos OpenOffice con un procesador de textos completos; la hoja de cálculo tenemos dos, Gnumeric [parte de Gnome, y programada por el propio Icaza] y OpenCalc, también de OpenOffice; para las cuestiones de web tenemos Netscape y Mozilla. Todo esto unido al escritorio, a la integración, creo que ya tenemos una solución para usuarios con necesidades muy básicas, para el grueso de la población. (MIGUEL DE ICAZA<sup>1</sup>)

### 10.1. Introducción

En el mundo Linux van apareciendo cada vez más paquetes ofimáticos y de gran calidad, entre ellos podemos destacar:

**OpenOffice** Hoy por hoy el “líder” indiscutible.

**StarOffice** El “OpenOffice” de Sun <http://es.sun.com/software/star/staroffice/6.0/index.html> (es gratuita para uso educativo)

**Applixware** (Linux Office Suite<sup>2</sup>)

**Koffice** <http://www.koffice.org><sup>3</sup>

Nosotros sólo vamos a comentar el primero ya que tiene una serie de características que lo hacen “único”:

- Es gratuito
- Tiene filtros de importación/exportación con las herramientas de Microsoft bastante elaborados
- Su interfaz de usuario está muy conseguida
- Está disponible para multitud de plataformas (Windows, Linux, etc)
- Está en castellano
- Se instala por defecto con Red Hat 8

---

<sup>1</sup>ENTREVISTA: "Nuestro objetivo es que haya más gente usando Gnome que Windows"<http://www.el-mundo.es/navegante/2000/11/21/portada/974831460.html>

<sup>2</sup>No es gratuita, se puede bajar una demo.

<sup>3</sup>No se instala por defecto.

## 10.2. Características de OpenOffice 1.0.1

OpenOffice es un paquete de oficina como los existentes para Windows de Corel o el propio Microsoft. Podemos acceder a las distintas aplicaciones que componen el paquete:

**Writer** Tratamiento de textos

**Calc** Hoja de Cálculo

**Impress** Presentaciones de diapositivas

**Draw** Tratamiento de gráficos

**Writer/Web** Editor de páginas Web

**Math** Creación de fórmulas matemáticas

Todos los módulos anteriores (los 4 primeros son los “fundamentales”) están integrados, lo que permite que se puedan ejecutar desde el mismo entorno y con menús similares.

Para trabajar con este programa sin desesperarnos, nuestro equipo debería disponer de al menos :

- Mínimo de 64 MB de memoria RAM. La verdad es que para “funcione” bien son necesarios al menos 128 MB.
- Microprocesador Pentium o compatible<sup>4</sup>
- Tarjeta aceleradora de vídeo con más de 2 MB de Ram y una resolución mínima de 800x600 .

Existen bastantes documentos traducidos sobre la forma de usar esta *suite* y se pueden conseguir en Documentos sobre OpenOffice traducidos al castellano <http://es.openoffice.org/servlets/ProjectDocumentList>. Para no sobrecargar este tema, y ya que su uso es similar al de cualquier suite ofimática, en este capítulo sólo vamos a enumerar brevemente los módulos que componen OpenOffice. Para ampliar, os remitimos a la página web antes comentada y a la ayuda que acompaña al programa.



- En la sección downloads de la Web del curso hemos puesto manuales en castellano de los módulos fundamentales del programa.
- La ayuda del programa aparece en inglés. Si deseamos que esté en castellano solo hay que bajarse el fichero `openoffice_es.tgz` desde la Web del curso y descomprimirlo/desempaquetarlo en `/usr/lib/openoffice/help/es`. Los pasos a seguir son:

1. Bajar el fichero `openoffice_es.tgz` desde el servidor.
2. Moverlo al directorio adecuado:  

```
#mv openoffice_es.tgz /usr/lib/openoffice/help/
```
3. Descomprimirlo y desempaquetarlo  

```
#tar -xzf /usr/lib/openoffice/help/openoffice_es.tgz
```

La próxima vez que ejecutemos el programa, dispondremos de la ayuda en castellano.



PEDRO REINA, profesor madrileño, ha desarrollado un curso de informática muy interesante para trabajar en el aula. La parte Ofimática está basada en OpenOffice (StarOffice) y puede ser de utilidad, se trata de Web de Pedro Reina <http://www.pedroreina.org/>

---

<sup>4</sup>Si pasa de 200 Mhz mejor que mejor.


## 10.3. El programa

### 10.3.1. Inicio del programa

Para iniciar los componentes del programa disponemos de varias opciones:

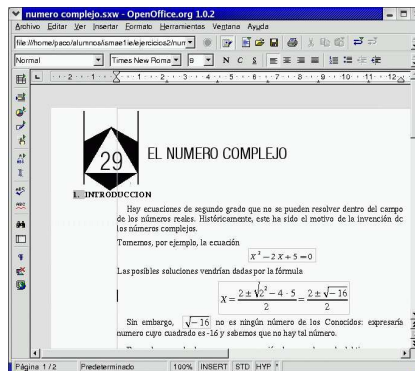
- Usando la barra de herramientas de GNOME:



-  → **Ofimática** y optaremos por cualquiera de los componentes del programa.
- Desde un terminal, ejecutando los comandos que aparecen en 8.1.3 en la página 123

### 10.3.2. Write

Posee todas las características usuales en un procesador de textos: estilos de escritura, corrección ortográfica, plantillas, macros... Permite la corrección ortográfica mientras se escribe.

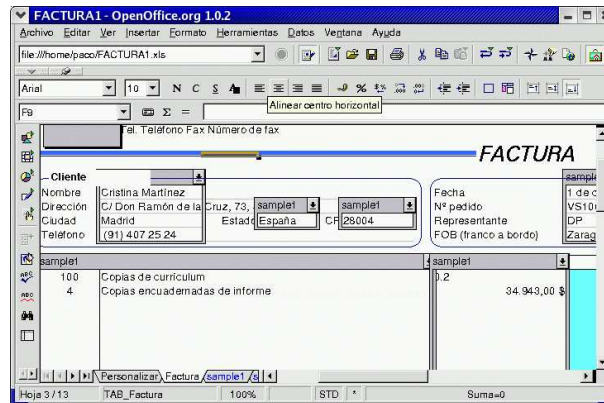


Otras características que presenta son:

- Creación de páginas Web escribiendo como si lo hiciéramos con un documento normal
- **Math**, módulo para la creación de funciones matemáticas y edición de ecuaciones
- Inserción dinámica de hiperenlaces y direcciones de correo
- Si tenemos módem podemos enviar documentos como fax. Para hacerlo directamente podemos poner un icono en la barra de funciones.
- Permite insertar imágenes en el documento (bien desde la “Galería de imágenes” del propio OpenOffice, o bien desde otras aplicaciones). También podemos insertar imágenes y fotografías digitalizadas, o digitalizar nosotros la imagen.
- Tiene funciones de dibujo (dentro del **Writer**)

### 10.3.3. Calc

Es muy similar a Excel. Tiene todas las posibilidades de una hoja de cálculo profesional<sup>5</sup>. La realización de gráficos y diagramas basados en los datos introducidos no es complicada, ya que posee un asistente que nos facilita la labor. Los filtros de exportación de Excel son bastante buenos, de hecho la captura que sigue se ha realizado sobre una hoja en formato .xls



Además, permite la importación de los datos de una base de datos.

### 10.3.4. Impress

Se trata de un editor de presentaciones. Permite crear transparencias de forma similar a como lo hace el PowerPoint. Podemos obtener:

- figuras con texto
- efectos con figuras 3D
- líneas y gráficos
- animaciones (movimiento, gifs animados)
- efectos de sonido
- inserción de objetos OLE
- exportar a HTML

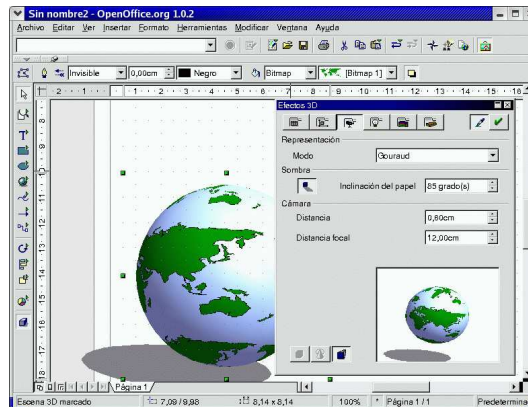
Otra característica nada desdeñable cuando empezamos a trabajar con él es la cantidad de plantillas que trae y que nos pueden facilitar bastante la labor.

<sup>5</sup>Si no necesitamos una hoja de cálculo tan potente, disponemos de una que puede solucionarnos la mayoría de problemas, se trata de **gnnumeric**, la hoja de cálculo de Gnome y se carga en la instalación básica.



### 10.3.5. Draw

Se trata de una herramienta de dibujo muy completa. Quizá una de las posibilidades más llamativa, sea la de realizar dibujos en 3D o convertir texto en 3D, con la opción de aplicarle una gran variedad de efectos.



Otra característica importante es que permite la vectorización de bitmaps. Podemos convertir los bitmaps (imágenes en pixels) en imágenes vectoriales. Estos últimos tienen la ventaja de poder ser ampliados sin que disminuya su calidad al imprimirlos. También podemos alinear y agrupar objetos.

## **Parte III**

# **Primeros pasos con Linux**



# Capítulo 11

## Inicio del Sistema

Al proceso de encendido de la computadora, es decir, el *arranque*, proviene del inglés *booting*, palabra que deriva del término *bootstrapping*, que es una alusión a la idea de que una computadora se levanta a sí misma jalándose de las correas de sus botas, como queriendo dar a entender que fragmentos pequeños de código simple arrancan a fragmentos más grandes y más complejos hasta hacer que todo el sistema acabe por funcionar. (*LINUX. Recursos para el usuario*. JAMES MOHR)

Los ordenadores actuales usan los mismos componentes generales y la misma secuencia general para iniciar un sistema operativo. Tras el control hardware del proceso de arranque, se carga el siguiente nivel de software a ejecutar (cargador de inicialización) y éste toma el control iniciando el proceso de carga del sistema operativo, comienza la carga de Linux.

Linux es “diferente” y esto se pone manifiesto ya desde el arranque del sistema, dado que dispone de una gran cantidad de opciones de configuración de inicio.

### 11.1. Gestores de arranque

#### 11.1.1. Grub

Si hemos cargado el gestor de arranque GRUB (*G*Rand *U*nified *B*oot *L*oader) en el MBR (*M*aster *B*oot *R*ecord)<sup>1</sup>, es él el que se encarga de controlar el arranque del sistema. Una vez que se inicia GRUB pasa el control del sistema al núcleo de Linux y éste a su vez al padre de todos los procesos: el proceso *init*. Pero no corramos tanto, el estudio de ese proceso corresponde al apartado siguiente.

Continuemos con el GRUB<sup>2</sup>: GRUB es un *boot loader* que permite iniciar la máquina según unas directivas concretas, además permite pasar parámetros al kernel e iniciar otros sistemas que no sean Linux.

---

<sup>1</sup>Se encuentra en el primer sector del disco (cilindro 0, cabeza 0, sector 1). Además del código necesario para iniciar la carga del sistema, contiene la tabla de particiones del disco.

<sup>2</sup>Para encontrar información más amplia sobre GRUB consultar el Capítulo 4 de The Official Red Hat Linux Reference Guide <http://europe.redhat.com/documentation/rhl8.0/rhl-rg-es-8.0/>



Tras cargarse en la memoria RAM la parte de GRUB contenida en el MBR se inicia la carga del fichero de configuración del gestor de arranque (`/boot/grub/grub.conf`). Si todo el proceso es correcto veremos en nuestro monitor el logotipo de RedHat y un menú con el que poder cargar el sistema operativo que deseemos.

Sea cual sea el entorno elegido, si dejamos pasar un tiempo él sólo comienza la secuencia de arranque de Linux.

Pero, STOP; congelemos el proceso de arranque y analicemos un poco qué podemos hacer:

### 11.1.2. Interfaces de GRUB

#### De Menú

Con esta “ventanita” inicial ya tenemos trabajo suficiente. Si nos fijamos en el gráfico podemos ver que en el texto inferior, además de indicarnos la posibilidad de usar las flechas de cursor nos avisan de que podemos pulsar la tecla

- e** para modificar los parámetros de arranque del sistema operativo resaltado
- a** para modificar o añadir parámetros al núcleo
- c** para acceder a la interfaz de línea de comandos.

#### Del editor de entrada de menú

Si pulsamos **e** en la ventana anterior veremos algo similar a:



Desde ella podemos:

- o agregar una línea de comandos después de la línea en vídeo inverso
- O agregar una línea antes de la línea actual
- e modificar la línea actual
- d elimina la línea actual
- b iniciar el arranque del sistema
- Esc para omitir los cambios y volver a la pantalla anterior
- c para acceder a la interfaz de línea de comandos.

### De línea de comandos

Si desde la pantalla inicial optamos por pulsar sobre **c**, accedemos a ventana de línea de comandos. Para ver qué comandos tenemos disponibles podemos usar **help**:



Soporta la autocompletación de comandos (igual que las shell de Linux) mediante la tecla **[Tab]**<sup>3</sup>

<sup>3</sup>Es decir, no tenemos que conocer el nombre completo de un comando, si escribimos las letras iniciales de ese comando y después **TAB** se autocompleta el comando si hay uno sólo con esas letras iniciales, si hay varios se muestran en pantalla para ver cuál elegimos.



Aclaremos cómo interpreta GRUB las particiones de nuestros discos duros:

hd0 primer disco duro (IDE o SCSI)

hd1 segundo disco duro

...

En cuanto a las particiones se numeran secuencialmente desde el número 0.

Así por ejemplo (hd1, 3) representaría la cuarta partición del segundo disco duro de nuestro sistema

### ➡ Para practicar

Para poder realizar el ejemplo que sigue hay que adecuarlo a nuestro sistema:

- la línea 3 parte de la base de que tenemos un sistema Windows 9x en la primera partición del maestro del primer canal IDE
- el cuarto comando sólo se podrá ejecutar tal cual si /boot está en la segunda partición del primer disco duro.
- Otra nota: cuidado con el teclado, hay que localizar las teclas adecuadas ya que todavía no puede estar en “castellano”.

Ejecutar desde la ventana de línea de comandos de GRUB:

```
grub>help
grub>displaymem
grub>cat (hd0,0)/autoexec.bat
grub>cat (hd0,1)/boot/grub/grub.conf
```

Más adelante, cuando analicemos los niveles de arranque estudiaremos algunos parámetros que se le pueden pasar al núcleo. Por ahora, permitamos que GRUB continúe su labor y que pase el control al núcleo de Linux.

```
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 126k freed
VFS: Mounted root (ext2 filesystem).
Red Hat nash version 3.4.28 starting
Loading jbd module
Journalled Block Device driver loaded
Loading ext3 module
Mounting /proc filesystem
Creating block devices
Creating root device
Mounting root filesystem
kjournald starting. Commit interval 5 seconds
EXT3-fs: mounted filesystem with ordered data mode.
Freeing unused kernel memory: 200k freed
INIT: version 2.84 booting
       Welcome to Red Hat Linux
       Press 'I' to enter interactive startup.
Mounting proc filesystem: [ OK ]
Unmounting initrd: [ OK ]
Configuring kernel parameters: [ OK ]
Setting clock (localtime): dom abr 13 09:27:14 CEST 2003 [ OK ]
Loading default keymap (es): [ OK ]
Setting default font (latarcyrheb-sun16): [ OK ]
Setting hostname pci: _
```

### 11.1.3. Fichero /boot/grub.conf

Le ventaja que presenta GRUB sobre otros gestores de arranque (LILO por ejemplo) es que GRUB no necesita que su fichero de configuración esté en el MBR, es decir, sólo se instala el programa y él se encarga de leer la configuración desde un fichero externo localizado en algún lugar de nuestro sistema. Además, usar este método permite que:

- Si nos equivocamos en la configuración de GRUB siempre podremos arreglarlo con los interfaces anteriores.



- Cada vez que modificamos el fichero de configuración de GRUB no tenemos que sobrescribir el MBR.

¿Y cuál es el fichero de configuración de Grub? pues el fichero `/boot/grub/grub.conf`

Analicemos el de mi sistema (es distinto del que tenéis en vuestros quipos)

Las líneas que comienzan con `#` son comentarios (algo que ya veremos es estándar en Linux) que se han añadido para explicar el fichero.

```
$ cat /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You do not have a /boot partition. This means that
# all kernel and initrd paths are relative to /, eg.
# root (hd0,1)
# kernel /boot/vmlinuz-version ro root=/dev/hda2
# initrd /boot/initrd-version.img
#boot=/dev/hda
# sistema operativo por defecto, en mi caso el de título Red Hat ...
default=0
# tiempo de espera (en segundos) antes de iniciar la carga del sistema de forma
automática
timeout=10
# gráfico de fondo por defecto (el fondo azul con ceros y unos)
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
# texto que veremos en el menú
title Red Hat Linux (2.4.18-14)
    # Establece y monta la partición "root": partición de arranque
    root (hd0,1)
    # núcleo a cargar y opciones de arranque: tengo una grabadora
    kernel /boot/vmlinuz-2.4.18-14 ro root=/dev/hda2 hdc=ide-scsi
    # carga un disco RAM inicial (imagen initrd necesaria para dispositivos SCSI
    # y para el poder usar el sistema de ficheros ext3)
    initrd /boot/initrd-2.4.18-14.img
    # submenú de arranque del Windoze, y texto que veremos en el menú principal
    title DOS
    # establece la partición de arranque pero sin montarla
    rootnoverify (hd0,0)
    # carga el fichero del primer sector de la partición 0 del disco 0: partición
Windoze
    chainloader +1
```

¿Cómo puede ser el fichero de configuración si en vez de dos sistemas son tres?, un ejemplo en el que por defecto se carga el sistema DOS (`default=1`) podría ser:

```
$ cat /boot/grub/grub.conf
default=1
timeout=10
password --md5 $1$6U7Tn/$Pr3691BPbD/9nD8DZMg5A/
splashimage=(hd0,2)/boot/grub/splash.xpm.gz
title Red Hat Linux (2.4.18-14)
    root (hd0,2)
    kernel /boot/vmlinuz-2.4.18-14 ro root=/dev/hda2 hdc=ide-scsi
    initrd /boot/initrd-2.4.18-14.img
title DOS
    rootnoverify (hd0,0)
    chainloader +1
title XP
```



```
rootnoverify (hd0,1)
chainloader +1
```

Una nota sobre este fichero: cuando se instala Red Hat aparece la posibilidad de poner una contraseña al gestor de arranque, esa contraseña es la que aparece encriptada en la línea<sup>4</sup>:

```
password --md5 $1$6U7Tn/$Pr3691BPbD/9nD8DZMg5A/
```

Si usamos esta opción, es necesario introducirla (con **p**) para poder acceder a las opciones extendidas de GRUB.

Para desactivar esta opción sólo tenemos que comentarla con **#** o borrar la línea. Si no pusimos ninguna contraseña, para impedir al acceso a los comandos de GRUB en el arranque, podemos generar la cadena asociada a una contraseña usando el comando:

```
$/sbin/grub-md5-crypt
```

Después de introducir la contraseña nos la devolverá encriptada con el algoritmo MD5, esa es la cadena que hay que introducir tras **md5**.



GRUB se instala por defecto y, al mantener un fichero de configuración aparte no es necesario tener que “tocarlo”. En el caso de tengamos que eliminarlo podemos optar por reiniciar el sistema con un disco de Win 9x/ME y ejecutar el comando **fdisk /MBR**

Si trabajamos con otro gestor de arranque y, una vez instalado Linux deseamos cambiar a GRUB, lo podemos hacer usando el comando **grub-install**. Cuando el fichero de configuración de GRUB esté ajustado a nuestro sistema, lo instalaremos en el MBR con:

```
#grub-install /dev/hda
```



Si no hicimos un disco de inicio durante la instalación de Red Hat y GRUB no se instaló bien, podemos hacer un disco/installar GRUB como sigue:

1. Iniciamos el ordenador desde el lector de CDs con el CD1 de la distribución. Cuando aparezca boot: escribiremos  

```
boot: linux rescue
```

para entrar en el modo de rescate<sup>5</sup>.

2. Cuando aparezca el prompt de introducción de ordenes ejecutar<sup>6</sup>:  

```
#chroot /mnt/sysimage
```

3. Podemos optar por:

**Instalar GRUB:** Con el editor **vi** podemos modificar el fichero **/boot/grub/grub.conf** hasta ajustarlo a nuestra máquina, una vez que se adecúe a nuestro sistema ejecutaremos:

```
#grub-install /dev/hda
```

**Creación de un disco de inicio:** Introduciremos un disco en la unidad de disquettes y ejecutaremos el comando<sup>7</sup>:

```
#mkbootdisk --device /dev/fd0 2.4.18-14
```

4. Para terminar sólo tenemos que ejecutar el comando **exit** (un par de veces).

Para saber más: el capítulo del manual antes comentado y el comando:

```
$info grub
```

## ➡ Para practicar

<sup>4</sup>La contraseña utilizada es "linux".

<sup>5</sup>No presenta mayor problema: sólo hay que seleccionar el idioma y optar por las opciones por defecto.

<sup>6</sup>Nos permite trabajar como si el árbol de ficheros partiera de ese directorio, se trata del directorio en dónde se monta nuestro sistema Linux.

<sup>7</sup>Los números finales se han de corresponder con la versión del núcleo instalada. Podemos saber cuál es usando el comando:

```
uname -a
```

1. Si tenemos dos sistemas operativos instalados, modificar el fichero `/etc/grub.conf` para que cambie el sistema de arranque por defecto:
  - Bien cambiando la línea: `default`
  - Bien permutando los bloques existentes para los sistemas operativos.
2. Añadir una contraseña a GRUB que impida que se pueda acceder a las opciones de edición sin más.

El apartado que sigue lo hemos mantenido como información añadida. Sólo puede ser de utilidad si cambiamos el gestor de arranque por defecto cuando instalamos Linux. Por tanto, lo más normal es que tengamos GRUB instalado y en ese caso “saltemos” hasta 11.2 en la página 157.

#### 11.1.4. Por si se opta por LILO.

LILO es otro gestor de arranque que permite controlar qué núcleo o sistema operativo va a ser cargado e iniciado. Con él tenemos la posibilidad de pasar parámetros al kernel usando el archivo `/etc/lilo.conf`.<sup>8</sup> Para encontrar información más amplia sobre LILO, se puede consultar en:

- `/usr/share/doc/lilo-21.4.4`, en este subdirectorio, además de ejemplos tenemos información bastante amplia sobre él así como el manual de usuario.
- *Linux Instalación y Primeros Pasos*, de MATT WELSH.
- Los HOWTOs/mini-HOWTOS:
  - Linux-DOS-Win95-OS2.Como
  - Linux-NT-Loader
  - Discos-Grandes-Como
  - mini-HOWTO LILO
  - HOWTO BootPrompt
  - HOWTO BootDisk
- En <http://www.linux-es.com/Faq> está la FAQ sobre Linux para principiantes, en su punto 4<sup>9</sup> hace un estudio sobre él. En esa FAQ podemos encontrar un trabajo bastante bueno sobre LILO, de hecho, bastantes de las “ideas” que vienen a continuación están sacadas de él.

Cuando se ejecuta LILO, a la vez que se van mostrando las letras en la pantalla de nuestro ordenador se van cargando una serie de componentes de forma secuencial, si todo el proceso es correcto<sup>10</sup> veremos en nuestro monitor el logotipo de RedHat y un menú con el que poder cargar el sistema operativo que deseemos.

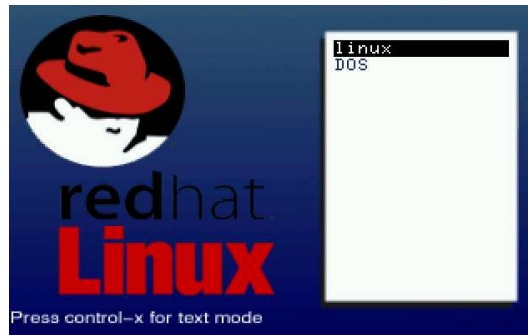
---

<sup>8</sup> `/etc/lilo.conf`, `anaconda`

<sup>9</sup> Sobre LILO

<sup>10</sup> Si surgen problemas en el arranque la presencia o ausencia de las letras de LILO nos permitirá diagnosticar el porqué del fallo. En la FAQ anteriormente mencionada se explica esto como sigue:

- **"L" error:** El primer tramo del código ha sido cargado pero no el segundo. Esto se debe generalmente a un error físico en el sector de botado o a un problema de geometría del disco.
- **"LI":** El segundo tramo del código ha sido invocado, pero no ha podido iniciarse. Esto sucede cuando hay un error de geometría de disco o se ha movido `/boot/boot.b` sin reinstalar LILO (`/sbin/lilo`).
- **"LIL":** El segundo tramo del código se ha iniciado pero no puede ubicar los datos que necesita en el fichero de mapeado. Esto puede deberse a un error físico del dispositivo de arranque o a un problema en la geometría del disco.
- **"LIL?":** El segundo tramo del código se ha cargado en una dirección de memoria equivocada. Esto se debe a un error en la geometría del disco o cuando se ha movido `/boot/boot.b` sin reinstalar LILO (`/sbin/lilo`).
- **"LIL-":** Los datos en el fichero de mapeado no son válidos. Las causas son las mismas que en el caso anterior.



Si no nos gusta ese entorno gráfico, podemos pasar a modo texto pulsando **[Ctrl]+[x]** y veríamos:

LILO boot:

Si ahora pulsamos sobre la tecla TAB nos aparecen las etiquetas de los sistemas operativos que tenemos en nuestra máquina, por ejemplo

linux                      dos

si pulsamos *intro* entraremos en el Sistema que hayamos definido por defecto.<sup>11</sup>

Sea cual sea el entorno elegido, si dejamos pasar un tiempo él sólo comienza la secuencia de arranque de Linux.

Podemos modificar los parámetros de LILO modificando el fichero `/etc/lilo.conf`, para hacer esto podemos escribir sobre él directamente o usar `linuxconf`.

Comentemos cómo modificar algunos parámetros de forma manual<sup>12</sup>, supongamos que nuestro fichero `/etc/lilo.conf` es:

```
#sección Global
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
timeout = 30
linear
prompt
message = /boot/message
default = linux
vga = normal

#sección del sistema Linux
image = /boot/vmlinuz-2.4.18-14
    label = linux
    read-only
    root = /dev/hdb1
    append = "root=LABEL=/"

#sección del sistema Windows
other = /dev/hda1
    label = dos
```

este fichero está dividido en dos partes, una parte común y luego dos secciones, una para cada sistema operativo de esta máquina, además podemos ver que si queremos poner algún comentario se usa el mismo símbolo de siempre (#)

---

<sup>11</sup>Si no hemos modificado nada en el proceso de instalación por defecto entramos en Linux y si tenemos Windows podremos acceder a él escribiendo

dos

<sup>12</sup>Como ya se ha dicho antes, siempre que queramos modificar la configuración del sistema tendremos que trabajar como root.



### Sección Global

- `boot=/dev/hda` indica que LILO se carga desde el MBR del maestro del primer canal ide
- `map=/boot/map` ubicación del fichero de mapeado.
- `install=/boot/boot.b` instala el fichero como nuevo sector de arranque.
- `timeout=30` tiempo (en décimas de segundo) que LILO debe esperar antes de comenzar a arrancar el sistema operativo que tenga definido por defecto.
- `linear` permite que las referencias a los sectores se escriban como direcciones lógicas en lugar de físicas. Se emplea cuando LILO no reconoce correctamente la geometría del disco duro (debido a un remapeado por parte del BIOS).
- `prompt` hace que veamos LILO boot: y que podamos seleccionar el sistema operativo
- `message=/boot/message` fichero con el gráfico que se muestra antes del indicador de arranque. Se corresponde con el logo de RedHat y el inicio “gráfico” del LILO.
- `default=linux` etiqueta del sistema a cargar por defecto, en este ejemplo si queremos comenzar con el otro sistema tendríamos que poner `dos`
- `vga=normal` especifica el modo de texto VGA que debe usarse al arrancar el sistema. Al poner `normal` seleccionamos el modo de texto 80×25.

### Sección linux

- `image=/boot/vmlinuz-2.4.18-14` ruta completa del fichero que contiene el kernel de Linux.
- `label=linux` etiqueta de este sistema, podemos poner los que nos plazca pero limitado a cadenas de 15 caracteres.
- `read-only` indica al kernel que monte inicialmente la partición raíz en modo de sólo lectura.
- `root=/dev/hdb1` nombre de la partición linux nativa, en este caso, es la primera partición del segundo disco duro.

### Sección otros ...

Si sólo tenemos Linux esta sección no aparecería. Existen más opciones para esta sección además de las aquí listadas.

- `other= /dev/hda1` donde indicamos esta vez la partición donde está cargado el otro sistema operativo.
- `label=dos`

### Otras (solo algunas) que no aparecen en el fichero de ejemplo

- `lba32` es “incompatible” con `linear`. Con esta línea “puede<sup>13</sup>” que podamos trabajar con LILO con discos grandes (más de 8.4 GB) en los que hayamos puesto el kernel en cualquier partición independientemente del cilindro de inicio.
- `append=parámetro` permite pasar parámetros y componentes de hardware al kernel como cadena de caracteres. Por ejemplo, puede que para que Linux reconozca dos tarjetas de red haya que pasarle aquí los parámetros adecuados para que reconozca la segunda tarjeta.
- `password=contraseña` contraseña que permite cargar LILO.

---

<sup>13</sup>Tanto la BIOS como el disco han de admitir transferencias de bloque EDD.

Si modificamos el fichero `/etc/lilo.conf` es necesario ejecutar:

```
# lilo14
```

para que lea los cambios del fichero y actualice lo que allí se le indica. La etiqueta marcada con un asterisco será la correspondiente al sistema operativo de arranque por defecto.

Si queremos desinstalar LILO de nuestra máquina tenemos varias opciones:

- desde Windows ejecutar

```
fdisk /MBR
```

- desde Linux ejecutar

```
# lilo -u
```

y restauramos el MBR anterior a la instalación de LILO

El comando `lilo` admite bastantes opciones que se pueden consultar en la página man del programa.

## 11.2. Fichero /etc/inittab

Comentábamos en la sección anterior que el GRUB/LILO pasa el control de la máquina al núcleo y comienza la autodetección del hardware de nuestra máquina y carga los drivers de dispositivos. Toda la información que se genera la tenemos a nuestra disposición cada vez que arrancamos Linux, va apareciendo en el monitor de nuestro equipo.<sup>15</sup>

```
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 126k freed
VFS: Mounted root (ext2 filesystem).
Red Hat nash version 3.4.28 starting
Loading jbd module
Journalled Block Device driver loaded
Loading ext3 module
Mounting /proc filesystem
Creating block devices
Creating root device
Mounting root filesystem
kjournald starting. Commit interval 5 seconds
EXT3-fs: mounted filesystem with ordered data mode.
Freeing unused kernel memory: 200k freed
INIT: version 2.84 booting
       Welcome to Red Hat Linux
       Press 'l' to enter interactive startup.
Mounting proc filesystem: [ OK ]
Unmounting initrd: [ OK ]
Configuring kernel parameters: [ OK ]
Setting clock (localtime): dom abr 13 09:27:14 CEST 2003 [ OK ]
Loading default keymap (es): [ OK ]
Setting default font (latarcyrheb-sun16): [ OK ]
Setting hostname pci: _
```

Si la cosa va bien comienza el proceso *init*: es el padre de todos los demás procesos, su finalidad es la de generar los procesos nucleares que necesita el sistema. Para iniciar el resto del sistema, *init* utiliza el fichero de configuración<sup>16</sup> `/etc/inittab`. Mediante este fichero podemos ajustar el inicio del sistema para que se ejecute según nuestros intereses.

Para el que quiera profundizar sobre el tema puede consultar:

### 1. La página man del fichero `inittab`

<sup>14</sup>`lilo` se puede ejecutar con las opciones `-t` (test, prueba) o `-q` (query, consulta) para mostrar lo que LILO haría durante un inicio real.

<sup>15</sup>Si después queremos acceder a esa salida por pantalla podemos usar la orden

```
$ dmesg
```

Para poder ver mejor esa salida mejor si usamos

```
$ dmesg | less
```

<sup>16</sup>Se trata de un fichero de configuración y no de un script (fichero en el que hay un conjunto de órdenes del sistema operativo que se van ejecutando secuencialmente, de igual manera que si las vamos escribiendo una detrás de otra desde la línea de comandos), es decir, el orden en el que se escriben las líneas no es significativo.



2. El capítulo 3 de *The Official RedHat Linux Linux Reference Guide*, ahí viene explicado cómo realizar cada una de estas acciones.

Un ejemplo de fichero /etc/inittab<sup>17</sup> es:

```
#
# inittab          This file describes how the INIT process should set up
#                  the system in a certain run-level.
#
# Author:          Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#                  Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
# Default runlevel. The runlevels used by RHS are:
#  0 - halt (Do NOT set initdefault to this)
#  1 - Single user mode
#  2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#  3 - Full multiuser mode
#  4 - unused
#  5 - X11
#  6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
ud::once:/sbin/update

# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
# xdm is now a separate service
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Las entradas de este fichero son de la forma:

etiqueta:niveldeejecución:acción:comando  
donde:

**etiqueta** etiqueta única de un máximo de 4 caracteres

**niveldeejecución** lista de niveles de ejecución (se ve después) en lo que se ejecuta el proceso

**acción** explicaremos sólo las que hay en el fichero (se ha mantenido el orden en que aparecen en él), para el resto de acciones mirar en los documentos de ampliación ya comentados

<sup>17</sup>Como siempre los # indican líneas de comentarios



**initdefault** nivel de ejecución por defecto

**sysinit** el *script* se ejecuta en la fase de arranque del sistema, independientemente del nivel de ejecución<sup>18</sup>.

**once** el comando se ejecuta sólo una vez después de entrar en el nivel de ejecución seleccionado

**ctrlaltdel** comando a ejecutar si se pulsa la combinación de teclas<sup>19</sup> [CTRL]+[ALT]+[DEL]

**powerfail** comando a ejecutar si se recibe una señal SIGPWR (indica algún problema con la alimentación eléctrica)

**powerokwait** si el fichero /etc/powerstatus contiene la palabra OK e *init* recibe la señal SIGPWR se ejecuta el comando. Es decir, se ejecuta el comando si el UPS manda un señal indicando que se ha restablecido el nivel de energía eléctrica.

**respawn** se inicia una nueva instancia del proceso cada vez que termine

**wait** se ejecuta el comando una sola vez al iniciar el nivel de ejecución. Pero hasta que no termina el comando no se hace nada más.

**comando** comando a ejecutar

Hay que tener mucho cuidado a la hora de modificar este fichero ya que nos arriesgamos a que Linux no pueda arrancar.

### ➡ Para practicar

1. Accede a la cuenta del root<sup>20</sup>
2. Antes de seguir deberíais fijaros que hay dos líneas en el fichero /etc/inittab de ejemplo que en el de vuestro ordenador no están comentadas, se trata de:

```
#5:2345:respawn:/sbin/mingetty tty5
```

```
#6:2345:respawn:/sbin/mingetty tty6
```

Si se comentan y se reinicia el equipo, se puede comprobar que el número de terminales que podemos tener abiertos ya no es de 6 sino que sólo dispondremos de 4<sup>21</sup>. Para eso ejecutar:

- Antes de reiniciar [Alt]+[Ctrl]+[Fi] *i* = 1.,6
- Después de reiniciar [Alt]+[Ctrl]+[Fi] *i* = 1.,6

De esta forma, en ordenadores con no demasiados recursos podemos ahorrar algo de memoria y aumentar un poco el rendimiento de nuestro equipo.

No vamos a entrar en la modificación de todos los aspectos que gestiona este fichero, tan sólo vamos a comentar cómo modificar los niveles de arranque.

<sup>18</sup>/etc/rc.d/rc.sysinit → A “grosso modo” podemos decir que se encarga de configurar/cargar todo el sistema (los OK del inicio).

<sup>19</sup>Merece la pena fijarse en la línea:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

si la mantenemos permitimos que cualquier usuario pueda hacer un reset con nuestra máquina con sólo pulsar [CTRL]+[ALT]+[DEL].

<sup>20</sup>Con el comando *su* se puede acceder a la cuenta del root sin abrir otra sesión. Es decir, si introducimos:

```
$su
```

y después la contraseña del root podremos trabajar como el “jefe” sin tener que abrir otra sesión de usuario. Cuando terminemos se sale de esta subshell con:

```
#exit
```

<sup>21</sup>Esto origina a su vez un cambio en cuál es la combinación de teclas que permite acceder al modo gráfico cuando usemos el sistema X ¿cuáles serían?.



### 11.2.1. Niveles de arranque

Al inicio de este fichero aparecen los distintos niveles de arranque de los que disponemos. Cada nivel de arranque permite configurar el sistema de forma diferente. Los niveles de arranque que tenemos a nuestro alcance son 6<sup>22</sup>. El objetivo de cada uno de estos niveles es:

- 0 para apagar el sistema
- 1 para entrar en modo monousuario (equivale a *s* → *single mode*)
- 2 modo Multiusuario, sin sistemas de ficheros de red.
- 3 modo Multiusuario completo (inicio en modo texto)
- 4 No usado
- 5 modo Multiusuario completo (inicio en modo gráfico)
- 6 para reiniciar el sistema.

Al entrar en cada nivel de ejecución se ejecutan una serie de comandos que se encuentran en el fichero `/etc/rc.d/rc`, estos comandos reciben como argumento el nivel de ejecución. Por ejemplo, si el nivel de arranque es el 5, `init` ejecuta

```
/etc/rc.d/rc 5
```

Este script ejecuta entonces los ficheros que se encuentran en el subdirectorio<sup>23</sup> `/etc/rc.d/rc5.d`. Los ficheros de este subdirectorio<sup>24</sup> comienzan por S (*start*) o K (*kill*), seguidos de dos dígitos y la función que realizan. Se comienzan ejecutando los que comienzan por K y después los que comienzan por S. Los K\* sirven para parar subsistemas (servicios) y los S\* para arrancarlos<sup>25</sup>.

Parace claro que los niveles de arranque que nos puede interesar modificar son el 1, 2, 3 y 5. Veamos como cambiar de uno a otro. Si nos fijamos en el fichero `/etc/inittab` anterior veremos que hay una línea de la forma

```
id:3:initdefault
```

mientras que esté esa línea así arrancaremos nuestro sistema en modo multiusuario completo en modo texto. Si lo que deseamos es que el sistema arranque directamente en modo gráfico sustituiremos esa línea por:

```
id:5:initdefault
```

La próxima vez que arranquemos el ordenador accederemos a una consola gráfica para introducir nuestro nombre de usuario y contraseña.

#### Más formas de modificar los niveles de arranque

La forma más sencilla es usando el comando:

```
#init [n]
```

donde *n* es un número que indica el nivel de arranque que deseamos “activar”

#### Al iniciar el sistema con Grub.

<sup>22</sup>Sólo analizaremos los *stándar*

<sup>23</sup>En realidad en el subdirectorio lo único que hay son enlaces simbólicos a otros ficheros (podemos entender un enlace simbólico como un acceso directo de Windows) que están en `/etc/rc.d/init.d`.

<sup>24</sup>Lo mismo que todos los de la forma `/etc/rc.d/rcn.d` donde *n* es un número comprendido entre 0 y 6.

<sup>25</sup>El comando

```
$ /sbin/chkconfig -list
```

enumera todos los subsistemas y si están en *on* o en *off* en cada nivel de ejecución.

## ➡ Para practicar

### 1. Niveles de arranque

- Accede a la cuenta del root
- Modificar el fichero /etc/inittab de vuestro sistema para que arranque para un nivel 3
- Reinicia el sistema y comprueba que es así.
- Cambia a nivel 5 ejecutando  
`#init 5`
- Reinicia el sistema y usa GRUB para entrar en nivel 5, para eso, cuando aparezca el menú principal de GRUB pulsa la tecla [**e**] y en la ventana de edición escribe un 5 en la segunda línea tal cual aparece en el gráfico (sólo el 5 que aparece tras el fichero que contiene al núcleo, el resto de la línea no debe de aparecer igual en tu ordenador):



### 2. Veamos como cambiar la contraseña del root (por si se nos olvida):

- Usando GRUB iniciar el sistema en modo 1 (monusuario)
- Usar el comando:  
`#passwd`  
y listo, cambiar a nivel 5 con  
`#init 5`

Contraseña cambiada. Moraleja: el único sistema medio seguro es aquel al que no tenga nadie acceso

### 3. Para terminar dejar otra vez el fichero /etc/inittab como estaba al principio (nivel de ejecución 5)

**Sólo por si tenemos LILO.** Para modificar de un nivel de arranque a otro tenemos como primera opción ejecutar

`linux número_nivel_arranque26`

<sup>26</sup>En el supuesto de que la etiqueta de nuestro sistema Linux sea linux

cuando aparece<sup>27</sup>


LILLO boot:

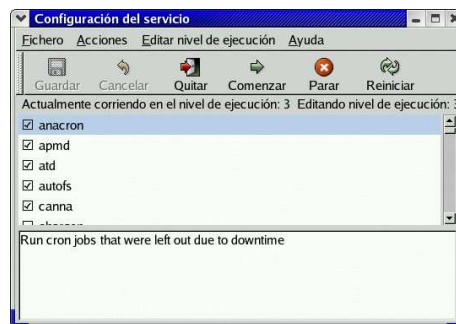
al inicio del sistema. Por ejemplo, si nosotros siempre entramos en modo gráfico y deseamos hacerlo en modo texto escribiríamos<sup>28</sup>:

LILLO boot: linux 3

### 11.2.2. Control de acceso a servicios

Vamos a enumerar las herramientas de que disponemos en RedHat para poder activar/desactivar determinados servicios según los distintos niveles de ejecución, se trata de<sup>29</sup>:

# **redhat-config-services** (desde Gnome:  → **Configuración de servidores** → **Servicio**), es una utilidad gráfica que permite seleccionar qué servicios están en activo para los niveles 3, 4 y 5. Permite reiniciarlos, pararlos y activarlos desde el propio programa.



También se puede acceder al programa escribiendo el comando `serviceconf`.

# **/usr/sbin/ntsysv** (se puede acceder a él desde el programa `setup`, opción **System services**). Los cambios no se activan en el momento, además afecta sólo al nivel desde el que se ejecuta.<sup>30</sup>



<sup>27</sup>Con [Ctrl]+[x] se puede conseguir.

<sup>28</sup>Merece la pena aclarar que podemos pasar de un nivel a otro ejecutando:

```
# init "nuevo_nivel"
```

Así, si estamos en nivel 3 y deseamos pasar el 5 escribiríamos:

```
# init 5
```

<sup>29</sup>El comando a ejecutar es el que aparece en negrita

<sup>30</sup>Para conseguir que "afecte" a los niveles 3 y 5 por ejemplo usar:

```
# ntsysv --levels 35
```

# `/sbin/chkconfig` utilidad en línea de comandos para activar/desactivar servicios. En general es más fácil trabajar con las dos anteriores.

La configuración de estos servicios se puede modificar en el arranque usando los ficheros contenidos en el subdirectorio `/etc/sysconfig`, para conocer qué se puede hacer con cada uno de ellos os remitimos al fichero `/usr/share/doc/initialscripts-6.95/sysconfig.txt` y al capítulo 3 de *The Official Red Hat Linux Reference Guide*.



Además de los programas anteriores, si tenemos un servicio en nuestra máquina, con:

```
$ /etc/rc.d/init.d/service
```

podemos comprobar qué parámetros admite, por ejemplo, con el servidor de impresión obtendríamos:

```
$ /etc/rc.d/init.d/lpd
```

```
Uso: /etc/rc.d/init.d/lpd {start|stop|restart|condrestart|reload|status}
```

O sea que si queremos pararlo sólo hay que ejecutar:

```
# /etc/rc.d/init.d/lpd stop
```

```
Parando lpd: [ OK ]
```

### 11.2.3. Distintos programas para arrancar en modo gráfico

Si accedemos a nuestra máquina en modo gráfico, podemos iniciar ese modo con distintos programas. Disponemos de tres programas que nos permiten registrarnos en el sistema, son:

**x****dm** X Display Manager


**g****dm** GNOME Display Manager

**k****dm** KDE Display Manager

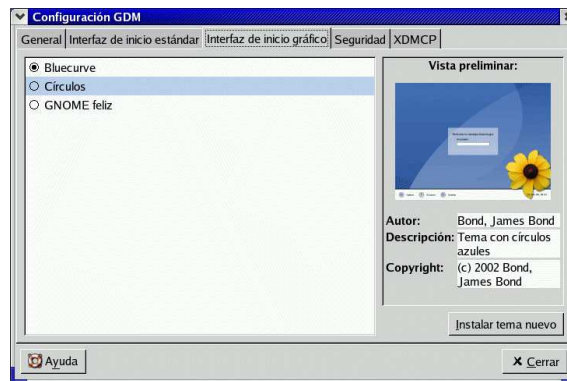
Al menos los dos primeros los tendréis todos instalados, el tercero lo tendréis si ya habéis instalado el KDE, si no es así cuando se instale podréis probar lo que se comenta de él a continuación.

Al iniciar Linux en nivel 5 (modo gráfico) habéis estado utilizando el programa **gdm**:



La configuración de **gdm** se consigue pulsando sobre  → **Configuración del sistema** → **Pantalla de registro**.

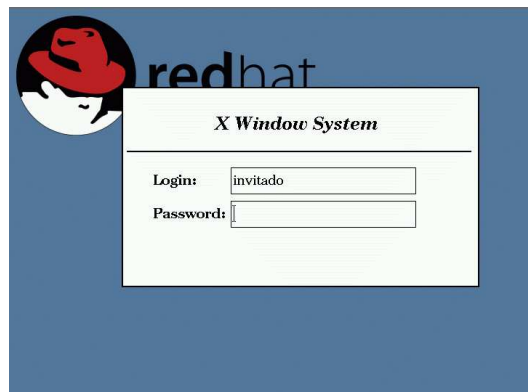




Con esta utilidad podemos personalizar gdm tanto para conexiones locales como remotas. Por defecto la conexiones remotas están desactivadas, si deseamos permitir las hay que marcar la casilla **Activar XDMCP** en la pestaña **XDMCP** e iniciar el sistema en nivel 5. Los cambios efectuados se almacenan en el fichero `/etc/X11/gdm/gdm.conf`

Si deseamos “probar” como quedan los otros<sup>31</sup> tendremos que editar el fichero `/etc/sysconfig/desktop`<sup>32</sup>, la línea a modificar es la de la variable `DISPLAYMANAGER`<sup>33</sup>

```
pondremos
DISPLAYMANAGER="GDM"
para que se inicie gdm
DISPLAYMANAGER="XDM"
para xdm
```

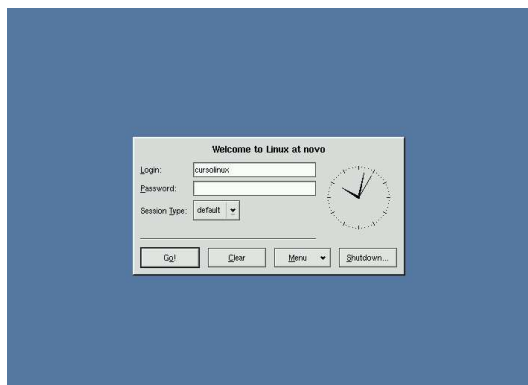


```
DISPLAYMANAGER="KDE"
para kdm
```

<sup>31</sup> Hay más formas de hacer esto pero sólo vamos a comentar la “más corta”.

<sup>32</sup> Para entender esto un poco más y para ampliar podéis mirar el fichero `/etc/X11/prefdm`

<sup>33</sup> En versiones anteriores de Red Hat la variable era `DESKTOP`. Si bien sigue apareciendo por defecto hay que cambiarla por la aquí comentada.



➡ **Para practicar:**

Comprobar lo aquí expuesto sin tener que reiniciar el sistema (se puede usar sólo `#init nivel_arranque`).

## Capítulo 12

# Sistemas de ficheros en Linux

Toda entidad física y lógica en Linux se representa como un archivo en el sistema de archivos de Linux. Las entidades físicas incluyen discos, impresoras y terminales; las entidades lógicas incluyen directorios y archivos normales, en los que se almacenan documentos y programas. (*Linux*, JACK TACKETT JR. Y DAVID GUNTER)

### 12.1. Introducción.

Revisemos los conceptos de formateo de un disco y de sistema de ficheros. En MS-DOS, lo que denominamos formatear (por ejemplo, un floppy) comprende en realidad estos dos procesos: el formateo de bajo nivel y la creación del sistema de ficheros.

*Formatear* es el proceso de escribir marcas en el medio magnético de un disco para distinguir las pistas y sectores, que posteriormente pueden ser accesibles por su localización. Para los discos duros suele venir ya de fábrica. Sobre ese disco ya formateado se pueden establecer particiones.

Crear un *sistema de ficheros* consiste en generar las estructuras de datos que un sistema operativo (como Linux o Windows) utiliza para contener los ficheros y directorios que usa. Se crean sobre las particiones que hayamos designado para ese sistema operativo.

En un sistema UNIX la información se guarda en ficheros, los cuales a su vez están organizados dentro de lo que se conoce como Sistemas de Ficheros. Un sistema de ficheros es simplemente una estructura situada en un área del disco llamada partición o sección. Puede haber, por tanto, varios sistemas de ficheros en un mismo disco (en varias particiones o secciones de él). También puede ocurrir, al menos en sistemas UNIX más recientes, que un sistema de ficheros esté montado sobre varias unidades de disco, dando lugar a lo que se llama sistemas de ficheros multivolumen.

En Linux trabajamos básicamente con cuatro tipos de ficheros:

1. Ficheros normales.
2. Directorios.
3. Enlaces.
4. Archivos especiales<sup>1</sup>.

Los dos primeros no presentan ningún problema y del 3º daremos algunas pinceladas en esta sección.



Un par de anotaciones antes de seguir:

- Linux es “casesensitive”, o sea que el fichero `curso.txt` es distinto del fichero `Curso.txt` y lo mismo con los directorios.

<sup>1</sup> Aquí englobaríamos a los ficheros de dispositivos que se encuentran en el subdirectorio `/dev` y otros ficheros especiales: *Fifo* o tuberías con nombre.

- Los nombres de fichero pueden contener los caracteres a los que estamos habituados, con un máximo de 255.

Todos los ficheros dentro del sistema de ficheros se organizan siguiendo una estructura en forma de árbol invertido, en la que el nodo superior se denomina nodo raíz.

La mayoría de los sistemas de ficheros UNIX tienen una estructura parecida, lo que varía de unos a otros son los detalles de la implementación, lo que los hará más o menos eficientes.

## 12.2. Tipos de Sistemas de Ficheros.

Linux, en particular, además de los tipos de sistemas de ficheros nativos (Minix, Ext, Xia, Ext2 y Ext3) soporta varios sistemas de ficheros ajenos, para facilitar el intercambio de datos con otros sistemas operativos. Algunos de los sistemas de ficheros soportados por Linux son<sup>2</sup>:

**ext2** Sistema de archivos estándar de Linux

**ext3** Sistema de ficheros que usa Red Hat 8.0 de forma predeterminada. Es una mejora del anterior con la ventaja de que es de tipo *journaling* (se reduce el tiempo de recuperación tras un apagado inesperado)<sup>3</sup>.

**msdos** Para la compatibilidad con el sistema de ficheros FAT del MS-DOS.

**vfat** Para compatibilidad con sistemas Windows 9x (Fat32). Soporta nombres largos de ficheros.

**iso9660** Es el tipo de sistema de ficheros estándar para CDROM; la extensión Rock Ridge al CD-ROM estándar, que permite nombres de fichero más largos, está soportada de forma automática.

**umsdos** Extiende el sistema de ficheros msdos bajo Linux, de forma que desde Linux se pueden usar nombres de fichero largos, propiedad, permisos, enlaces y ficheros de dispositivo. Esto permite que un sistema de ficheros msdos se use como si fuera un sistema de ficheros Linux, sin necesidad de hacer una nueva partición para Linux. Tiene como contrapartida un rendimiento inferior a los sistemas de ficheros nativos.

**hpfs** Para la compatibilidad con el sistema de ficheros de OS/2.

**nfs** Es un sistema de ficheros de red que permite compartir sistemas de ficheros entre varios ordenadores.

**sysv** Para compatibilidad con UNIX SystemV/386, Coherent y Xenix.

**minix** Primer sistema de ficheros utilizado para Linux

**ext** Primer sucesor de minix (está en desuso).

**xiafs** Sucesor del ext (tampoco se usa)

**proc** Sistema de archivos virtual de Linux.<sup>4</sup>

<sup>2</sup>Si compartimos sistema con Windows 2000/XP, veremos después como conseguir acceso al sistema de ficheros **ntfs**.  
<sup>3</sup>

- **Características:** Máximo tamaño de bloque 4Kb; Tamaño máximo del sistema de ficheros 16.384 GB; Tamaño máximo de fichero 2048 GB.
- Para pasar un sistema de ficheros **ext2** a uno **ext3** y viceversa disponemos del comando `/sbin/tune2fs`. Para saber como hacerlo, además de la *manpage* del programa mirar el capítulo 1 de *The Official Red Hat Linux Customization Guide*.

<sup>4</sup>No es en realidad un sistema de ficheros, aunque lo parece. Permite acceder a ciertas estructuras de datos del Kernel, como, por ejemplo, listas de procesos (de aquí su nombre). Organiza estas estructuras como un sistema de ficheros de modo que se pueda acceder a ellas con las herramientas habituales de acceso a ficheros.

Por ejemplo, para ver los procesos que están ejecutándose en el sistema usaremos el comando:

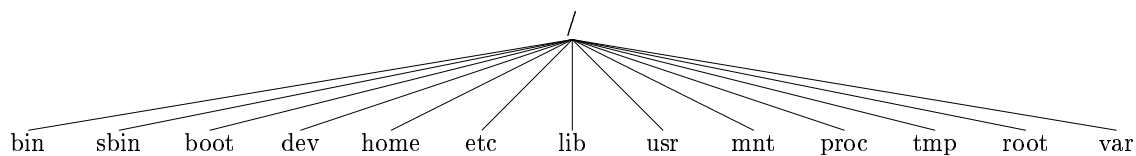
```
$ ls -l /proc
```

Existe además, generalmente, una partición o sección en el disco usada por el gestor de memoria, conocida como área de **swap** (intercambio). A través ella el gestor de memoria implementa la memoria virtual. Dependiendo de los sistemas, este área puede ser tratada bien como un sistema de ficheros más, o bien directamente a través del dispositivo en bruto (*raw disk*), .

## 12.3. Organización de los directorios

El **FHS** (*Filesystem Hierarchy Standar* <http://www.pathname.com/fhs/>) o Estándar de Jerarquía del Sistema de Ficheros, es un documento en el que se definen los nombres y situación de muchos ficheros y directorios.

En Linux, “todo son ficheros<sup>5</sup>”, para organizarlos tenemos una estructura de árbol de la que cuelgan todos los ficheros. El tronco principal es /, el directorio raíz.



De él cuelgan normalmente los directorios<sup>6</sup>:

<b>/bin</b> Contiene los comandos básicos del sistema operativo <sup>7</sup>	<b>/mnt</b> De él cuelgan directorios para montar sistemas de archivos. Por ejemplo, ahí montaremos la unidad de disquette ( <code>/mnt/floppy</code> ) o el CD ( <code>/mnt/cdrom</code> ) <sup>9</sup>
<b>/sbin</b> Comandos esenciales para la administración del sistema <sup>8</sup> .	<b>/proc</b> Ya se ha comentado anteriormente, este directorio no existe en realidad (físicamente). En él podemos encontrar información sobre el sistema.
<b>/boot</b> Aquí están los binarios de arranque del sistema.	<b>/tmp</b> En él se almacena información temporal.
<b>/dev</b> Todos los ficheros de dispositivos.	<b>/usr</b> Aquí estarán los programas que no forman parte del sistema más básico.
<b>/etc</b> Ficheros de configuración del sistema.	<b>/var</b> ficheros de datos variables: logs del sistema, datos administrativos, etc.
<b>/home</b> En él tendremos los directorios de trabajo de los usuarios.	
<b>/root</b> Directorio de trabajo del “jefe”.	
<b>/lib</b> Librerías básicas para trabajar en Linux.	

### 12.3.1. Ficheros de configuración del sistema

Se almacenan en distintas partes del sistema de ficheros de nuestro sistema Linux. Fundamentalmente en:

O para ver los distintos tipos de sistemas de ficheros soportados por el Kernel que tenemos arrancado:

```
$ cat /proc/filesystems
```

Aunque a veces se le llame sistema de ficheros, `/proc` no está montado sobre el disco, sino que sólo existe en el kernel. Cuando intentamos acceder a alguna parte de `/proc`, el Kernel crea la ilusión de que existe en el disco aunque no es así. Si miráis el contenido de los ficheros de ese “directorio” podéis ver qué procesos están en ejecución e información de toda la máquina.

<sup>5</sup>Incluso los directorios son ficheros cuyos datos son los archivos que contienen. Y los dispositivos son ficheros especiales de los que se puede leer y en los que se puede escribir.

<sup>6</sup>Red Hat crea además los directorios: `initrd` (si se borra no podremos acceder al sistema), `misc` y `opt` (como sus nombres indican) y `lost+found` (zona para poner los archivos “extraviados”).

<sup>7</sup>En `/usr/bin` también hay comandos del sistema pero son “menos básicos” que los que hay en `/bin`.

Se debe a que al iniciar el sistema, puede que todavía la estructura de `/usr` no esté disponible porque esté sobre otro sistema de ficheros. Pero `/bin` siempre lo tendremos a mano en caso de que haya problemas.

<sup>8</sup>En `/usr/sbin` y `/usr/local/sbin` también hay comandos de administración del sistema. De igual manera que antes son los comandos de “2ª” para esta labor.

<sup>9</sup>No es obligatorio hacerlo así pero con RedHat es lo usual.

**\$HOME** en el directorio de trabajo de cada usuario hay una serie de ficheros<sup>10</sup> que rigen el comportamiento de sus cuentas de registro.

**/etc** En este directorio están la mayoría de los ficheros de configuración del sistema<sup>11</sup>:

- |   |  |
|---|--|
| <b>alias</b> Puede contener listas de distribución utilizadas por el servicio de correo de Linux.   | <b>lilo.conf</b> parámetros usados por el gestor de inicio ( <b>lilo</b> ) de Linux, para arrancar su sistema.                                   |
| <b>bashrc</b> valores por defecto en todo el sistema para la shell <b>bash</b> .  | <b>mail.rc</b> parámetros del sistema asociados con el uso del correo.   |
| <b>crontab</b> fichero de configuración de cron.  | <b>man.config</b> fichero de configuración del comando <b>man</b> .  |
| <b>csh.cshrc</b> (o <b>cshrc</b> ) en él se establecen los valores por defecto de la shell <b>csh</b> (shell C).  | <b>modules.conf</b> en él se establecen los alias y opciones para los módulos cargables de kernel.   |
| <b>exports</b> lista de los directorios locales para ser compartidos utilizando NFS (Network File System).  | <b>mtab</b> lista de sistemas de archivos montados en ese momento.   |
| <b>fdprm</b> parámetros para formatos comunes de disquete.  | <b>passwd</b> cada línea de este fichero guarda información relativa a cada uno de los usuarios del sistema.                                     |
| <b>fstab</b> contiene información que describe los diversos sistemas de ficheros y las ubicaciones del sistema Linux en las que se montan.  | <b>printcap</b> archivo de configuración de las impresoras configuradas en nuestro equipo.   |
| <b>group</b> en él se almacenan los nombres de grupo e ID de grupo (GID) definidos en el sistema.   | <b>profile</b> fichero que configura el entorno de trabajo, tiene validez para todo el sistema, programas de entorno e inicio de todos usuarios. |
| <b>host.conf</b> fichero de configuración del “resolvidor” de nombres. En él se establecen localizaciones en las que se buscan los nombres de dominio (por ejemplo, redhat.com) en las redes TCP/IP (como por ejemplo, Internet). | <b>protocols</b> en él se describen los distintos protocolos para Internet que están disponibles en el subsistema TCP/IP.                        |
| <b>hosts</b> lista de direcciones IP y nombres de máquinas.   | <b>redhat-release</b> cadena que identifica el número de la versión actual de Red Hat.   |
| <b>hosts.allow</b> ordenadores que están autorizados para utilizar servicios TCP/IP desde el ordenador local.   | <b>resolv.conf</b> direcciones IP de los servidores de nombres.  |
| <b>hosts.deny</b> ordenadores que no están autorizados a utilizar servicios TCP/IP desde el ordenador local.  | <b>rpc</b> define nombres y números para los procedimientos remotos de llamada.  |
| <b>info-dir</b> encabezamiento de la información que está disponible con el comando <b>info</b> .   | <b>services</b> lista de servicios de red de Internet  |
| <b>inittab</b> mediante este fichero podemos ajustar el inicio del sistema para que se ejecute según nuestros intereses   | <b>shadow</b> lista de contraseñas encriptadas para los usuarios que aparecen en el fichero <b>passwd</b> .                                      |
| <b>issue</b> texto de bienvenida a una sesión en Red Hat Linux desde un terminal local o la consola en modo texto.  | <b>shells</b> lista las rutas de los intérpretes de línea de comando con los que contamos en nuestro sistema.                                    |
| <b>issue.net</b> texto de bienvenida de inicio de sesión que se muestran a los usuarios que intentan iniciar una sesión <b>telnet</b> .   | <b>sudoers</b> fichero de configuración del comando <b>sudo</b> <sup>12</sup> .  |
| <b>ld.so.conf</b> lista de directorios que contienen librerías del sistema.   | <b>syslog.conf</b> en él se define qué mensajes de inicio de sesión recoge el demonio <b>syslogd</b> y qué ficheros se almacenan en él.          |
|   | <b>termcap</b> base de datos con características de terminales <sup>13</sup> .   |
|   | <b>xinetd.conf</b> configuración básica que usa el proceso daemon <b>xinetd</b> (servicios de red).  |

<sup>10</sup>La mayoría empiezan su nombre con un punto

<sup>11</sup>Algunos puede que no estén (aún) en vuestro sistema.

<sup>12</sup>Con él podemos conseguir de una forma segura que determinados comandos restringidos al root puedan ser ejecutados por otros usuarios.

<sup>13</sup>Es un sistema obsoleto.

**/etc/X11** directorio de configuración del sistema gráfico X y de los diferentes gestores de ventanas.

**/etc/cron\*** distintos directorios con ficheros en los que se define la forma en que `cron` ejecuta las aplicaciones programadas con pautas diarias, horarias, mensuales o semanales.

**/etc/default** en este subdirectorio hay una serie de ficheros que establecen valores para diversas utilidades. Véase el fichero `useradd`, en él se definen los valores por defecto para el número de grupo, el directorio de inicio, la fecha de caducidad de la contraseña, el shell y el directorio esqueleto (`/etc/skel`) que se utiliza cuando se crea un nuevo usuario del sistema.

**/etc/http** subdirectorio que contiene los ficheros necesarios para configurar el servidor de web Apache.

**/etc/init.d** contiene las copias permanentes de los guiones de nivel de ejecución. Estos guiones están vinculados a ficheros en los directorios `/etc/rc?` para que cada servicio se asocie con un guión iniciado o detenido en el nivel de ejecución en cuestión.

**/etc/ppp** ficheros de configuración de la conexión a internet usando un módem/RDSI.

**/etc/rc?.d** directorios que determinan los procesos a ejecutar en los diferentes niveles de ejecución.

**/etc/security** aquí se establecen una serie de condiciones de seguridad por defecto para nuestro ordenador.

**/etc/skel** directorio “esqueleto” para crear los directorios de usuario de las nuevas cuentas del sistema. Los ficheros de este directorio se copian al directorio de trabajo (`/home/usuario`) del usuario (la mayoría de estos ficheros son ficheros que comienzan por un punto).

**/etc/sysconfig** En este subdirectorio están entre otros los ficheros que controlan:

- `/etc/sysconfig/keyboard` → el teclado
- `/etc/sysconfig/i18n` → establece el idioma predeterminado
- `/etc/sysconfig/clock` → los valores leídos del reloj del sistema.
- `/etc/sysconfig/mouse` → información sobre el ratón
- `/etc/sysconfig/network` → información sobre la red (nombre de la máquina, etc)
- `/etc/sysconfig/soundcard` → la tarjeta de sonido
- `/etc/sysconfig/network-scripts` → ficheros de conexión a la Red

Las entradas de los ficheros del subdirectorio `/etc/sysconfig` no son complicadas de entender, para conocer qué se puede hacer con cada uno de ellos os remitimos al fichero `/usr/share/doc/initscripts-6` y al capítulo 3 de *The Official Red Hat Linux Reference Guide*.



Si tenemos problemas con el modo texto (acentos, eñes). Se puede solucionar modificando los ficheros como sigue:

```
$ cat /etc/sysconfig/i18n
LANG="es_ES@euro"
SUPPORTED="es_ES@euro:es_ES:es"
SYSFONT="lat0-sun16"
SYSFONTACM="iso15"

$ cat /etc/sysconfig/keyboard
KEYTABLE="es-cp850"
```

**/etc/xinetd.d** ficheros que permiten configurar un servicio de red<sup>14</sup>

---

<sup>14</sup>Los veremos después

### 12.3.2. Logs del sistema

Si en algo sobresale Linux es en la posibilidad de mantener multitud de ficheros en los que se almacenan todas las acciones que realiza el sistema (ficheros de registro). Esto se consigue gracias a los demonios:

**syslogd** monitoriza el registro general del sistema

**klogd** registro específico de la actividad del kernel

El fichero de configuración de los logs del sistema se lleva a cabo a partir del fichero `/etc/syslog.conf`, y los ficheros en dónde se almacenan los resultados están en: `/var/log`. Algunos de ellos son:

**cron** mensajes del comando `cron`<sup>15</sup>.

**boot.log** mensajes del arranque.

**secure** almacena los intentos de conexión, tanto locales como remotas, nos puede servir para indicar fallos de seguridad.

**maillog** mensajes generados por el demonio `sendmail`<sup>16</sup>.

**messages:** “cajón de sastre” de los mensajes del sistema.

**xferlog** para controlar el servidor de ftp

**XFree86** guarda información sobre lo que pasa con el entorno gráfico (servidor X).

...


Algunos subdirectorios específicos de logs:

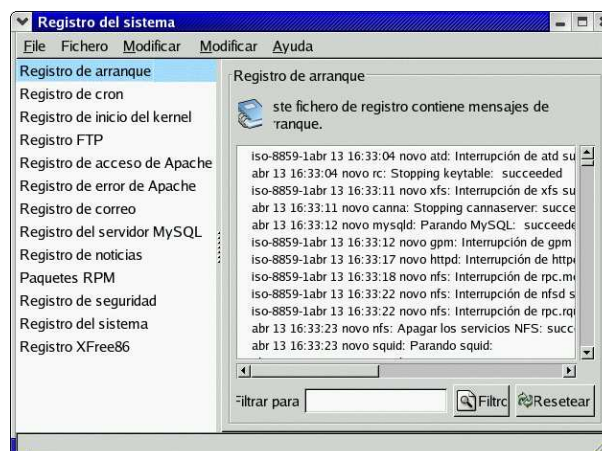
**/var/log/httpd** mensajes procedentes del servidor Web Apache.

**/var/log/samba** mensajes de la utilidad samba (permite interconectar equipos Windows con Linux).

**/var/log/squid** mensajes generados por el proxy/cache `squid` (¿queréis saber qué páginas visita cada equipo del instituto?).

...

Con `redhat-logviewer` (  → **Herramientas del sistema** → **Registro del sistema**) podemos acceder a los logs del sistema en modo gráfico:



<sup>15</sup>Cron permite planificar tareas en el tiempo

<sup>16</sup>MTA: agente para el transporte de correo electrónico



**➡ Para practicar**

1. Modificar el fichero de bienvenida para que el texto sea más amigable.
2. Mostrar por pantalla las últimas 10 líneas del fichero `/var/log/secure`:  

```
#tail -f /var/log/secure
```
3. Visualizar el fichero `/var/log/messages`
  - a) Usando un editor de texto (`gedit` por ejemplo)
  - b) Usando una tubería (`|`) y algo de comandos:  

```
$dmesg | less
```

para salir hay que pulsar `q`
  - c) Con `redhat-logviewer`

## 12.4. Creación de un Sistema de Ficheros.

Una vez instalada Red Hat, disponemos de varias utilidades para poder trabajar sobre las particiones de un disco duro. Si lo que deseamos es ver la tabla de particiones de un disco, añadir, borrar o cambiar el tipo de sistema de ficheros podemos usar `fdisk` o `sfdisk`. Pero si además de todo eso deseamos modificar el tamaño de una partición tendremos que usar `parted`. La sintaxis básica de uso en los tres es similar y consiste en ejecutar el programa (como `root`) pasándole como argumento la unidad de disco con la que se va a trabajar (`/dev/hdx`).

Para saber más sobre su uso, además de la documentación instalada, se puede ver

- Para `fdisk`: Capítulo 5 de *Red Hat Linux 7.3: Manual oficial de referencia de Red Hat Linux*  
<http://www.europe.redhat.com/documentation/rhl7.3/rhl-rg-es-7.3/>
- Sobre `parted`: Capítulo 5 de *The Official Red Hat Linux Customization Guide*

Una vez creada la partición disponemos de:

**mkfs** Los sistemas de ficheros se crean con el comando `mkfs` (Make Filesystem), `mkfs` nos permite construir un sistema de ficheros sobre un dispositivo, normalmente una partición o sección de un disco. La sintaxis<sup>17</sup> de este comando es:

```
mkfs [-t tipo_sf] sistema ficheros
```

Donde:

*tipo\_sf* es el argumento mediante el que se pasa el tipo de sistema de ficheros a crear (`ext3`, `ext2`, `hfs`, `Minix`, etc). Si se omite este argumento, `mkfs` lo deducirá buscando en `/etc/fstab`. Si se omite y no hay entrada en `fstab` tomará por defecto **ext2**.

*sistema ficheros* es el único argumento obligatorio, y corresponde a la partición o sección del disco sobre la cual queremos crear el sistema de ficheros (`/dev/hda1`, `/dev/sda3`, etc), o también podría pasársele el punto de montaje<sup>18</sup> (`/tmp`, `/users`, etc).

Por ejemplo, la sentencia,

```
# mkfs -t ext3 /dev/sda1
```

<sup>17</sup>No es la sintaxis “completa”, para ampliar sobre este comando mirar en las páginas man del comando.

Por si no lo hemos comentado antes, la sintaxis de los comandos se expresa como una expresión regular: opciones o argumentos entre corchetes significa que pueden o no aparecer (`[]`), la barra vertical (`|`) significa que puede aparecer un elemento u otro pero uno de ellos obligatoriamente ...

<sup>18</sup>En este caso, intentará encontrar la partición en la que crear el sistema de ficheros de la entrada en el fichero `/etc/fstab` donde encuentre el punto de montaje.

nos creará un sistema de ficheros tipo ext3, que ocupará todo el espacio de la partición `/dev/sda1` del disco<sup>19</sup>.

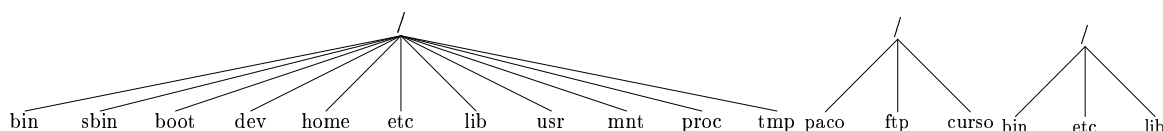
### ➤ Para practicar

Si deseamos crear un sistema de ficheros de tipo ext2 en un disquete (`/dev/fd0`), escribiremos:

```
#mkfs -t ext2 /dev/fd0
```

## 12.5. Montaje y Desmontaje.

Para poder usar un sistema de ficheros tiene que estar *montado*. El concepto de montaje tiene que ver con que en Unix/Linux todos los elementos son ficheros. Para ello, cada nuevo sistema de ficheros se enlaza mediante la operación de montaje con otra estructura de directorios de la que “cuelga”. El sistema de ficheros inicial se denomina el *sistema de ficheros raíz* y posee el símbolo `/`. Este sistema de ficheros se monta al arrancar el sistema y ocupa el lugar más alto<sup>20</sup>. Por ejemplo, supongamos que en tres particiones de nuestro disco (o discos diferentes) hemos creado tres sistemas de ficheros. Podemos verlos de la siguiente forma. Cada uno de ellos visto de forma independiente posee una raíz. Supongamos que son `/dev/hda1`, `/dev/hda2` y `/dev/hda3` respectivamente:



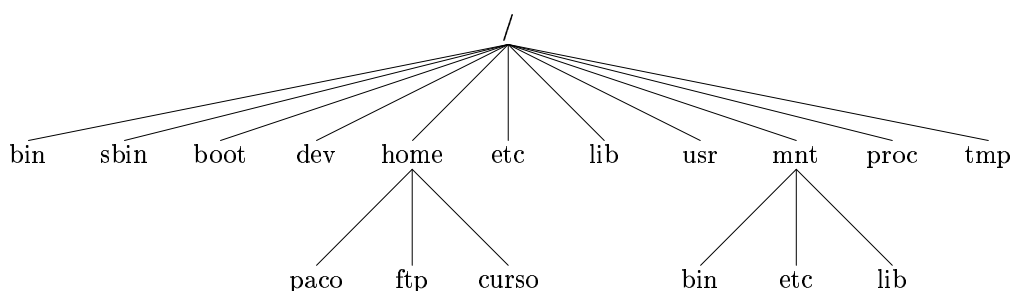
Cuando montamos un sistema de ficheros, indicamos un directorio del sistema de ficheros “padre”, del cual va a colgar la estructura de directorios del sistema de ficheros “hijo”.

Mediante las órdenes de montaje<sup>21</sup> siguientes:

```
# mount /dev/hda2 /home
```

```
# mount /dev/hda3 /mnt
```

obtenemos la siguiente estructura de directorios.



Como vemos, hemos montado el sistema de ficheros que hay en `/dev/hda2` bajo el directorio `/home` del sistema de ficheros raíz. Y el sistema de ficheros de `/dev/hda3` bajo el directorio `/mnt`.

<sup>19</sup>Con `e2label` podemos asignarle una etiqueta a la partición, la ventaja de usar este método es que desde este momento podemos usar la etiqueta y no la ruta del dispositivo para trabajar con él.

<sup>20</sup>En él se encuentra el “corazón” de nuestro sistema.

<sup>21</sup>La sintaxis la estudiaremos un poco más adelante.

### 12.5.1. El fichero /etc/fstab

El fichero `/etc/fstab` contiene información descriptiva sobre los distintos sistemas de ficheros del sistema. Este fichero es de solo lectura y debe ser mantenido por el administrador del sistema. Cada sistema de ficheros ocupa una línea de este fichero, y los campos de cada línea están separados por tabuladores o espacios. El orden de los registros es importante ya que `fsck`, `mount` y `umount` actúan secuencialmente sobre este fichero.

**device** Es el primer campo y especifica el dispositivo de bloque o el sistema de ficheros remoto a montar.<sup>22</sup>

**directorio** Es el segundo campo y especifica el punto de montaje para el sistema de ficheros. Para particiones de swap este campo debería estar a "none". El directorio destino de montaje tiene que existir en nuestro sistema antes de montar el sistema de ficheros.<sup>23</sup>

**tipo** Es el tercer campo y especifica el tipo de sistema de ficheros. Si el contenido de este campo es "ignore", el sistema de ficheros no se monta. Esto puede ser útil para mostrarnos las particiones de disco que están actualmente sin ser usadas.

**opciones** Es el cuarto campo y especifica las opciones de montaje asociadas al sistema de ficheros. Las opciones van separadas por comas. Algunas de las opciones posibles son:

`auto` → La partición se monta al arrancar  
`noauto` → No se monta la partición en el arranque  
`user` → Se permite a los usuarios montar la partición  
`nouser` → Sólo el root puede montar esta partición.  
`ro` → Partición de sólo lectura.  
`rw` → Se permite la lectura y la escritura.  
`exec` → Se pueden ejecutar los binarios de esa partición.  
`async` → El sistema sigue trabajando tras una petición de escritura del dispositivo, aunque todavía no haya recibido la confirmación.  
`defaults` → equivale a: `rw, exec, auto, nouser, async`.

**frecuencia** Con él determinamos la frecuencia con que deben de hacerse copias de seguridad del sistema por el comando `dump`. Si este campo no está presente se devuelve a `dump` el valor cero, lo que indica que el sistema de ficheros no necesita ser salvado.

**secuencia** Es el sexto campo (`fsckorder`) y es usado por `fsck` (se estudia en 12.6 en la página 178) para determinar el orden en que se realizan los chequeos de los sistemas de ficheros en tiempo de arranque. El sistema de ficheros raíz debería ser especificado con un 1, y los demás sistemas de ficheros deberían tener 2. Los sistemas de ficheros en un mismo disco deberían chequearse secuencialmente, pero los sistemas de ficheros en diferentes discos deberían de chequearse al mismo tiempo, para utilizar el paralelismo disponible en el hardware. Si el sexto campo no está presente o es cero, se devuelve el valor cero, y `fsck` asume que el sistema de ficheros no necesita ser chequeado.

Ejemplo de fichero `/etc/fstab`<sup>24</sup>:

<device>	<directorio>	<Tipo>	<Opciones>	<dump>	<fsckorder>
/dev/hdc1	/	ext3	defaults	1	1
/dev/hdc6	swap	swap	defaults	0	0
/dev/fd0	/mnt/floppy	auto	noauto,user	0	0

<sup>22</sup>Desde la versión 7 de RedHat se han incorporado etiquetas para los distintos sistemas de archivos. Si hemos actualizado nuestro sistema no aparecerán, pero si es una instalación nueva sí. La diferencia reside en que si hemos instalado por primera vez linux en nuestra máquina en vez de aparecer en el campo device el sistema que vamos a montar veremos la etiqueta que referencia ese sistema. Como ya hemos comentado, el comando que permite mostrar/cambiar las etiquetas del sistema de archivos es `e2label`.

<sup>23</sup>La orden para crear un subdirectorio es `mkdir`

<sup>24</sup>La primera línea de ese mismo fichero usando etiquetas sería de la forma:

`LABEL=/ / ext3 defaults 1 1`

/dev/cdrom	/mnt/cdrom	iso9660	noauto,ro,user	0	0
/dev/hda1	/mnt/windows	vfat	noauto		
/dev/hda2	/mnt/win2	ntfs	auto		
none	/proc	proc	defaults	0	0
none	/dev/pts	devpts	gid=5,mode=620	0	0

### 12.5.2. El comando mount

El comando `mount` nos permite montar los sistemas de ficheros y anclarlos a puntos de montaje (directorios). A partir de este momento podremos acceder a ellos. La sintaxis más usual es:

```
mount [-t fstype] [device] dir
```

Si montamos una unidad que está incluida en el fichero `/etc/fstab` se omiten los argumentos `[-t fstype]` y `[device]`, la información necesaria la obtiene el comando `mount` del fichero `/etc/fstab`. Por ejemplo, con el `/etc/fstab` anterior, para montar la unidad de CD escribiríamos

```
$ mount /mnt/cdrom25
```

Si lo que queremos es montar un disquete formateado bajo Windows tendremos que escribir

```
$ mount /mnt/floppy
```

ya que tal cual está el fichero `/etc/fstab`, el sistema de forma automática detectará el tipo, así si ese mismo disco es del tipo `ext2` usaremos el mismo comando.

El comando `mount` soporta una gran variedad de opciones, muchas de ellas dependientes del tipo de sistema de ficheros a montar. Para una descripción detallada consultar las páginas del manual en línea :

```
man 8 mount
```

### 12.5.3. El comando umount

La sintaxis más usual es:

```
umount [opciones] dir
```

El comando `umount` elimina el anclaje entre el dispositivo especial y el punto de montaje.

Para desmontar la unidad de CD:

```
$ umount /mnt/cdrom
```

Si tenemos una unidad de CD montada no podremos extraer el CD hasta que no lo desmontemos. Si estamos trabajando con una unidad de disquete es fundamental desmontarla antes de extraer el disco

Si damos la opción `-a`:

```
# umount -a
```

serán desmontados todos los sistemas de ficheros contenidos en `/etc/mtab`. Con la opción `-t` le podemos decir a `umount` que desmonte solo los sistemas de ficheros de un tipo en concreto (o varios, separando los argumentos por comas).

Para desmontar un sistema de ficheros debemos de estar situados fuera de él. Es decir, el directorio actual de la sesión desde la cual realizamos el desmontaje no debe de estar dentro del sistema de ficheros a desmontar. Es más, ningún proceso del sistema debe estar utilizando el sistema de ficheros que deseamos desmontar, en caso contrario nos diría que está ocupado. La solución será “matar” al proceso o procesos que están utilizando el sistema de ficheros que deseamos desmontar o esperar a que acaben “por las buenas”.

Para saber qué procesos están utilizando un sistema de ficheros podemos utilizar la orden `fuser`. Por ejemplo, si en `/mnt/windows` tenemos montada nuestra partición con el sistema de ficheros **vfat** y no nos deja desmontarla, podemos ver de qué procesos se trata con:

```
$ /sbin/fuser -a /mnt/windows
```

### 12.5.4. Herramientas gráficas para montar dispositivos

Si trabajamos con GNOME, el montaje y desmontaje de unidades de CD se puede realizar de forma automática o pulsando con el ratón sobre el icono correspondiente y seleccionando la opción adecuada.

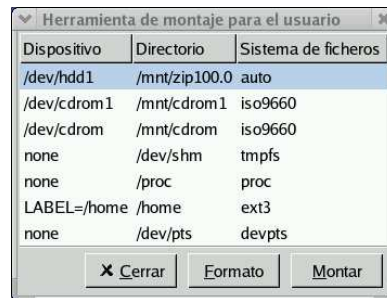
<sup>25</sup>Notad que la orden la puede ejecutar un usuario al tener como opción `user` el dispositivo `/dev/cdrom`


### Usando usermount

Si desde una xterm ejecutamos como root

```
# usermount
```

se nos abrirá una ventana parecida (el número de entradas dependerá del usuario que lo ejecute, la captura se ha realizado como root)



Este programa es una interfaz gráfica para montar, desmontar y formatear sistemas. En GNOME se encuentra en  → **Herramientas del sistema** → **Gestión de disco**.

### ➡ Para practicar

1.
  - a) Crear en un disquette un sistema de ficheros de tipo **vfat**.
  - b) Montar el disquette y copiar en él el fichero `/etc/hosts`
  - c) Desmontar el disquete.
  - d) Comprobar que todo ha salido bien (habrá que montar el disquete de nuevo y visualizar su contenido)
2. Montar la partición Windows (partimos de la idea de que es el dispositivo `/dev/hda1`, si no se tiene o es otra partición hay que ajustarlo)
  - a) Crear un directorio destino de montaje:

```
#mkdir /mnt/windows
```
  - b) Montarla:

```
#mount -t vfat /dev/hda1 /mnt/windows
```
3. Modificar el fichero `/etc/fstab` para que:
  - a) permita montar el cdrom a cualquier usuario del sistema
  - b) monte la partición Windows en el arranque

### 12.5.5. ➡ Para practicar: Montar una partición ntfs

Por defecto, Red Hat no permite montar sistemas de ficheros **ntfs** (Windows 2000/XP). Vamos a guiar la forma de conseguirlo.



Para conocer más sobre las posibilidades de este módulo mirar en The Linux-NTFS Project <http://linux-ntfs.sourceforge.net/>. En esa página se comenta que, aunque no debe haber problemas si realizamos procesos de lectura en los sistemas de ficheros **ntfs**, es extremadamente peligroso escribir sobre ellos. Habrá que esperar un poco para poder disponer de esta funcionalidad sin peligro.

1. Necesitamos instalar el paquete `kernel-ntfs` adecuado a nuestro núcleo y microprocesador.
  - a) Para saber con exactitud cuáles son podemos ejecutar:  
`$uname -a`
  - b) Bajar, a partir de esa información, el paquete (en formato RPM) para nuestra plataforma de <http://linux-ntfs.sourceforge.net/info/redhat.html>.
  - c) Instalarlo:  
`#rpm -ivh kernel-ntfs-*.rpm`
2. Cargar el módulo con<sup>26</sup>:  
`#/sbin/modprobe ntfs`
  - a) Podemos comprobar que disponemos de un nuevo tipo de sistema de ficheros para poder cargar con:  
`$cat /proc/filesystems`
3. Crear el punto de montaje. Por ejemplo con:  
`#mkdir /mnt/win2`
4. Listo, si disponemos de un sistema **ntfs** ya podemos montarlo ejecutando:  
`#mount -t ntfs punto_montaje destino_montaje`  
Por ejemplo, si el sistema **ntfs** está en `/dev/hda2` escribiríamos:  
`#mount -t ntfs /dev/hda2 /mnt/win2`
5. Si lo que deseamos es que el sistema se monte en el arranque hay que añadir una línea similar (adecuándola a nuestro equipo) a la línea 6 del fichero de ejemplo `/etc/inittab` de la página 174.

### 12.5.6. ➡ Para practicar: Paquete mtools

El paquete `mtools` incluido en RedHat y que se instala por defecto, trae los comandos: `mcopy`, `mdir`, ... similares a los de MS-DOS. Por ejemplo:

```
mcopy a:* /home/thales
```

copia el contenido del floppy en el subdirectorio indicado. Es interesante resaltar que **para usarlos no es necesario montar el floppy**.

El fichero de configuración de este paquete es `/etc/mtools.conf`. En general no hay que modificarlo nunca pero si algo no funciona bien puede que tengamos que ajustarlo a nuestro sistema.

1.
  - a) Comprobar qué comandos componen el paquete usando  
`$info mtools`
  - b) Formatear un disquete con la orden  
`$mformat a:`
  - c) Listar el contenido del disquete con `mdir`  
`$mdir a:`
  - d) Copiar el fichero<sup>27</sup> `~/ .bashrc` al disquete usando las `mtools`:  
`$mcopy ~/ .bashrc a:`
  - e) Montar el disquette creado con la orden `mount` y comprobar que la copia se ha realizado bien.

<sup>26</sup>Sólo se hace la primera vez y no es necesario repetirlo más

<sup>27</sup>El caracter `~` referencia el home de ese usuario.

- f) El sistema mantiene una lista de los sistemas de ficheros montados actualmente, en el fichero `/etc/mstab`. Se puede ver el contenido del fichero utilizando el comando `mount` sin argumentos:

```
$ mount
```

## 12.6. Chequeo y recuperación: fsck.

El comando `fsck` nos permite chequear y, opcionalmente, reparar un sistema de ficheros (`ext3` o `ext2`). `fsck` repara inconsistencias en los sistemas de ficheros después de un apagado incorrecto de la máquina.

Si trabajamos con el sistema de ficheros `ext3`<sup>28</sup>, en general, no surgen problemas por apagados incorrectos del ordenador y sólo es necesario chequear el sistema por problemas hardware del disco. Además, en caso de tener que recuperar un sistema de ficheros `ext3` no hay que chequear todo el sistema y el proceso es mucho más rápido (un par de segundos como máximo).

Si nuestro sistema de ficheros es de tipo `ext2`, el funcionamiento es el siguiente. Cuando un sistema de ficheros se monta, se marca como “sucio” porque el sistema en su trabajo normal, tendrá datos mantenidos en memoria en vez de bajarlos directamente al disco, con el fin de mejorar las prestaciones. Si el sistema lo apagamos correctamente, una de sus tareas es descargar todo lo que se encuentre en la memoria y que corresponda a los discos a su lugar correspondiente. Tras hacer esto, se puede marcar el sistema de ficheros como “limpio”. Si por cualquier razón la máquina se apagase<sup>29</sup> sin que se pueda realizar el proceso correcto de apagado, los discos se quedarán como “sucios” y deberán comprobarse al arrancar la máquina. Para chequear un sistema de ficheros éste debe estar desmontado.

Su sintaxis básica es:

```
fsck [-t fstype] [-fsopcion] device
```

en donde `device` es el fichero especial correspondiente al dispositivo<sup>30</sup>, por ejemplo,

```
# /sbin/fsck /dev/hda2
verificaría el volumen /dev/hda2.
```

Podemos indicarle al programa que repare los ficheros sin confirmación por nuestra parte con el parámetro `-a`, por ejemplo:

```
# /sbin/fsck -a /dev/hda2
```

Los ficheros perdidos que recupere los podemos encontrar en el subdirectorio `/lost+found` del dispositivo `/dev/hda2`.

Es interesante comentar la opción

```
# /sbin/fsck -A
```

con esta orden `fsck` recorre el fichero `/etc/fstab` y verifica todas las unidades en función del parámetro `fsckorder` del fichero `/etc/fstab`. Pero no se debería usar con sistemas de ficheros ya montados.

Para información sobre las opciones consultar el manual en línea.

### ➡ Para practicar

1. Chequear el disquete antes creado ¿qué problema surge?<sup>31</sup>
2. Chequear la partición `/boot`:

- a) Si no sabemos cual es podemos usar:

```
$mount
```

o

```
#/sbin/fdisk -l /dev/hda
```

<sup>28</sup>Para saber más sobre el sistema `ext3`: <http://linuxmobile.sourceforge.net/recursos/documentacion.html>

<sup>29</sup>Por ejemplo un apagón de luz.

<sup>30</sup>Es el único parámetro obligatorio

<sup>31</sup>Si nos dice que no detecta el tipo es que habrá que decírselo con `-t vfat`

3. Ya es sencillo, usar `fsck` tal cual se ha explicado.

## 12.7. Enlaces

**Enlaces simbólicos** Son ficheros que únicamente contienen el nombre de otro fichero<sup>32</sup>. Como un enlace simbólico apunta a un fichero (con su camino completo), es posible establecer enlaces simbólicos entre distintos sistemas de ficheros, y entre cualquier tipo de fichero, incluso con un fichero que no exista.

Se crean con la orden `ln` seguidos de la opción `-s`, por ejemplo:

```
#ln -s /usr/X11R6/bin/XF86_FBDev /etc/X11/X
```

Con esta orden lo que hacemos es crear un fichero nuevo (`/etc/X11/X`) que apunta al fichero `/usr/X11R6/bin/XF86_FBDev`.

Podemos usar dos herramientas para crear o modificar enlaces simbólicos:

- Programa **mc**.
- Gestores de ficheros Nautilus y Konqueror (gestor de ficheros del KDE: si no se dispone de él, en 14.5.1 en la página 204 se explica cómo instalarlo).

### ➡ Para practicar

1. Crear un enlace simbólico desde nuestro directorio de usuario al directorio `/var/log`:

```
$ln -s /var/log/ logs
```

- a) Comprobar los efectos de ese enlace.
- b) Borrar el enlace y crear uno para acceder al directorio `/tmp`

**Enlaces hard** Un enlace duro es un nombre adicional para un fichero ya existente. Esto es posible porque en Linux un fichero puede aparecer en varios sitios simultáneamente y sin embargo ocupar un único espacio en el disco. No se pueden realizar enlaces hard entre ficheros de dos sistemas de ficheros distintos. Para crear enlaces duros se usa también la orden `ln`. Para conocer más sobre este tema así como sobre inodos podéis consultar la página man de la orden `ln` o el libro *Linux Instalación y Primeros Pasos*, de MATT WELSH.

---

<sup>32</sup>En cierto sentido se parecen a los “accesos directos” de Windows.



## Capítulo 13

# Permisos. Gestión de Usuarios

Complots apocalípticos futurísticos aparte, sin los usuarios, los ordenadores no tienen objeto alguno.  
(*Red Hat Linux 7*, CHRISTOPHER NEGUS)

### 13.1. Introducción.

Linux es un sistema operativo multitarea y multiusuario, y, por tanto, permite que varios usuarios ejecuten distintas tareas a la vez. En un sistema Linux, aunque lo use una sola persona, al menos deberíamos tener dos cuentas: una para el root que se debe de usar sólo cuando vayamos a cambiar la configuración del sistema y otra como usuario de “a pie” para la realización del trabajo cotidiano. La forma de identificar a cada usuario ante el sistema se realiza mediante la asignación de cuentas de usuario. Cada usuario dispone de un nombre de usuario que lo identifica. Además cada usuario puede pertenecer a uno o varios grupos. La identidad de cada usuario junto con el grupo al que pertenece determina los derechos de acceso a ficheros y otros recursos del sistema.

### 13.2. Permisos de Acceso a los distintos Objetos.

En UNIX cada fichero tiene un propietario, conocido por el sistema por su UID (*User ID* o identificador de usuario), y cada usuario pertenece a un grupo de usuarios (al menos), conocido por el sistema por su GID (*Group ID* o identificador de grupo). Basado en esta estructura, el sistema asigna permisos a los distintos objetos del sistema de ficheros a tres niveles:

1. a nivel del propietario,
2. a nivel del grupo al que pertenece el usuario, y
3. a nivel de todos los demás usuarios.

Para cada uno de estos tres niveles, asigna tres tipos de permisos básicos:

**r** de lectura

**w** de escritura, y

**x** de ejecución.

Esta información está guardada en el inodo<sup>1</sup> del fichero. Se utilizan los bits 0-8 para los 9 permisos. Estos 9 permisos, así como el tipo de objeto del sistema de ficheros, pueden ser visualizados con la opción `-l` (formato largo) del comando `ls`. Supongamos que la salida de la ejecución de la sentencia:

---

<sup>1</sup>Linux asigna a cada archivo un único número llamado inodo. Cuando formateamos un disco se crea la tabla de inodos. En el inodo de un archivo se almacena toda la información referente a ese archivo (propietario, permisos, tamaño...)



```
$ ls -l /home/cursolinux/ed03linux
```

es

```
-rwxr-xr-- cursolin thales 386 Nov 17 22:37 /home/cursolinux/ed03linux
```

analicemos la salida: `-rwxr-xr--`

para eso la vamos a dividir en cuatro bloques: `-`      **`rwx`**      **`r-x`**      **`r--`**

- El primer guión por la izquierda (`-`) nos indica que `ed03linux` es un fichero normal, si tuviera una **`d`** indicaría un directorio.
- `l`** enlace simbólico
- `c`** dispositivo de caracteres
- `b`** dispositivo de bloques
- `p`** canalización con nombre
- El siguiente grupo (**`rwx`**) que este fichero tiene permisos de lectura, escritura y ejecución para el propietario que en este caso es `cursolin`,
- El grupo (**`r-x`**) que el fichero tiene permisos de lectura y ejecución para el grupo que es `thales`, y
- El último grupo (**`r- -`**) que el resto de usuarios tan sólo podrán leer el fichero pero no modificarlo ni ejecutarlo.

El significado de los permisos **`rwx`**, cuando es un directorio, es el siguiente:

- `r`** Permite leer el contenido del directorio, es decir los números de los inodos y los nombres de los ficheros (por ejemplo, el usuario no podría ejecutar el comando `ls -l`, sino solamente `ls`).
- `w`** Permite escribir en el directorio, es decir crear y suprimir ficheros, otros subdirectorios, etc.
- `x`** Permite recorrer el directorio (podría hacer un `cd` a él-meterse dentro del directorio), y utilizar la información de los objetos del directorio, es decir, acceder a los inodos (se podría ejecutar `ls -l`).



El propietario y grupo de un fichero lo podemos modificar con el comando `chown` (*change owner*-cambiar propietario), con `chgrp` (*change group*-cambiar grupo) podemos modificar sólo el grupo, y los permisos los podemos modificar con el comando `chmod` (*change mode*-cambiar modo), si somos el superusuario o el propietario del objeto. La sintaxis básica de las dos primeras no presenta problemas es:

```
chown usuario fichero
```

```
chgrp grupo fichero
```

para cambiar el usuario o el grupo a un fichero. Después veremos cómo hacerlo con programas que nos facilitan el trabajo.

### 13.2.1. chmod

La sintaxis básica es<sup>2</sup>:

```
chmod modo fichero
```

Al comando `chmod` se le pasa como primer argumento los permisos que vamos a asignar al fichero que pasamos como segundo argumento. El primer argumento admite dos tipos de sintaxis: con notación octal o

<sup>2</sup>Existe un parámetro opcional interesante que es `-R` (recursivamente), permite cambiar todos los permisos de los ficheros de un directorio y de todos sus subdirectorios (se puede usar también con `chown` y `chgrp`). Si queréis usarlo mirad cómo se hace en la documentación en línea.



con notación nemónica. Para la notación octal del primer argumento del comando `chmod`, el primer dígito corresponde con el propietario, el segundo con el grupo, y el tercero con todos los demás. Cada dígito octal corresponde con tres dígitos binarios (octal codificado en binario): el primero para la lectura, el segundo para la escritura y el tercero para la ejecución. Si el dígito está a 1 el permiso está habilitado.

Octal	Binario	Permisos
0	000	(ninguno)
1	001	- - x
2	010	- w -
3	011	- w x
4	100	r - -
5	101	r - x
6	110	r w -
7	111	r w x

Por ejemplo, para colocarle al fichero `curlinux` los permisos: `-rwxr-x--x` utilizando la notación de la tabla, introduciríamos:

```
$ chmod 751 curlinux
```

### umask

Por último, comentar que los permisos con los que se crean por defecto los objetos en los sistemas de ficheros, pueden cambiarse con la orden `umask`. `umask` enmascara los permisos en función del valor con que

- cree el programa en cuestión el objeto
- el valor de `umask`


Su valor se define en `/etc/profile`.

**Ejemplo:** si un programa crea ficheros con permisos 666 (`rw-rw-rw-`) y la `umask` es de 002 el fichero finalmente tendría de permisos `666-002=664`, es decir, `rw-rw-r--`. Si embargo, si para otro usuario la `umask` es de 022, el mismo programa crearía los ficheros con permisos `666-022=644`, o sea `rw-r--r--`

Para conocer mejor esta orden os remitimos al manual en línea.

### 13.2.2. Más fácil todavía

Por ahora, además de lo ya comentado podemos usar dos herramientas para modificar fácilmente los permisos de los ficheros:

- Programa **Nautilus** () . Vamos a ver la forma de cambiar los permisos al fichero `~/Xclients-Default`<sup>3</sup>. Si pulsamos sobre él con la tecla derecha del ratón aparecerá el menú de opciones:

<sup>3</sup>Ojo que el fichero es un fichero “punto”, o sea que hay que decirle a ambos gestores de ficheros que nos los muestren.

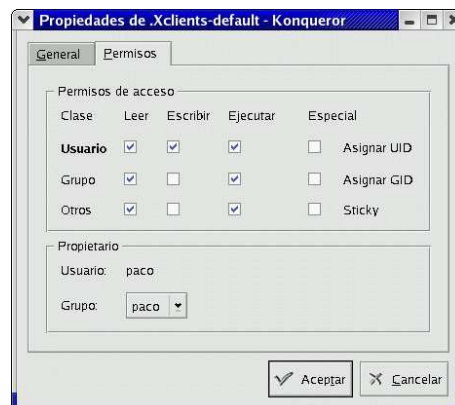


la opción **Propiedades** de este menú nos permite acceder a la ventana

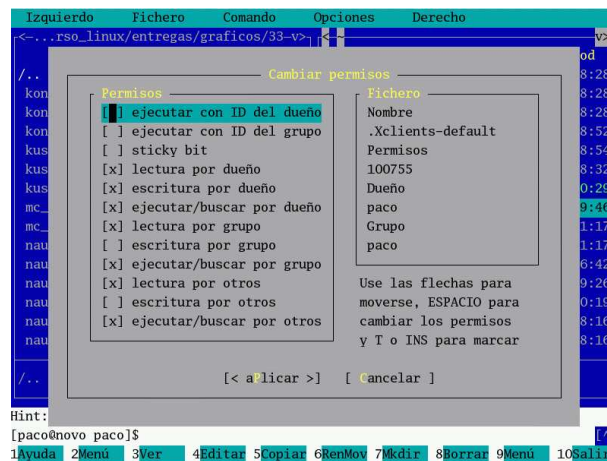


cambiar los permisos, dueño y grupo con esta ventana es ya un “juego de niños”. En este caso, el fichero en cuestión tendría de permisos: `-rwxr-xr-x`. Si modificamos algún permiso veremos como, automáticamente, cambia el número octal correspondiente al modo actual.

- Sólo si tenemos instalado el KDE: Programa **Konqueror**. De nuevo, con el archivo seleccionado, pulsemos sobre el botón derecho del ratón y después en Propiedades



- Programa **mc** (ya visto en la 1ª entrega).



### 13.2.3. Más sobre permisos

En la captura gráfica del programa *mc* y de *konqueror* se puede comprobar que aparecen tres campos más de los que no hemos hablado aún, se trata de los bits 9, 10 y 11, que corresponden con el *sticky bit*, el *SGID* (*Set Group ID*), y el *SUID* (*Set User ID*) respectivamente. Analicemos qué significado tienen:

- El *sticky bit* dice al sistema, que el fichero que lo tiene activo tiene tendencia a ser ejecutado frecuentemente, y debería ser retenido en el área de swap aún cuando no está siendo ejecutado. Esto consume espacio de swap, pero reduce notablemente el tiempo de ejecución. Si el objeto del sistema de ficheros con el *sticky bit* activado es un directorio, no se permite a un usuario borrar ficheros de ese directorio, salvo que los permisos de los ficheros lo permitan. Por último este bit activo nos lo muestra la salida del comando `ls -l` con una *t* en la posición que correspondería al permiso de ejecución para el resto de usuarios<sup>4</sup>. Se puede utilizar para poner un directorio público, donde varios usuarios puedan escribir, pero no borrarse o modificar ficheros simultáneamente.

El *Sticky bit* corresponde con el valor octal 1000, de modo que si quisiéramos activarlo, utilizaríamos la sintaxis:

```
$ chmod 1000 nombre_objeto
```

La sintaxis anterior activaría el modo *sticky bit* a *nombre\_objeto*, pero eliminaría todos los demás si los tuviera. Por tanto, si los permisos de *nombre\_objeto* fueran: `-rwxr-x-` para mantener esos permisos y además activar el *sticky bit*, introduciríamos<sup>5</sup>:

```
$ chmod 1750 nombre_objeto
```

- El *SUID*<sup>6</sup> le indica al kernel, que el usuario que ejecute el fichero que tiene este bit activo, tome la personalidad, durante la ejecución, del usuario propietario del fichero. De esta forma, con ficheros *SUID* root, se soluciona, por ejemplo, el problema de escribir en el fichero `/etc/passwd` para cualquier usuario, ya que cogería momentáneamente la personalidad del root, al ejecutar el comando `passwd` que es *SUID* root. Este bit activo nos lo muestra la salida del comando `ls -l` con una *s* en la posición que correspondería al permiso de ejecución para el propietario<sup>7</sup>. Por ejemplo, podemos comprobarlo con

```
$ ls -l /usr/bin/passwd
```

- El *SGID* tiene un significado parecido, pero referido al grupo de usuarios al que pertenece el fichero. Así, cuando ejecutamos un programa que tiene activo este bit, nuestro GID toma el valor del GID

<sup>4</sup>Será *t* o *T* dependiendo de que el bit esté en 1 ó 0 respectivamente.

<sup>5</sup>La notación nemónica de `chmod` nos permitiría activar el *sticky bit*, o cualquier otro modo, sin tener que conocer cuáles son los modos actuales del fichero.

<sup>6</sup>Los modos con valores octales 4000 y 2000 son para activar el *SUID* y el *SGID* respectivamente.

<sup>7</sup>Será *s* o *S* dependiendo de que el bit esté en 1 ó 0 respectivamente.



del propietario del programa; durante el tiempo de ejecución. Algunos sistemas (SVR4) utilizan este bit para con los directorios: si el directorio tiene el SGID activado, los nuevos ficheros heredan el GID del directorio y los subdirectorios heredan el GID y el bit SGID; en otro caso tanto los ficheros como los directorios son creados con el GID primario del proceso que los crea. Este bit activo nos lo muestra la salida del comando `ls -l` con una **s** en la posición que correspondería al permiso de ejecución para el grupo<sup>8</sup>. Si bien estos bits aportan funcionalidades importantes al sistema, no están exentos de riesgos en cuanto a lo que seguridad se refiere.

### La notación nemónica de `chmod`

Mediante la notación nemónica (o simbólica) también podemos asignar permisos a los ficheros. Es equivalente a la octal y podemos utilizar cualquiera de las dos.

La sintaxis básica es:

```
chmod [usuario] [operador] [permiso] fichero
```

En el lugar de usuario podemos poner:

- u** propietario (*user*)
- g** grupo (*group*)
- o** el resto (*other*)
- a** todos (*all*)

El operador puede ser:

- +** añade el permiso
- quita el permiso
- =** fija el permiso

Los permisos pueden ser:

- r** lectura
- w** escritura
- x** ejecución
- s** set uid o gid
- t** sticky bit

Veamos qué significa con ejemplos:

```
# chmod o=r fichero
```

Con esta orden fijamos los permisos para el resto de la gente (los que no son ni el propietario del fichero, ni el grupo al que pertenece el fichero) como de sólo lectura. El equivalente nemónico sería un 4 en la posición de otros.

```
# chmod u+x fichero
```

Con este comando añadimos (+) al propietario (u) el permiso de ejecución (x)

Es importante destacar, que con la adición no modificamos el resto de los permisos. Si este fichero tuviera permiso de lectura, éste no se vería modificado.

---

<sup>8</sup> Será **s** o **S** dependiendo de que el bit esté en 1 ó 0 respectivamente.

### 13.3. Gestión de usuarios en modo texto.

LINUX proporciona herramientas muy potentes para la administración de usuarios y de grupos. Las órdenes<sup>9</sup> que usaremos para gestionar los usuarios del sistema son:

**useradd** para añadir un usuario<sup>10</sup>.

**passwd** para asignarle la contraseña a un usuario.

**userdel** permite eliminar un usuario.

**usermod** para modificar los datos asociados a una cuenta de usuario.

Supongamos que queremos añadir el valiente usuario THALES CICA a nuestro sistema, y que la contraseña con la que va a poder entrar en el sistema va a ser Averroes, para esto escribiremos:

```
# useradd thales
```

Al ejecutar este comando, primero se lee el fichero `/etc/login.defs`, en él se establecen los valores por defecto a la hora de crear cuentas de usuario. También se creará un directorio de inicio para el usuario en el subdirectorio `/home`, de nombre `thales` cuyo contenido se basa en el directorio de inicio de la plantilla configurada en `/etc/skel`. Además, se añadirán entradas en los ficheros: `/etc/passwd`, `/etc/shadow` y `/etc/group`<sup>11</sup>.

Tras esto deseamos añadir información relativa a este usuario, para realizarlo escribiremos

```
$ chfn thales
```

al ejecutar esta orden, y de forma secuencial, podremos rellenar los campos que aparecen terminados en “:”:

```
Changing finger information for Thales.
Name []: Thales Cica
Office []:
Office Phone []:
Home Phone []:
Finger information changed.
```

en este caso tan solo hemos rellenado el nombre completo de nuestro aguerrido Linuxero.

Ahora tenemos que asignarle una contraseña<sup>12</sup> a este usuario, usaremos el comando:

```
# passwd thales
```

y en la pantalla nos preguntarán por la contraseña que queremos asignar a THALES, como es obvio al escribir la contraseña de nuestro usuario no veremos los caracteres que introducimos. Además, la contraseña tendremos que escribirla dos veces para garantizarnos que escribimos la misma. Si todo ha ido bien el sistema nos informará de ello.

Cada vez que añadimos un usuario se modifica el archivo de contraseñas de usuarios, este archivo es `/etc/passwd`. Si lo mirásemos ahora veríamos que la última línea es:

---

<sup>9</sup>Como siempre no estamos usando todas las opciones de estos scripts, para conocerlas en profundidad os remitimos a las páginas man de cada uno de ellos.

<sup>10</sup>También se puede usar `adduser`

<sup>11</sup>Cada vez que se añade un usuario al sistema se crea un grupo del que él sólo forma parte, se trata del grupo privado de ese usuario.

<sup>12</sup>Hay una serie de normas a seguir al asignar contraseñas:

1. Tienen que ser cadenas de al menos 6 caracteres (mejor si ponemos 8)
2. En las contraseñas es mejor mezclar letras mayúsculas y minúsculas, signos de puntuación y números.



```
Thales:x:506:506:Thales Cica:/home/Thales:/bin/bash
```

En esa línea hay distintos campos separados por “:”, el significado de ella es que el usuario Thales tiene de contraseña “x”<sup>13</sup>, que el número que biunívocamente utiliza el sistema operativo para él es 506 (UID<sup>14</sup>), igual que su número primario de grupo (GID), después vemos el nombre completo<sup>15</sup> y por último el shell<sup>16</sup> que va a ser utilizado por este usuario al arrancar. ¿Fácil no? **Arj**, ¡me salté un campo! el penúltimo campo es el subdirectorio inicial de este famoso usuario.

Cada usuario es miembro de al menos un grupo. El sentido de trabajar con grupos es que se pueden dar privilegios de acceso a determinados ficheros a todos los usuarios pertenecientes a un grupo. La información sobre los grupos se guarda en `/etc/group`.



Algunas órdenes para trabajar con grupos son:

**groups** ver los grupos a los que pertenece el usuario pasado como argumento.

**groupadd** crear un nuevo grupo

**groupmod** modifica un grupo

**groupdel** borra un grupo

Pero nuestro usuario ya no nos es necesario, al finalizar este apartado ya no lo necesito más en mi sistema, ante esta situación escribiré

```
# userdel thales
```

Con esta orden se ha eliminado a nuestro usuario de los ficheros `/etc/passwd` y `etc/shadow` pero su directorio de trabajo sigue sin borrar; para que al ejecutar la orden también borre el directorio de trabajo tendríamos que haber escrito

```
# userdel -r thales
```

como yo no lo escribí, abriré el programa `mc`,



me situaré sobre ese subdirectorio y tras pulsar **F8** y confirmar, se acabó, directorio eliminado.

## 13.4. Gestión de usuarios en modo gráfico.

Disponemos de dos herramientas de uso inmediato:

- GNOME:  → **Configuración del Sistema** → **Gestor de usuarios** o desde una xterm

```
#redhat-config-users
```

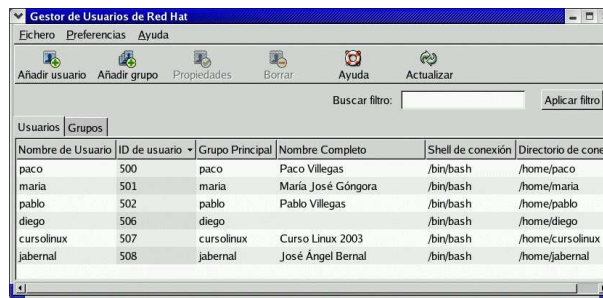
<sup>13</sup>No puede ser tan fácil ver la contraseña ¿verdad?

<sup>14</sup>Los *uids* identifican a los propietarios de los archivos, directorios y procesos.

<sup>15</sup>Si hemos añadido más datos sobre él al usar el comando `chfn` irían aquí separados por comas.

<sup>16</sup>Si ponemos `/bin/false` ese usuario no podrá ejecutar comandos de nuestra máquina aunque sí podrá acceder a ella.





- Si tenemos instalado el KDE<sup>17</sup>:

#kuser



## 13.5. ➡ Para practicar

En esta práctica, se propone crear un entorno de trabajo que ofrezca una posible solución al problema de la compartición de ficheros por parte de una clase de alumnos/departamentos didácticos de un centro.

El problema se va a plantear proponiendo crear un entorno de trabajo para realizar prácticas:

- Habrá un profesor (profesor),
- alumnos (alumno1..alumno4) y
- grupos de prácticas (grupo1 y grupo2).

Por extensión se podría aplicar a un grupo más amplio.

Se trata de conseguir:

1. Para las prácticas individuales, cada alumno tendrá un directorio `/home/alumnoX`. Dentro de él, deben crear el fichero `practica_individual1`. Ese fichero podrá ser visto por el profesor pero no podrá modificarlo. Cada alumno sólo tendrá acceso a sus ficheros.
2. Para las prácticas en grupo cada grupo dispondrá de un directorio `/home/grupo1`, `/home/grupo2` en el que podrán escribir los miembros del grupo. Todos los alumnos pertenecientes a un grupo (alumno1 y alumno2 pertenecen al grupo1 y alumno3 y alumno4 al grupo2) pueden escribir en el directorio del grupo (`/home/grupoX`), pero no pueden leer ni escribir en el directorio de otros grupos.

<sup>17</sup>En 14.5.1 en la página 204 se explica como instalarlo.

3. Habrá un directorio `/home/clase_linux` en el que todos los alumnos tendrán un fichero de nombre `alumnoX`, en el que sólo podrá escribir dicho alumno, pero podrá ser leído por todos los alumnos.

Tenemos que ver qué permisos deben tener los ficheros/directorios para conseguir las condiciones anteriores.

El problema se puede resumir en determinar:

1. Permisos del directorio `/home/alumnoX` y del fichero `/home/alumnoX/practica_individual1` para llevar a cabo la política de accesos adecuadamente. El profesor, sin ser root (superusuario), deberá poder acceder al contenido de todas las prácticas individuales.
2. Permisos del directorio `/home/grupoX` y del fichero `/home/grupoX/practica_grupo1` para llevar a cabo la política de accesos a prácticas de grupo. Por supuesto, el profesor podrá consultar las prácticas del grupo sin ser superusuario.
3. Permisos del directorio `/home/clase_linux` y de los ficheros `alumnoX` dentro de ese directorio.

### 13.5.1. SOLUCIÓN

#### -Prácticas individuales.

##### 1. Creación de los usuarios:

Creamos el usuario profesor mediante el método que prefiramos. Por ejemplo:

```
#adduser profesor
```

Después creamos los distintos usuarios-alumnos:

```
#adduser alumnoi (i:1..4)
```

Para cada usuario asignamos las contraseñas de acceso con el comando

```
#passwd usuario
```

Cada usuario que creamos con RedHat pertenece a un grupo propio, es decir, el propio nombre de usuario. Así al crear el usuario `alumno1`, el identificador de usuario será `alumno1` y el grupo al que pertenece será `alumno1`.

Los permisos que tiene el directorio de cada usuario por defecto son **drwx- - - - -**

Hacemos que el profesor pertenezca al grupo `alumno1`, editando y añadiéndolo en la entrada correspondiente al grupo `alumno1` del fichero `/etc/group`<sup>18</sup>

```
alumno1:x:501:profesor
```

```
...
```

```
alumno4:x:504:profesor
```

##### 2. Modificación de los permisos:

Para ver los permisos de los distintos directorios de usuario lo haremos con la orden

```
#ll /home
```

Si los permisos del directorio cuando se crea son:

```
drwx----- 5 alumno1 alumno1 4096 oct 9 20:33 alumno1
```

Los modificaremos con

```
#chmod g+rx /home/alumno1
```

<sup>18</sup>Se puede en modo texto o con las herramientas gráficas `redhat-config-users` o `kuser`. Los UID de los usuarios no se deben corresponder con las que aparecen a continuación.

quedando:

```
drwxr-x--- 5 alumno1 alumno1 4096 oct 9 20:33 alumno1
```

Los permisos del usuario no necesitan comentario. Para el grupo alumno1 (al cual el profesor pertenece) se permite `r` (ver qué ficheros hay en el directorio) y `x` (poderse meter dentro del directorio). Hacer `cd /home/alumno1`). Con esto conseguimos que el profesor pueda entrar en el directorio del alumno. Es importante destacar lo que no se puede hacer. El profesor, aunque podrá entrar en el directorio, no podrá crear nuevos ficheros, porque no tiene el permiso de escritura en el directorio.

El resto de usuarios del sistema, no podrán entrar, ver, ni modificar nada del directorio del alumno.

### 3. Crear los ficheros **practica\_individuali** para cada uno de los alumnos

Podemos optar por crearlos con un editor de textos (cuidado con quién los crea, si es el root<sup>19</sup>, él será su dueño y después habrá que cambiar propietario y grupo), o bien con el comando:

```
$ touch practica_individual1
```

Los permisos del fichero `practica_individual1` pueden ser:

```
-rw-r--r-- 1 alumno1 alumno1 4 oct 9 20:34 practica_individual1
```

Así, el profesor podrá revisar la práctica del alumno, pero no la podrá modificar. Si el fichero se ha creado con permisos 664, tendremos que cambiarlos, bien como root, bien como alumno1

```
#chmod 644 /home/alumno1/practica_individual
```

## -Práctica de grupo

Creamos los grupos

```
#groupadd grupo1
```

```
#groupadd grupo2
```

y añadimos en el fichero `/etc/group` a alumno1, alumno2 como usuarios pertenecientes al grupo 1 (lo mismo haremos con los alumnos 3 y 4 para el grupo 2)<sup>20</sup>.

```
grupo1:x:506:alumno1,alumno2
```

Podemos comprobar como va el proyecto ejecutando:

```
$id usuario
```

```
y/o
```

```
$groups usuario
```

Después crearemos el directorio `/home/grupo1`:

```
#mkdir /home/grupo1
```

```
#chgrp grupo1 /home/grupo1
```

```
#chown profesor /home/grupo1
```

Nos interesa que el directorio tenga los permisos

```
drwxrwx--- 2 profesor grupo1 4096 oct 9 22:43 grupo1
```

Con esto, el profesor podrá acceder al directorio del grupo, ya que es el propietario, y los alumnos, por el hecho de pertenecer al grupo (`grupo1`) podrán `r` (ver el contenido del directorio), `x` (meterse en el directorio) y `w` (escribir en el directorio, que equivale a poder crear ficheros dentro del directorio).

<sup>19</sup>Recordar que podemos pasar de ser un usuario a otro usando el comando:

```
$su usuario
```

e introduciendo después la contraseña. Para salir

```
$exit.
```

<sup>20</sup>El GID de ejemplo no tiene por que ser el que aparezca en vuestro fichero.



Pero los permisos son:

```
drwxr-xr-x
```

Para cambiarlos y conseguir el efecto deseado, ejecutaremos la orden:

```
#chmod 770 /home/grupo1
```

El resto de alumnos que no pertenecen al grupo1, no podrán hacer nada (ni por supuesto copiar la práctica). Con sólo esto, si el alumno1 hace

```
$cd /home/grupo1
```

y crea el fichero practica\_grupo1, éste se crearía con los permisos:

```
-rw-rw-r-- 1 alumno1 alumno1 2 oct 9 22:42 practica_grupo1
```

y el resto de los alumnos del grupo, podrían ver la práctica, pero no modificarla.

Para solucionar esto, hacemos

```
#chmod g+s /home/grupo1
```

quedando el directorio

```
drwxrws--- 2 profesor grupo1 4096 oct 9 22:43 grupo1
```

El bit *s* en directorios hace que el fichero que se cree dentro de él, posea como grupo al mismo grupo al que pertenece el directorio. Así, el grupo del fichero se mantendrá como grupo1 y el resto de alumnos que pertenecen al grupo, podrán modificar (trabajar sobre) la práctica, aunque el propietario haya sido el alumno que lo ha creado.

```
-rw-rw-r-- 1 alumno1 grupo1 2 oct 9 22:43 practica_grupo1
```

### **-/home/clase\_linux**

Creemos el directorio /home/clase\_linux con el profesor como propietario y perteneciente al grupo profesor.

```
#mkdir /home/clase_linux
#chgrp profesor /home/clase_linux
#chown profesor /home/clase_linux
#chmod a+rw /home/clase_linux
```

el resultado sería:

```
$ll /home
```

```
...
```

```
drwxrwxrwx 2 profesor profesor 4096 oct 9 23:12 clase_linux
```

```
...
```

Si fueran estos permisos, los alumnos (mediante el *rw* del resto de usuarios, puesto que no son ni propietarios ni pertenecen al grupo) pueden crear los ficheros *alumnoX*, pero otros alumnos podrán borrar e incluso modificar los ficheros de otros alumnos, con el consiguiente peligro. Una forma de evitar esto es mediante el *sticky bit*. Hacemos

```
#chmod +t /home/clase_linux
```

El efecto es que aunque el directorio sea de escritura pública (*rw* para todos) solamente el propietario del fichero podrá borrarlo.

```
$ll /home/clase_linux
```

```
drwxrwxrwt 2 profesor profesor 4096 oct 9 23:21 clase_linux
```

Supongamos que alumno1 crea el fichero /home/clase\_linux/fichero1 de permisos



```
-rw-rw-r-- 1 alumno1 alumno1 0 oct 9 23:21 fichero1
```

Aunque por los permisos del directorio pudieran borrarlo otros alumnos, el bit `t` hace que sólo el propietario (`alumno1`) pueda borrarlo<sup>21</sup>. Además, los permisos de este fichero impiden que pueda modificarlo otro alumno.

---

<sup>21</sup>Es lo que pasa en el directorio `/tmp` de uso público.

## Capítulo 14

# Instalación/desinstalación de paquetes.

¿Desea instalar o eliminar una aplicación? No hay problema. ¿Desea actualizar un programa que ya ha instalado? Muy fácil. Con un par de simples comandos o pulsando algunos botones este proceso lo podrá realizar usted mismo. *Red Hat Linux 7.0: The Official Red Hat Linux Getting Started Guide*

### 14.1. Introducción

En este capítulo vamos a estudiar cómo instalar y desinstalar programas. Ya en temas anteriores se ha visto la forma básica de hacerlo. Además, probablemente seáis ya “especialistas” en usar alguna de las utilidades que acompañan a GNOME y KDE<sup>1</sup>.

Estudiamos el comando `rpm` aparte de los otros comandos debido a la “importancia” merecida que tiene. Con respecto a los programas gráficos para instalar y desinstalar programas tan sólo comentaremos los aspectos más básicos para trabajar con ellos ya que, si se sabe qué se puede hacer con el programa `rpm`, su manejo es casi inmediato.

Se puede ampliar sobre el tema en:

- La página man del programa.
- El subdirectorio `/usr/share/doc/rpm-4.1`
- El Howto: Rpm Como
- Y sobre todo la web<sup>2</sup>: [www.rpm.org](http://www.rpm.org)



De nuevo, recordar un comentario realizado al inicio de la 2ª entrega.

Guiaremos la instalación de paquetes de la forma más general posible: en el supuesto de que se tiene que montar el CD y que se instala el paquete en modo comando.

Si se trabaja en modo gráfico (será lo más usual), cuando se introduzca un CD se inicia un proceso de automontado, y, por tanto, no es necesario el comando `mount` (de hecho si se ejecuta dará error). Además, si se trabaja con Nautilus<sup>3</sup>, para instalar un paquete sólo hay que pulsar sobre él (dos veces) con el ratón, lo que implica que tampoco es necesario ejecutar el comando `rpm -ivh nombre_paquete`.

---

<sup>1</sup>En todo el tema daremos por supuesto que se tiene KDE instalado. Si no es así, la práctica final explica cómo hacerlo y una vez instalado se puede comprobar lo que se expone a lo largo del capítulo.

<sup>2</sup>En esta página está el libro de Ed Baile *Maximun RPM* disponible para bajárselo. Está en formato ps/L<sup>A</sup>T<sub>E</sub>X, en Inglés y son sólo 442 páginas.

<sup>3</sup>Gestor de ficheros de GNOME



Red Hat 8.0 instala el agente de actualización *Red Hat Network* (up2date) que nos avisa de las actualizaciones necesarias para tener a la “última” nuestro sistema. Si deseamos trabajar con total funcionalidad con este programa hay que registrarse. Tenéis información sobre él en el Capítulo 33 de *The Official Red Hat Linux Customization Guide* <http://europe.redhat.com/documentation/rhl8.0/rhl-cg-es-8.0/>.

Antes de meternos de lleno con este apartado merece la pena comentar el porqué de la importancia del sistema de paquetes rpm y qué ha supuesto dentro del mundo Linux.

En toda la sección supondremos que el paquete a instalar se encuentra en el directorio actual.

## 14.2. ¿Qué es esto de rpm?

RPM son las siglas de *RedHat Package Management*, es decir, sistema de paquetes de RedHat. El sistema RPM es un sistema abierto, escrito en C, basado en la licencia GNU y hoy por hoy es el sistema de paquetes más extendido<sup>4</sup>.

La idea de paquetes es específica del mundo Linux/UNIX y se justifica en la filosofía imperante en el mundo UNIX: la de disponer de pequeñas utilidades que aunque hagan una sola tarea la hagan muy bien. Mezclando estas “pequeñas” utilidades podremos resolver cualquier problema por grande y complejo que sea. Además, estos programas usan librerías compartidas con el objetivo de minimizar el código duplicado y optimizar el uso del disco y de la memoria. Esta forma de organizar el sistema obliga a mantener un control estricto sobre los programas y librerías instaladas en nuestro equipo.

Aclaremos qué es un **paquete**: es un archivo que contiene todos los ficheros de un determinado componente instalable y que además almacena información de control y *scripts* que se ejecutan al instalar o borrar el paquete.

Con un sistema de paquetes se pretende mantener un control efectivo sobre las aplicaciones (programas, librerías, etc) que instalemos en nuestra máquina y las modificaciones realizadas sobre ellas.

Las características fundamentales del sistema de paquetes RPM son:

- Mantiene una base de datos en la que se almacena la información de todos los paquetes, tanto los paquetes instalados como los ficheros que contiene cada paquete.<sup>5</sup>
- Control sobre dependencias: controlando las dependencias antes de instalar o desinstalar un paquete sabemos si ese paquete necesita/es usado de otros paquetes o librerías<sup>6</sup> para funcionar correctamente. Si no tenemos instalado ese programa/librería, antes de instalar se nos avisará con un error. Lo mismo pasará si queremos desinstalar un paquete que es necesario para otras aplicaciones.
- Control sobre las incompatibilidades: si intentamos instalar un paquete que va a impedir que otro que ya tenemos instalado funcione correctamente surgirá una incompatibilidad.
- Podemos instalar paquetes sin tener que reiniciar el equipo.<sup>7</sup>

El sistema de paquetes RPM sigue una “nomenclatura”<sup>8</sup> que permite identificarlo, se basa en dar de cada paquete los campos: nombre, versión, revisión, plataforma y extensión. Por ejemplo, consideremos el paquete

```
XFree86-VGA16-3.3.6-33.i386.rpm
```

veamos qué significa cada uno de esos campos:

<sup>4</sup>Si bien el sistema RPM es el usado por RedHat, Caldera, SUSE, etc, la realidad es que un paquete creado para una determinada distribución no siempre se instala bien en las otras.

<sup>5</sup>Podremos saber en cualquier momento si hemos modificado los ficheros de un determinado paquete, comprobar la integridad de un paquete o saber a qué paquete pertenece un determinado fichero.

<sup>6</sup>RPM hace referencia a las librerías dinámicas directamente y no a los paquetes que las contienen.

<sup>7</sup>Como en otros sistemas operativos, ¿verdad?

<sup>8</sup>Hay otros paquetes cuyo nombre es de la forma `nombre_paquete-versión.src.rpm`. Estos paquetes constan de los archivos oficiales (fuentes) y se les ha añadido un fichero de especificaciones para facilitar la creación del binario (véase 14.4 en la página 203).

**XFree86-VGA16** es el nombre del paquete.

**3.3.6** es el número de versión, es decir, las fuentes de este paquete se corresponden con la versión 3.3.6, donde este sistema de numeración es el estándar.

**33** número de la revisión, en este caso indica que es la trigésimo tercera modificación realizada por RedHat para adaptar el paquete original a esta distribución.

**i386** nos indica la plataforma<sup>9</sup> para la que está construido. En plataformas Intel disponemos de: i386, i586, athlon, i686. Si bien un paquete para la plataforma i386 podremos instalarlo en cualquier máquina Intel o compatible, uno con “extensión” i686 será sólo para micros Pentium II (o compatibles) y superiores.

**rpm** La extensión común a todos los paquetes RedHat.<sup>10</sup>

En Internet tenemos páginas bastante buenas en las que poder bajar multitud de paquetes, una de ellas es:  
<http://rpmfind.net/linux/RPM/>

Además, con `rpmbuild` podemos construir nosotros paquetes. Este tema se “escapa” de un curso inicial de Linux y sólo daremos algunas pinceladas en 14.4 en la página 203. Tenéis información sobre él en el HOWTO RPM Como y sobre todo en el libro *Maximun RPM*.

## 14.3. Comando rpm

Para instalar/desinstalar paquetes, tanto fuentes como binarios, RedHat dispone de `rpm`<sup>11</sup>. `rpm` es una herramienta que permite:

- Instalar paquetes
- Desinstalar paquetes
- Actualizar paquetes
- Nos da información variada sobre los paquetes.
- Verificar los paquetes.

Antes de estudiar esta herramienta, comentar que hay *front-end* gráficos que facilitan su uso pero que no permiten hacer todo lo que se puede hacer en modo texto. Además, cuando nos habituamos a ella es más rápido y potente su uso que trabajar en modo gráfico.

No vamos a estudiar todas y cada una de las opciones de este comando sino que vamos a enumerar aquellas más “importantes” y usuales. Para ampliar sobre el uso de este comando os remitimos al manual en línea del programa

```
$ man rpm
```

o bien para obtener un listado de las opciones ejecutar

```
$ rpm --help | less
```

Para ejemplificar esta sección trabajaremos sobre el paquete “inexistente”

```
ed03linux-1.0.1-3.rpm
```

utilidad *casi mágica* que nos va a permitir tener un control completo sobre nuestro ordenador.

<sup>9</sup>Para la plataforma Intel también son válidos los paquetes con nombre *noarch*

<sup>10</sup>Existe otro sistema de paquetes menos extendido que el de RedHat, es el sistema de paquetes de la distribución DEBIAN cuya extensión es *.deb*

<sup>11</sup>En *The Official Red Hat Linux Customization Guide*, capítulos 31 a 33, se estudian las posibilidades de este comando bastante bien.



### 14.3.1. Instalación de paquetes

**Sintaxis** `rpm -i [opciones] paquete.rpm`

Comencemos, nos han pasado este paquete y estamos ansiosos de poder usarlo en nuestro equipo, nos situamos en el directorio<sup>12</sup> que contiene el paquete y escribimos:

```
$ rpm -i ed03linux-1.0.1-3.rpm
```

y, como era de esperar la cosa no funciona, veríamos algo parecido a:

```
failed to open //var/lib/rpm/packages.rpm
```

```
error: cannot open //var/lib/rpm/packages.rpm
```

para poder instalar el paquete tenemos que hacerlo como root, como usuario no es posible (afortunadamente) administrar el sistema.

O sea que, o bien abrimos otra sesión como root o bien escribimos

```
$ su
```

y después escribimos la contraseña correspondiente. Cuando todo esté solucionado, intentémoslo de nuevo:

```
# rpm -i ed03linux-1.0.1-3
```

Al poner como opción `-i` le decimos a rpm que vamos a instalar un paquete, la forma genérica de instalar paquetes es:

```
rpm -i[opciones] nombre_paquete
```

las opciones que usualmente acompañan a `-i` son:

```
# rpm -ivh nombre_paquete.rpm
```

dónde las opciones anteriores significan:

**v** (*verbose*) al final de la instalación nos informa de que se ha instalado el paquete, si queremos una información más detallada escribiríamos **vv**, en este caso veremos qué archivos se han instalado.

**h** nos muestra 50 # mientras se instala el paquete, al mezclarlo con **v** se obtiene una monitorización de la instalación más “agradable”.

#### ➡ Para practicar: Instalación de **xfig**.

Se trata de un programa de dibujo vectorial que no se carga cuando instalamos RedHat. Una vez instalado lo podemos ejecutar desde un terminal gráfico usando el comando:

```
$ xfig
```

Primero vamos a comprobar si lo tenemos ya instalado, para esto ejecutamos la orden:

```
$ rpm -q xfig
```

y en pantalla nos aparece:

```
package xfig is not installed
```

Montamos entonces el CD3 (hemos visto que en él está el paquete)

```
$ mount /mnt/cdrom
```

y desde el directorio `/mnt/cdrom/RedHat/RPMS` ejecutamos

```
# rpm -ivh xfig-3.2.3d-10.i386.rpm
```

la salida que nos aparece es<sup>13</sup>:

```
warning: xfig-3.2.3d-10.i386.rpm: V3 DSA signature: NOKEY, key ID db42a60e
```

<sup>12</sup>También se pueden instalar paquetes desde otros directorios, en ese caso habrá que dar el path completo. Es posible instalar paquetes vía Internet (vía ftp), por ejemplo si queremos instalar el paquete `paquete.rpm` desde [mileto.cica.es/pub](http://mileto.cica.es/pub) escribiríamos:  

```
# rpm -i ftp://mileto.cica.es/pub/paquete.rpm
```

<sup>13</sup>La primera línea (no la pondremos más aunque os saldrá al instalar los paquetes) me indica que no tengo instalada la clave necesaria para verificar la firma del paquete. Si se desea instalarla se puede consultar **El control de la firma del paquete** en el capítulo 31 de *The Official Red Hat Linux Customization Guide*.

```
error: failed dependencies:
```

```
transfig >= 3.2.3d is needed by xfig-3.2.3d-10
```

Esto quiere decir que necesitamos el paquete `transfig` (una versión igual o posterior a la 3.2.3d) para poder instalar `xfig`. Para ver qué versión es la que tenemos usamos la orden

```
#ls tran*
```

que nos informa de la versión que viene en el CD3

```
transfig-3.2.3d-7.i386.rpm
```

Ahora instalamos primero el paquete `transfig` y por último el `xfig`

```
#rpm -ivh transfig-3.2.3d-7.i386.rpm xfig-3.2.3d-10.i386.rpm
```

```
Preparing... #####
[100%]
    1:transfig #####
[ 50%]
    2:xfig #####
[100%]
```

Al intentar instalar nuestro paquete `ed03linux-1.0.1-3.rpm` el sistema nos devuelve el mensaje:

```
error: failed dependencies:
```

```
libed03 >= 2 is needed by ed03linux-1.0.1-3
```

### Fallo de dependencias.

Ante este error tenemos varias opciones:

1. Pasar olímpicamente de las dependencias y obligar a que se instale el paquete  

```
# rpm -ivh --nodeps ed03linux-1.0.1-3.rpm
```

Esta opción no es recomendable ya que nos arriesgamos a que de todas maneras no funcione nada.
2. **Solucionar el problema de dependencias.** Como en general no tenemos ni idea de dónde proviene el problema podemos buscar si ya tenemos instalada<sup>14</sup> esa “librería” y ver de qué paquete forma parte. Para saber si ya está instalada<sup>15</sup> podemos usar varias herramientas<sup>16</sup>:

a) Comando `find`

b) Programa `mc` y buscar ficheros.

c)  → **Buscar Archivos** (`gnome-search-tool`).

d) Usar el comando `locate`: el comando `locate` permite buscar un fichero en la base de datos (con todos los ficheros del sistema) que se genera al usar `updatedb`. Si no hemos usado nunca el comando `updatedb` escribiremos:

```
# updatedb &17
```

Al finalizar la ejecución del comando escribiremos:

```
$ locate libed03
```

si el sistema nos responde que ha encontrado la librería, será porque la versión que tenemos instalada es menor que la que necesita nuestra aplicación.

<sup>14</sup>Esto que vamos a ver ahora es del punto 14.3.5, pero ...

<sup>15</sup>Si no está instalada tenemos que saber antes de nada a qué paquete pertenece, si no, no podemos hacer nada. Ante esta opción lo mejor es preguntar en alguna lista de correo o usar un buscador para ver de dónde ha salido. Un lugar bastante socorrido para esto (para buscar) es la propia página de RedHat:

[www.redhat.com](http://www.redhat.com)

<sup>16</sup>Hay más posibilidades.

<sup>17</sup>El `&` para que se ejecute en segundo plano. Este comando no hay que usarlo cada vez que queramos buscar.

Supongamos que la respuesta es:

```
/usr/lib/Thales/libed03.so.1
```

Para saber a qué paquete pertenece escribiremos:

```
$ rpm -qf /usr/lib/Thales/libed03.so.1
```

a lo que el sistema responde:

```
libed03-1
```

¿Qué significa esto? ¿cómo arreglarlo?. La cuestión está clara, tenemos instalado el paquete `libed03-1.rpm` y necesitamos un paquete más “moderno”; una posible opción es buscarlo, para nuestra distribución, en Internet<sup>18</sup>.

Una vez encontrado el paquete `libed03-2.rpm`<sup>19</sup>, lo instalaremos en nuestro sistema<sup>20</sup>.

3. Si la librería `libed03` no está instalada en nuestro sistema no podemos saber con este método a qué paquete pertenece. Para enfrentarnos a esta opción, en el CD3 de la distribución disponemos de un paquete con una base de datos en la que aparecen todas la librerías y los paquetes que las contienen, es: `rpmdb-redhat-8.0-0.20020910.i386.rpm`. Si montamos el CD3 y lo instalamos:

```
# rpm -i rpmdb-redhat-8.0-0.20020910.i386.rpm
```

ejecutaremos

```
$ rpm -q --redhatprovides libed03
```

y si está en algún paquete de los CDs nos dirá en cual.

En última instancia, si no funciona ningún método no tendremos más remedio que usar Internet (algún buscador) para poder saber dónde encontrarla y bajarla.

Una vez localizada y en nuestro equipo, actualizamos el paquete:

```
# rpm -Uvh libed03-2.rpm21
```

Pero nuestro disco duro está ya sobrecargado de programas y queremos que el paquete que vamos a instalar no nos instale la documentación que contiene, en este caso escribiremos:

```
# rpm -ivh --excludedocs ed03linux-1.0.1-3.rpm
```

## ➡ Para practicar:

1. Uso del comando `locate`

Primero vamos a instalar el paquete `rpmdb-redhat-8.0-0.20020910.i386.rpm` del CD3

```
# rpm -ivh rpmdb-redhat-8.0-0.20020910.i386.rpm
```

```
Preparing... #####
[100%]
```

```
1:rpmdb-redhat #####
[100%]
```

Una vez instalada ejecutemos:

```
# locate libphp4.so
```

y comprobaremos que esa librería no está en nuestro sistema, veamos de qué paquete es con:

```
# rpm -q --redhatprovides libphp4.so
```

<sup>18</sup>Google o <http://rpmfind.net/linux/RPM/>

<sup>19</sup>Notad que la librería requerida era de la forma `libed03 >= 2`

<sup>20</sup>¿Por qué no puede ser todo más sencillo?, ¿por qué no instalará este sistema las cosas sin que yo me entere de nada ni de dónde las instala? ¿por qué no modifica los ficheros sin mi consentimiento?, ¿por qué no “machaca” ficheros sin decirme nada? ...¿qué más da si luego no funciona nada!.

<sup>21</sup>La U significa *Update*, es decir, actualizar el paquete. Después ampliaremos sobre sus opciones.

que nos dará en pantalla

```
php-4.2.2-8.0.5
```

esta salida nos informa que pertenece al paquete `php-4.2.2-8.0.5` (lenguaje de programación, ya veremos algo más sobre él).

2. Observemos un cambio generado al instalar el paquete anterior, si intentamos instalar del CD3 el paquete

```
#rpm -ivh gsl-devel-1.1.1-3.i386.rpm
```

```
error: Failed dependencies:
```

```
gsl = 1.1.1 is needed by gsl-devel-1.1.1-3
```

```
Suggested resolutions:
```

```
gsl-1.1.1-3.i386.rpm
```

Es decir, no sólo se nos avisa de un problema de dependencias, si no que si el paquete está en algún CD de la distribución, nos informa de cuál se trata: en este caso de `gsl` (rutinas y librerías escritas en C para análisis numérico).

### 14.3.2. Actualización de paquetes

**Sintaxis** `rpm -U [opciones] paquete.rpm`

Nos enteramos de que el magnífico programa con el que estábamos trabajando se ha quedado viejo, ha salido una versión nueva, la 1.0.2, y ante esta tesitura está claro, hay que actualizar, para esto:

```
# rpm -Uvh ed03linux-1.0.2-1.rpm
```

¿Pero qué pasa ahora?. Cuando actualizamos un paquete la utilidad `rpm` detecta si se han producido cambios en los ficheros de configuración del paquete y, si esto ha ocurrido y los ficheros de configuración son compatibles con la versión antigua, mantendrá estos ficheros sin tocar. En el caso de que los ficheros de configuración no sean totalmente compatibles instalará los nuevos y renombrará los antiguos con el mismo nombre y extensión `.rpmsave`.



Es mejor actualizar un paquete que desinstalar la versión antigua e instalar después la nueva. De echo una opción a tener en cuenta es actualizar los paquetes, tanto para las actualizaciones como para las instalaciones.

#### ➡ Para practicar: Actualización de paquetes

En <http://www.redhat.com/apps/support/errata/> podemos ver qué actualizaciones de paquetes han aparecido para las distintas versiones de Red Hat. Veamos cómo actualizar el primer paquete en el que se encontró un error para la 8.0.

Comprobemos que versión del paquete `fetchmail`<sup>22</sup> hay instalado en nuestro sistema

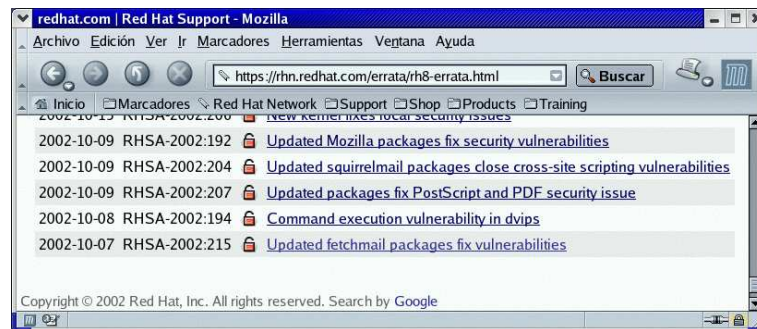
```
$ rpm -q fetchmail
```

```
fetchmail-5.9.0-16
```

Si en la página anterior accedemos a las *updates* de Red Hat 8.0, veremos que el primer paquete a actualizar fue `fetchmail`

---

<sup>22</sup>Se utiliza para bajar y redistribuir el correo-e a los usuarios del sistema.



Accedamos a la página enlazada y bajémosnos el paquete actualizado para Red Hat 8.0: `fetchmail-5.9.0-21.i386`. Para actualizarlo

```
# rpm -Uvh fetchmail-5.9.0-21.i386.rpm
Preparing... #####
[100%]
1:fetchmail ##### [100%]
```

hacer notar que si no está instalado lo instala y si ya está instalado lo actualiza.

**Sintaxis** `rpm -F [opciones] paquete.rpm`

Pero, ¿qué hacer cuando me dan un CD con un programa que ya tengo instalado y que se compone de varios paquetes?, ¿cómo actualizar sólo aquellos que ya tengo instalados?. Supongamos que tenemos instalada la versión 1.9 del KDE y un “amigo” nos ha pasado la versión 3.0. El KDE no es un paquete, está formado de múltiples paquetes y algunos estarán ya instalados en nuestro sistema y otros no. A nosotros sólo nos interesa actualizar aquellos que están ya instalados pero sin hacerlo de uno en uno, en ese caso sólo tenemos que ejecutar la orden:

```
# rpm -Fvh *
```

desde el directorio que contiene la actualización, y `rpm` se encargará de ver qué paquetes están instalados y los actualizará a la nueva versión.



- Antes de actualizar el sistema a lo “bestia” se debería mirar en <https://rhn.redhat.com/errata/rh8-errata.html>
- Mientras que con el núcleo es necesario revisar: <http://www.redhat.com/support/resources/howto/kernel-upgrade/>

#### ➡ Para practicar: Actualizar el sistema

Veamos como actualizar el navegador Web Mozilla sin comprobar previamente qué componentes tenemos instalados. Si accedemos a <https://rhn.redhat.com/errata/RHSA-2002-192.html> podemos comprobar que están disponibles los paquetes:

```
galeon-1.2.6-0.8.0.i386.rpm
mozilla-1.0.1-26.i386.rpm
mozilla-chat-1.0.1-26.i386.rpm
mozilla-devel-1.0.1-26.i386.rpm
mozilla-dom-inspector-1.0.1-26.i386.rpm
mozilla-js-debugger-1.0.1-26.i386.rpm
```

```
mozilla-mail-1.0.1-26.i386.rpm
mozilla-nspr-1.0.1-26.i386.rpm
mozilla-nspr-devel-1.0.1-26.i386.rpm
mozilla-nss-1.0.1-26.i386.rpm
mozilla-nss-devel-1.0.1-26.i386.rpm
mozilla-psm-1.0.1-26.i386.rpm
```

Algunos están instalados en nuestra máquina y otros no.

Creemos un directorio para ponerlos (por ejemplo en `/root/mozilla`) y bajémoslos a ese directorio. Una vez en ese subdirectorio ejecutamos

```
# rpm -Fvh *.rpm
```

### 14.3.3. Desinstalación de paquetes

**Sintaxis** `rpm -e [opciones] paquete`

La opción que hay que usar para desinstalar un paquete es `-e`.

Supongamos que ya no necesitamos más nuestro programa, entonces escribiremos:

```
# rpm -e ed03linux
```

Hay que hacer notar que a la hora de desinstalar un paquete tan sólo hay que escribir el nombre del paquete y que no es necesario escribir ni la versión ni la extensión.

Si un paquete es necesario para que otros funcionen se nos avisará y, salvo que forcemos su desinstalación, ésta no será posible.

Es útil saber que podemos testear qué ocurriría si desinstalamos un paquete sin hacerlo realmente, para eso usaremos la opción `--test`, así si queremos saber qué problemas pueden presentarse al desinstalar nuestro paquete `ed03linux` escribiríamos:

```
$ rpm -e --test ed03linux
```

Si no sale ningún mensaje de error es que no tenemos ningún problema para hacerlo.

Si al querer desinstalar un paquete se nos avisa de que no podemos por un fallo en las dependencias y, aún así, nos empeñamos en ello (cosa no recomendable) podemos escribir:

```
# rpm -e --nodeps paquete
```

#### ➤ Para practicar: Desinstalación de paquetes

Desinstalemos el paquete `xfig`, ya que por ahora no vamos a dibujar:

```
# rpm -e transfig
```

!EH! así no se puede, hace falta para `xfig`

```
# rpm -e xfig
```

ahora sí

```
# rpm -e transfig
```

### 14.3.4. Información sobre los paquetes.

**Sintaxis** `rpm -q [opciones] paquete.rpm`

Ya hemos visto en apartados anteriores algunas opciones que permiten obtener información sobre los paquetes, ampliemos un poco sobre este tema.

Si queremos ver el listado de ficheros que contiene el paquete `ed03linux`, que ya está instalado en nuestra máquina escribiremos:

```
$ rpm -ql ed03linux
```

Como ya se comentó en entregas anteriores, si deseamos ver ese listado más “cómodamente” usaremos:

```
$ rpm -ql ed03linux | less
```

Para ver un listado de todos los paquetes instalados en nuestra máquina escribiremos:

```
$ rpm -qa
```

mejor usar el filtro less para ver los paquetes, así la orden quedaría:

```
$ rpm -qa | less
```

¿Cómo podemos determinar si un/unos determinados paquetes están instalados en nuestro sistema?

Sencillo, de todos los paquetes seleccionemos aquellos que nos interesan. Por ejemplo, para conocer qué paquetes instalados en nuestra máquina contienen los caracteres “ed03” escribiremos:

```
$ rpm -qa | grep ed0323
```

Otras posibles opciones para verificar paquetes son:

```
$ rpm -qi paquete → Muestra información de ese paquete: nombre, revisión
```

```
$ rpm -qc paquete → Muestra los ficheros de configuración
```

```
$ rpm -qd paquete → Muestra los ficheros de documentación del paquete.
```

```
$ rpm -qR paquete → Muestra todas las dependencias del paquete.
```

### Paquetes no instalados

En el caso de que el paquete no esté instalado tendremos que usar la opción -p. Así, para listar los ficheros del paquete (no instalado) ed03linux-1.0.1-3.rpm, escribiremos:

```
$ rpm -qpl ed03linux-1.0.1-3.rpm
```

y para obtener información sobre él:

```
$ rpm -qpi ed03linux-1.0.1-3.rpm
```

### ➡ Para practicar:

#### 1. Información de un paquete

- a) Comprobar qué paquetes instalados contienen la cadena "ta"

```
$rpm -qa | grep ta
```

- b) Comprobar a qué paquete pertenece un fichero ya instalado, por ejemplo /etc/inittab

```
$rpm -qf /etc/inittab
initscripts-6.95-1
```

- c) Qué ficheros componen ese paquete

```
$rpm -ql initscripts
```

- d) Comprobar de qué va un paquete

```
$rpm -qi initscripts
```

- e) ¿Qué dependencias tiene?

```
$rpm -qR initscripts
```

#### 2. Ampliación:

- a) Comprobar las opciones -qa, -qi, -qc, -qd, -qR para algún paquete instalado en el sistema que en su nombre contenga la cadena “pop”.
- b) Comprobar las opciones -qpl y -qpi para algún paquete no instalado del CD2.

### 14.3.5. Verificación de los paquetes.

**Sintaxis** rpm -V [opciones] paquete.rpm

Si queremos verificar el paquete ed03linux-1.0.1-3.rpm escribiremos:

```
$ rpm -V ed03linux
```

La salida de este comando por pantalla son 8 caracteres por cada fichero del paquete, cada uno de esos caracteres se corresponde con una verificación diferente del paquete. Una posible salida para un fichero es:

```
SM5DLUGT nombre_fichero_del_paquete
```

dónde cada una de las letras anteriores significa:

<sup>23</sup>grep es un filtro que busca la cadena pasada como argumento.

**S** Se ha modificado el tamaño del fichero.

**M** Los permisos del fichero y el tipo de fichero son distintos.

**5** Fallo al comprobar la suma del chequeo interno<sup>24</sup>.

**L** Se ha modificado el link simbólico.

**D** El fichero de dispositivo es diferente.

**U** Se ha modificado el usuario al que pertenece el fichero.

**G** Se ha modificado el grupo al que pertenece el fichero.

**T** Fallo en Mtime, el tiempo de modificación de archivo es diferente.

En el caso de que aparezca *missing* pues está claro, desaparecido; si no aparece el nombre de ningún fichero es que no se hay problemas y un “.” indica que ese test se ha superado satisfactoriamente.

Para verificar todo el sistema de paquetes usaremos:

```
$ rpm -Va
```

#### ➡ Para practicar

Comprueba el efecto de la orden

```
$ rpm -Va
```

sobre tu sistema.

## 14.4. ➡ Para practicar: Paquetes fuentes.

Vamos a dar un par de pinceladas sobre la posibilidad de crear nosotros un paquete basándonos en un fichero fuente *src*. La herramienta necesaria es *rpmbuild*

Lo ejemplificaremos sobre un paquete fuente contenido en el CD3 del que ya hemos hablado, se trata de *transfig-3.2.3d-7.src.rpm*<sup>25</sup>. Veamos qué hay que hacer:

Montemos el CD3

```
# mount /mnt/cdrom
```

Instalemos el paquete de fuentes

```
# rpm -ivh /mnt/cdrom/SRPMs/transfig-3.2.3d-7.src.rpm
```

El contenido de este paquete se encuentra en varios subdirectorios de */usr/src/redhat*. El que más nos interesa es el fichero de especificaciones del paquete, se trata de */usr/src/redhat/SPEC/transfig.spec*

si ahora ejecutamos

```
# rpmbuild -bb /usr/src/redhat/SPEC/transfig.spec
```

se inicia el proceso de construcción sólo del binario (para ampliar: *man rpmbuild*). Cuando finalice este proceso lo tendremos a nuestra disposición en */usr/src/redhat/RPMS/ix86*. Compilado por nosotros y optimizado para nuestra máquina.

En este proceso se genera una serie de “basura” que podemos borrar cuando finalice todo, se trata del contenido de los directorios<sup>26</sup>

- */usr/src/redhat/BUILD*

- */usr/src/redhat/SOURCES*

- */usr/src/redhat/SPECS*

<sup>24</sup>Suma de control calculada con el algoritmo MD5

<sup>25</sup>Construir el binario de este paquete no tiene sentido práctico ya que está en ese mismo CD, el motivo de optar por él reside en que no presenta ningún problema su construcción.

<sup>26</sup>No los directorios



## 14.5. ➡ Para practicar: redhat-config-packages

Como ya se comentó en la introducción, tan sólo vamos a iniciarnos en el uso de este programa. Para ampliar sobre cómo se usa se puede mirar en la documentación que acompaña a RedHat o el capítulo 32 de *The Official Red Hat Linux Customization Guide* (<http://europe.redhat.com/documentation/rhl8.0/rhl-cg-es-8.0/>).

El *front-end* que se carga para trabajar con paquetes es el programa `redhat-config-packages`. Para acceder a él, escribiremos desde una `xterm`:

```
# redhat-config-packages
```


y si trabajamos con el lenguaje en castellano hará “crack”. Pero en Linux todo tiene solución, y en este caso la solución se consigue con una actualización del paquete (disponible en <http://people.redhat.com/~katzj/redhat-config-packages/1.0.2-1/>) se trata del paquete<sup>27</sup>: `redhat-config-packages`

O sea que lo primero es bajarlo e instalarlo en nuestro ordenador:

```
#rpm -Uvh redhat-config-packages-1.0.2-1.noarch.rpm
```

Ya sí, con esta versión podemos trabajar sin problema.

### 14.5.1. Instalación del KDE

Ejecutemos el programa tal cual se ha explicado antes, o bien con  → **Configuración del sistema** → **Paquetes**:

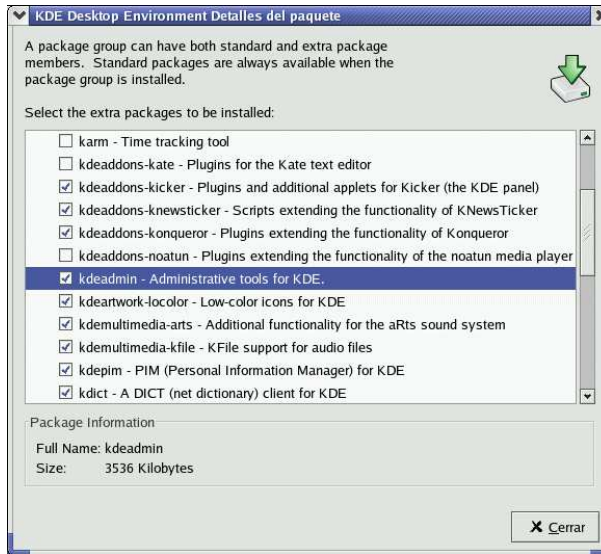


En esta ventana tenemos un árbol de aplicaciones que nos permite comprobar qué paquetes tenemos instalados en nuestro equipo. Si deseamos instalar un paquete no instalado, sólo tendremos que marcarlo y después pulsar sobre el botón **Actualización**.

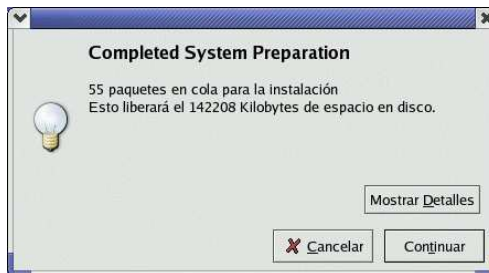
Vamos a instalar el KDE así que optemos por marcarlo tal cual aparece en el gráfico. Se nos informa que de los 58 paquetes que lo componen se van a instalar 53. Como queremos ver como funciona el program `kuser`<sup>28</sup> que forma parte del paquete `kdeadmin`, pulsamos sobre detalles y los marcamos

<sup>27</sup>Se puede bajar desde la Web del curso.

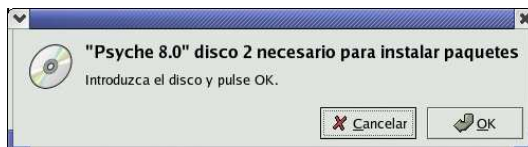
<sup>28</sup>Gestor de usuarios del KDE. Se habló de él en la página 188.



Una vez cerrada esta ventana optamos por pulsar sobre **Actualización** y comienza el porceso de instalación. Después de comprobar los problemas de dependencias, nos aparecerá un cuadro de diálogo mediante el que prodremos comprobar qué se va a instalar. Una vez que estemos de acuerdo, pulsaremos sobre **Continuar**.



Se inicia el proceso de preparación de la instalación y, tras un momento, se nos pedirá que introduzcams, primero el CD2 de Red Hat 8



Y por último el CD 3. Listo, si no habíamos instalado KDE, ya sí que podemos comprobar todo lo comentado en los apuntes sobre este gran entorno de escritorio.

# **Parte IV**

## **La Shell Bash**

# Capítulo 15

## La Shell Bash I

Tras la instalación de todos los programas educativos preguntaremos una y otra vez “TODO ESTÁ EN EL MENÚ, ¿NO?”. Haremos eso cada vez que el *pringao* nos intente explicar qué son ficheros, carpetas y chorradas de esas. Si nos intenta enseñar una ventana negra en la que hay que ¡¡ESCRIBIR!! (sí, amigos, en pleno siglo 21 hay que escribir cosas) y además ¡¡EN INGLÉS!! le diremos que no lo entendemos y que nos ponga eso en el menú. Si nos dice que no se puede poner eso en el menú haremos referencia a que creíamos que él sabía más de informática... (*Pringao Howto* (o *Windows-es-fácil-Howto*), SANTIAGO ROMERO AKA NOP/COMPILER)

### 15.1. La Shell Bash

#### 15.1.1. ¿Qué es una shell?

Básicamente, una *shell* (la traducción al pie de la letra de caparazón o concha no parece muy adecuada) es un procesador de órdenes que sirve para ejecutar comandos. En un principio, cuando no había entornos de ventanas, era la forma de poder comunicar nuestras órdenes al sistema operativo. Además de para ejecutar comandos, la shell de Unix/Linux sirve como lenguaje de programación y para combinar comandos, formando otros más complejos.

Específicamente, *bash* (la shell que presenta Linux por defecto) es un intérprete de un lenguaje de órdenes compatible con *sh* (una shell anterior) que ejecuta órdenes leídas desde la entrada estándar<sup>1</sup> o desde un fichero. Bash también incorpora características útiles tomadas de otras shells como la Korn y la C shell (*ksh* y *csh*).

El nombre de *bash* viene de Bourne Again Shell<sup>2</sup>. Bash está pensado con la intención de ser una implementación conforme con la especificación POSIX de Shell y Herramientas, de la IEEE (Grupo de Trabajo 1003.2 de la IEEE).

Ya que hemos mencionado POSIX, debemos decir que es un estándar<sup>3</sup> (normas escritas en papel, para que lo veamos de una forma más práctica) que pretende definir un Sistema Operativo Abierto<sup>4</sup> definido por la IEEE (que es una organización internacional de estándares). Linux intenta cumplir con los estándares POSIX.

Para poder estudiar más sobre la shell *bash*:

- Los *howtos* traducidos en el INSFLUG.
- La completa página *man* de la shell Bash

---

<sup>1</sup>Más adelante veremos más sobre esto. Por ahora, podemos identificar la entrada estándar con el teclado y la salida estándar con la pantalla.

<sup>2</sup>Algo así como la shell Bourne viene de nuevo, aunque una traducción más libre quedaría como “la Bourne Shell Contraataca”.

<sup>3</sup>Los estándares de derecho (*de iure*), emitidos por organismos independientes y reconocidos, son importantes porque permiten la independencia de un determinado fabricante y fomentan la interoperabilidad de distintos sistemas. Un “estándar” de hecho (*de facto*), simplemente puede reconocer el monopolio de un fabricante.

<sup>4</sup>Abierto en el sentido de no perteneciente a ninguna empresa o grupo y con unas reglas claras para poder operar con él.

- Bash Reference Manual <http://www.gnu.org/manual/bash-2.05a/>

### 15.1.2. Características básicas de la Shell.

El funcionamiento de la shell es el siguiente:

1. Lee la entrada desde teclado o desde un fichero.
2. Divide la entrada en palabras y operadores, obteniendo los comandos.
3. Realiza las expansiones correspondientes y las redirecciones de salida.
4. Ejecuta la o las órdenes.
5. Espera (opcionalmente) a que terminen las órdenes y devuelve un valor de estado de finalización. El valor de estado 0 (cero) significa finalización sin errores y un valor distinto de cero indica el código de error producido.

### 15.1.3. Variables de entorno de la Bash

La shell utiliza unas ciertas variables ya predefinidas (algunas de ellas las veremos a continuación) o podemos definir las nuestras mediante las órdenes:

`$MIVAR=valor` (damos valor a la variable de entorno)

`$export MIVAR` (la exportamos para que sea visible en esta shell y todos los procesos hijos<sup>5</sup> de esta shell)

Si queremos ver el valor, podemos ejecutar

`$echo $MIVAR`

Si pusiéramos como valor de la variable un comando, por ejemplo `ls`, podríamos invocarlo de la siguiente forma

`$MICOMANDO=ls`

`$$MICOMANDO`

(el primer símbolo de `$` es el prompt y el segundo sirve para obtener el valor de la variable).

Podemos ver el valor de todas las variables de entorno definidas en nuestra shell con el comando `set`.

Algunas de las variables de entorno que utiliza bash son:

**HOME:** El directorio de comienzo del usuario.

**PATH:** Una lista de directorios separados cada uno de ellos por el carácter dos puntos (`:`) que nos indica en qué directorios busca la shell para encontrar los comandos. Escoge el comando que primero encuentre, en caso de que pueda encontrarse en varios sitios. Si no lo encuentra dentro de esta lista de directorios, nos devolverá un error con el mensaje “Comando no encontrado” o “command not found”.

**PS1:** El prompt (o indicador de inicio) que presenta la bash al usuario.

**PWD:** El directorio de trabajo actual.

➡ **Para practicar: Ver el valor de las variables de entorno.**

`$echo $PATH`

`$echo $PWD`

`$set`

---

<sup>5</sup>Ya hablaremos sobre los procesos. Por ahora, sepamos que la shell ejecuta los comandos que le introducimos como procesos hijos. La shell se encarga de que nazcan, realicen su tarea y mueran cuando finalicen.

#### 15.1.4. Ficheros de inicio de la bash

Dependiendo de la shell con que entre el usuario al sistema, se ejecutan una serie de ficheros que le configuran su entorno de trabajo. Existen unos ficheros generales que se ejecutan para todos los usuarios que entran al sistema con una misma shell (como por ejemplo el `/etc/profile` para las shell Bourne y Korn), y otros específicos para cada usuario y que se encuentran en su directorio HOME. Estos ficheros de inicialización son utilizados para establecer el camino de búsqueda de ficheros ejecutables, establecer protección por defecto de los ficheros que se creen, tipo de terminal desde el que se trabaja y otras variables de entorno. Estos ficheros son:

**/etc/profile** con él configuramos información de entorno de usuario para todos los usuarios del sistema. Es del root. Se lee una sola vez cuando se inicia el sistema y con él se establecen:

- el prompt por defecto
- el path por defecto
- el tamaño máximo de los ficheros que podemos crear
- los permisos por defecto para los ficheros que creamos.
- tamaño de los ficheros de historial
- ...

**/etc/bashrc** es el fichero de configuración genérico de todos los usuarios que ejecutan el shell bash. En él se puede determinar el prompt por defecto y alias para determinados comandos. Es del root.

**~/ .bash\_profile** permite introducir información específica para cada usuario para las shell del sistema. Se lee sólo una vez cuando el usuario accede en el sistema. En él hay una llamada que hace que se ejecute `.bashrc`.

**~/ .bashrc** información/configuración específica de un usuario para la shell bash. Puede modificar los valores que se cargaron al leer el fichero `/etc/bashrc`. Su contenido se lee cada vez que se entra en el sistema y cada vez que se abre un nuevo shell bash.

**~/ .bash\_logout** se ejecuta cada vez que sale del sistema (borra la pantalla).

Cuando la bash es llamada como una shell interactiva<sup>6</sup> de comienzo, lo primero que hace es leer y ejecutar los comandos que se encuentran en el fichero `/etc/profile`. Después, pasa a los ficheros `~/ .bash_profile`, `~/ .bash_login`, y `~/ .profile`, en ese orden.<sup>7</sup>

Una característica menos usada es que al terminar, la shell bash lee y ejecuta el fichero `~/ .bash_logout` si éste existe

Cuando se trata de una shell interactiva pero que no es de comienzo<sup>8</sup>, el fichero que ejecuta es `~/ .bashrc`.

Cuando la shell no es interactiva, ejecuta el fichero especificado en la variable `BASH_ENV`.

Veamos por ejemplo el contenido de `/etc/profile`. Se encarga de que tengamos el entorno listo para trabajar. Este fichero sería ejecutado al entrar por todos los usuarios del sistema y modificable sólo por el superusuario, mientras que los que se encuentran bajo el directorio HOME (`~`) de cada usuario son configurables y personalizables por éstos.

```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

pathmunge () {
```

---

<sup>6</sup>La que vemos normalmente y le introducimos comandos. Cuando termina el comando, nos presenta otra vez el símbolo del sistema para continuar.

<sup>7</sup>El carácter `~` nos indica el directorio HOME del usuario.

<sup>8</sup>En la entrada al sistema.



```

        if ! echo $PATH | /bin/egrep -q "^(|:)$1($|:)" ; then
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
        fi
    }

# Path manipulation
if [ `id -u` = 0 ]; then
    pathmunge /sbin
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
fi

pathmunge /usr/X11R6/bin after

unset pathmunge

# No core files by default
ulimit -S -c 0 >/dev/null 2>&1

USER=`id -un`
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"

HOSTNAME=`/bin/hostname`
HISTSIZE=1000

if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC

for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        . $i
    fi
done

unset i

```

### 15.1.5. Personalizando el Prompt

Como vimos antes, el valor de la variable PS1 es lo que se nos presenta en el prompt del sistema. Existen algunos valores que podemos utilizar.

El valor predeterminado es PS1="[\u@\h \W]\\$". Podemos ver el nuestro mediante `$echo $PS1`

**\d** la fecha en el formato "Día-Semana Mes Día" (ejemplo, "Tue May 26") en inglés

**\e** un carácter de escape (ESC) ASCII (033)

**\h** el nombre del computador hasta el primer '.'

- `\H` el nombre del computador con dominio completo
- `\n` salto de línea
- `\r` retorno de carro
- `\s` el nombre del shell. El nombre base del ejecutable de la shell (la porción que sigue a la última barra inclinada)
- `\t` la hora actual en el formato de 24 horas HH:MM:SS
- `\T` la hora actual en el formato de 12 horas HH:MM:SS
- `\@` la hora actual en el formato de 12 horas con indicador AM/PM
- `\u` el nombre de usuario del usuario actual
- `\v` la versión de bash (e.g., 2.00)
- `\W` la distribución de bash, versión + nivel de parches (e.g., 2.00.0)
- `\w` el directorio de trabajo en curso
- `\W` el nombre base del directorio de trabajo
- `\!` el número de historia de esta orden
- `\#` el número de orden de este comando en la shell actual
- `\$` si el UID efectivo es 0 (el super-usuario root), un #. Si no lo es, un \$
- `\nnn` el carácter correspondiente al número octal nnn
- `\` una barra inclinada invertida
- `\[` empieza una secuencia de caracteres no imprimibles, que pueden emplearse para insertar una secuencia de control del terminal en el indicador
- `\]` termina una secuencia de caracteres no imprimibles

➡ **Para practicar:** Comprobar el resultado de establecer `PS1="[ \t] [\u@\h:\W] "`

Para ampliar sobre este tema se puede consultar el HOWTO: *Bash Prompt COMO*.

### 15.1.6. Los Alias

Los alias permiten que una cadena (normalmente un comando complejo) se sustituya por una sola palabra cuando se emplee como la primera palabra de una orden simple. ¿Qué conseguimos con esto? Economía de escritura, pues con sólo teclear una palabra estaremos realizando una labor más compleja con menos caracteres que teclear.

El shell mantiene una lista de alias que pueden ponerse y quitarse con las órdenes internas `alias` y `unalias`. Se mira para ver si la primera palabra de cada orden, si no está entrecomillada, tiene un alias. Si es así, la palabra se reemplaza con el texto del alias. El nombre del alias y el texto por el que se reemplaza, pueden contener cualquier entrada válida para el shell, incluyendo metacaracteres, con la excepción de que el nombre del alias no puede contener un `=`. La primera palabra del texto de reemplazo se comprueba también para ver si es un alias, pero si es un alias idéntico al que se está expandiendo, no se expande una segunda vez. Esto significa que uno puede poner un alias `"ls"` a `"ls -F"`, por ejemplo, y bash no intenta expandir recursivamente el texto de reemplazo. Si el último carácter del valor del alias es un blanco, entonces la siguiente palabra de la orden que sigue al alias también se mira para la expansión de alias.

Los alias se crean y muestran con la orden `alias`, y se quitan con la orden `unalias`.



No hay ningún mecanismo para poder usar argumentos en el texto de reemplazo. Si se necesitan, debería emplearse mejor una función del shell.

Para definir un alias utilizamos

```
alias [-p] [nombre[=valor] ...]
```

Por ejemplo,

```
$ alias ll="ls -laF"
```

conseguimos con sólo dos caracteres ("ll", mnemónico de Listado Largo, por ejemplo) realizar la misma función que con siete.

Para eliminar un alias utilizamos

```
unalias [-a] [name ...]
```

Por ejemplo, para eliminar el alias que habíamos generado anteriormente

```
$ unalias ll
```

➡ **Para practicar: Crear un alias que permita que funcione el comando `cd ..`.**

```
$ alias cd..="cd .."
```

Probar que funciona y eliminarlo después con

```
$ unalias cd..
```

### 15.1.7. Historia de órdenes.

Cuando se habilita la opción `-o history` (opción que ya está normalmente por defecto<sup>9</sup>), el shell da acceso a la historia de órdenes: lista de órdenes tecleadas con anterioridad.

El texto de los últimos mandatos (la variable `HISTSIZE` nos indica el número, por omisión 1000) se guarda en una lista de historia. El shell almacena cada orden en la lista de historia antes de la expansión de parámetros y variables. En el arranque, la historia se inicia a partir del fichero nombrado en la variable `HISTFILE` (por omisión `~/.bash_history`<sup>10</sup>). `HISTFILE` se trunca, si es necesario, para contener no más de `HISTFILESIZE` líneas.

Para visualizarlas:

```
$history
```

o mejor

```
$history | less
```

Podemos recorrer las órdenes anteriores con las teclas de flecha hacia arriba y flecha hacia abajo.

➡ **Para practicar**

1. Ejecutar el último comando `echo` con

```
$!e
```

2. Tras la salida del comando

```
$history
```

ejecutar

```
$!número_línea_comando_echo
```

### 15.1.8. Los Builtins (Órdenes internas)

Los *builtins* (u órdenes internas) son comandos que ya vienen implementados dentro de la propia bash. No hay que buscar un ejecutable externo porque la propia bash lo lleva incorporado. Por ello, se ejecutan mucho más rápido. Algunos de ellos son:

**cd:** Que como ya sabemos nos cambia de directorio de trabajo.

**pwd:** Que nos indica en qué directorio estamos situados.

---

<sup>9</sup>Podemos ponerla con `set -o history`

<sup>10</sup>Podemos ver con un editor el contenido de este fichero

## 15.2. Redirección

Antes de que se ejecute una orden, su entrada y salida pueden ser redirigidas usando una notación especial interpretada por el shell. Con ello podemos obtener la entrada a un comando desde una ubicación determinada o dirigir su salida hacia donde queramos.

- El descriptor de fichero número 0 se refiere a la entrada estándar (STANDARD INPUT), que para un terminal es el teclado.
- El descriptor de fichero número 1 se refiere a la salida estándar (STANDARD OUTPUT), que en un terminal es la pantalla.
- El descriptor de fichero número 2 se refiere a la salida de error estándar (STANDARD ERROR), que por omisión coincide con la salida estándar.

En las descripciones siguientes, si se omite el número del descriptor de fichero, y el primer carácter del operador de redirección es <, la redirección se refiere a la entrada estándar (descriptor de fichero 0). Si el primer carácter del operador de redirección es >, la redirección se refiere a la salida estándar (descriptor de fichero 1).

El orden de las redirecciones es significativo. Por ejemplo, la orden

```
$ ls >fichero_salida 2>&1
```

dirige la salida estándar normal a fichero\_salida y después dice que la salida de error estándar (2) vaya a donde va la salida estándar, que ya es fichero\_salida. Sin embargo, la orden

```
$ ls 2>&1 >fichero_salida
```

dirige solamente la salida estándar a fichero\_salida, porque la salida de errores estándar se ha redirigido antes hacia la antigua salida estándar (que en este caso puede ser la pantalla).

### 15.2.1. Redirección de la entrada (<)

La redirección de la entrada hace que el fichero cuyo nombre resulte de la expansión de "entrada" se abra para lectura en el descriptor de fichero n, o la entrada estándar (descriptor de fichero 0) si no se especificó n.

El formato general para la redirección de la entrada es:

```
[n]<entrada
```

Ejemplos:

```
$ cat </etc/passwd
```

El comando cat (que muestra por salida estándar lo que recibe por la entrada estándar) recibe como entrada el fichero /etc/passwd

```
$ mail root <mensaje_a_root
```

Con esta orden el comando mail recibe como entrada el fichero mensaje\_a\_root

### 15.2.2. Redirección de la salida (>)

La redirección de la salida hace que el fichero cuyo nombre resulte de la expansión de "salida" se abra para escritura en el descriptor de fichero n, o la salida estándar (descriptor de fichero 1) si n no se especificó. Si el fichero no existe se crea; si existe, se trunca a longitud cero; es decir, se crea vacío.

El formato general para la redirección de la salida es:

```
[n]>salida
```

Por ejemplo, si queremos ver los procesos que hay en un momento determinado en nuestro sistema y guardarlos en un fichero llamado procesos para su análisis, haremos:

```
$ps -aux >procesos
```

Posteriormente podremos ver el contenido del fichero procesos

### 15.2.3. Añadir a la salida redirigida (>>)

La redirección de la salida en esta forma hace que el fichero cuyo nombre resulte de la expansión de "salida" se abra para añadir en el descriptor de fichero `n`, o la salida estándar (descriptor de fichero 1) si no se especificó. La diferencia con respecto a la redirección de salida (`>`) es que no perdemos los datos del fichero en caso de que éste existiese. Si el fichero no existe, se crea.

El formato general para añadir a la salida es:

```
[n]>>salida
```

Por ejemplo, si tenemos un fichero en el que vamos guardando las salidas del comando `df` que nos muestra el consumo de disco en nuestro sistema para ver su incremento, podemos hacer

```
$df >>consumo_disco
```

y no perderemos los valores que vamos almacenando, sino que se van acumulando en el fichero.

#### ➡ Para practicar: Redirección

1. Al final de cada apartado, comprobar el resultado, por ejemplo con

```
$more prueba
```

- a) Crear un fichero de texto

```
$echo "Hola Mundo" >prueba
```

- b) Añadir al final "cruel"

```
$echo "cruel" >>prueba
```

- c) Almacenamos en `prueba` los ficheros (ocultos y directorios) de nuestro `$HOME`, en una columna y ordenados por tiempo de creación

```
$ls -alt >prueba
```

- d) ¿Y si lo ordenamos alfabéticamente?

```
$sort <prueba >prueba_o
```

- e) ¿Qué pasa ahora?

```
$more prueba >>prueba
```

- f) Para terminar

```
$cat prueba | grep X >prueba_x
```

2. ¿Qué pasa con los errores?. Ejecutemos

```
$ls /home/no_existe >$HOME/prueba
```

y obtendremos

```
ls: /home/no_existe: No existe el fichero o el directorio
```

- a) Comprobar que con:

```
$ls /home/no_existe >$HOME/prueba 2>&1
```

```
$cat prueba
```

```
ls: /home/no_existe: No existe el fichero o el directorio
```

esto significa que tenemos la salida y cualquier mensaje de error en el mismo lugar: fichero `~/prueba`

- b) Si escribimos

```
$ls /home/no_existe 2>&1 >$HOME/prueba
```

```
ls: /home/no_existe: No existe el fichero o el directorio
```

Así redirigimos a *stderr* al mismo lugar que *stdout* antes de haber redirigido a *stdout*, o sea que, los errores van a la pantalla pero *stdout* al archivo `~/prueba`.

- c) ¿Qué se obtendría de?

```
$ls ~/ 2>&1 >$HOME/prueba
```

## 15.3. Comandos de la Shell I

### 15.3.1. Comandos simples

Un comando simple es la clase de comandos que nos encontramos más frecuentemente. Consiste en una secuencia de palabras separadas por blancos. La primera palabra especifica el comando a ejecutar, seguido por unas opciones (como por ejemplo `ls -l`<sup>11</sup>) o unos argumentos (`cat /etc/profile`<sup>12</sup>).

### 15.3.2. Tuberías

Una tubería es una secuencia de una o más órdenes separadas por el carácter "|" (barra vertical). El formato de una tubería es:

```
orden1 [ | orden2 ... ]
```

La salida estándar de `orden1` se conecta a la entrada estándar de `orden2`. Esta conexión se realiza antes que cualquier redirección especificada por la orden.

Cada orden en una tubería se ejecuta como un proceso separado (esto es, en una subshell).

Por ejemplo, para contar el número de líneas de un fichero, ejecutaríamos:

```
$cat fichero | wc -l
```

Su explicación es que con el comando `cat` visualizamos el contenido del fichero, pero esta salida, en vez de ir a la pantalla, se mete en la tubería que va hacia la entrada de la orden `wc` (de *word count*, contador de palabras) que con su opción `-l` nos dice el número de líneas que ha leído.

En esta característica se apoya gran parte de la elegancia de los sistemas Unix/Linux. Con comandos simples podemos llegar a realizar acciones verdaderamente complejas.

### 15.3.3. Listas de comandos

Una lista es una secuencia de comandos simples o tuberías separados por uno de los operadores `;`, `&`, `&&`, or `||`, y terminada por `;`, `&`, o retorno de carro.

Si un comando se termina con el operador de control `&`, la shell ejecuta el comando de forma asíncrona en una subshell. Esto se conoce como ejecutar el comando en segundo plano (*background*). En este caso, la shell no espera a que el comando termine e inmediatamente aparece otra vez el indicador de inicio (prompt), mientras el comando se ejecuta por detrás de ella.

Los comandos separados por `;` se ejecutan secuencialmente, uno detrás de otro.

```
$ comando1; comando 2
```

La shell espera a que terminen los comandos en su turno correspondiente. Por ejemplo:

```
$ cd /home/Thales; ls
```

primero se posiciona en el subdirectorio `/home/Thales` y después lista los ficheros de ese directorio.

Los operadores de control permiten ejecuciones condicionales.

El efecto de

```
comando1 && comando2
```

es que `comando2` se ejecutará si y sólo si `comando1` termina de forma satisfactoria (devuelve un código de cero).

En cambio, en la lista

```
comando1 || comando2
```

el `comando2` se ejecutará si y sólo si `comando1` falla (devuelve un código distinto de cero).

#### ➡ Para practicar: comando `tee`

Podemos conseguir guardar la salida de un comando en un fichero y dirigirla a la salida estándar usando el comando `tee`. El nombre del comando viene de que se comporta como una T de fontanería. El caudal que llega por una rama, pasa por la T y sale por los otros dos orificios.

<sup>11</sup>En este caso la opción es `l`, y el guión sirve para indicar que lo que viene detrás es una opción. Una opción normalmente modifica el comportamiento de un comando.

<sup>12</sup>El fichero `/etc/profile` es un argumento. Los argumentos normalmente indican sobre qué actúa el comando.



Por ejemplo, supongamos que deseamos ver los usuarios de nuestra máquina y guardarlos en un fichero ordenados, escribiremos:

```
$ cut -f1 -d: /etc/passwd | sort | tee usuarios.txt | less
```

Explicuemos un poco el comando:

- `cut -f1 -d: /etc/passwd` → Obtiene del fichero `/etc/passwd` el primer campo (`f1` de *field1*), especificando como separador de campo (`-d` de delimitador) el carácter `:`
- `sort` → ordena alfabéticamente los nombres de usuario
- `tee usuarios.txt` → guarda el resultado en el fichero `usuarios.txt` y además lo dirige a la salida estándar. Son las dos salidas de la T.
- `less` → es un filtro que nos permite ver el resultado desplazándonos hacia adelante y hacia atrás. No es “obligatorio” ponerlo.

## Capítulo 16

# Comandos básicos de Unix/Linux

Como regla general, se podría decir lo siguiente: "Todo lo que se puede hacer en modo gráfico, se puede hacer también en modo texto, a base de comandos. Pero no todo lo que se puede hacer en modo texto, se puede hacer en modo gráfico". (*FAQ sobre Linux para principiantes* - es.comp.os.linux)

### 16.1. Introducción

En este apartado veremos los comandos más usuales de Linux. Ni están todos ni tiene sentido ver todas y cada una de las opciones de ellos. Para ampliar sobre algunos de ellos os remitimos a las páginas de ayuda de cada comando, a las infopages, así como a los manuales comentados en la primera entrega. Y sobre todo el uso que vayamos haciendo de ellos.

Ante la duda de si es necesario conocer los comandos la respuesta es clara: sí. Al menos los más usuales. Además, es necesario saber qué se puede hacer aunque a veces necesitemos la chuleta con la orden que nos permita saber cómo hacerlo. Si sólo nos dedicásemos a usar Linux como un entorno de oficina es posible que el número de comandos necesarios fuese mínimo, pero si deseamos administrar nuestro sistema Linux no queda más remedio que ampliar el conocimiento sobre ellos.

El tema sobre comandos se ha dividido en dos partes: por un lado tenéis una referencia rápida de qué hace cada uno. Por otro, se han analizado con más detalle aquellos que tienen más utilidad.



Recordar de nuevo la facilidad de uso que representa la autocompletación de comandos. Cuando queramos ejecutar un comando, no tenemos que conocer su nombre exacto ni el del fichero que le pasamos como parámetro para poder trabajar con él. Así, por ejemplo, si deseamos saber qué comandos comienzan por las letras `wh` escribiremos

```
$ wh  
  
whatis    whereis while    who    whois  
whatnow   which    whiptail whoami whom
```

y tras pulsar la tecla **[Tab]** dos veces, nos aparecerán las concordancias encontradas en nuestro `path`. Si la concordancia es única, se autocompletará el comando pulsando una sola vez la tecla.



Para “abrir” boca un mini resumen de la equivalencia entre los comandos más usuales del DOS y los de Linux

Descripción	DOS/Windows	Linux
Ayuda	help	man
Copiar ficheros	copy	cp
Contenido de un fichero	type	cat
Renombra un fichero	ren	mv
Mover ficheros/directorios	move	mv
Lista archivos	dir	ls
Borra archivos	del	rm
Borra la pantalla	cls	clear
Terminar una sesión	exit	exit
Crea un directorio	mkdir	mkdir
Borra un directorio	rmdir	rmdir
Cambiar de directorio	cd	cd
Cambiar atributos de ficheros	attrib	chmod
Cambiar la fecha	date	date
Compara ficheros	fc	diff
Memoria libre	mem	free
Imprimir un fichero	print	lpr
Editar un fichero	edit	mcedit
Mandar paquetes	ping	ping
Configuración interfaz de red	ipconfig	ifconfig
Configuración interfaz de red	winipcfg	ifconfig

### 16.1.1. Convenciones en cuanto a la sintaxis

La sintaxis común a todos los comandos es:

```
comando [opciones] [parámetro_1] parametro_2 ...
```

donde las opciones y los parámetros son opcionales si van entre corchetes (por ejemplo para el comando `free`<sup>1</sup>) e imprescindibles cuando van solos (por ejemplo para el comando `write`<sup>2</sup>). Si además algún parámetro va seguido de tres puntos suspensivos es para indicar que pueden incluirse cuantos parámetros de ese tipo se quieran.

Las opciones, en general se le pasan al comando como una serie de valores precedidos por un guión, por ejemplo:

```
$ df -h -l
Filesystem Size Used Avail Use% Mounted on
/dev/hdc1 2.3G 1.5G 754M 66% /
/dev/hdc5 1.5G 728M 740M 50% /datos
/dev/hda5 994M 318M 676M 32% /mnt/d
```

nos informa de la utilización del espacio en disco del sistema de ficheros. Al pasarle como opciones:

- h** (`--human-readable`) añade una letra indicativa del tamaño, como M para megabytes, a cada tamaño
- l** hace que se limite el listado a los sistemas de ficheros locales, no en máquinas remotas que pudieran estar montados por NFS, por ejemplo.

En general, esta forma de poner las opciones es equivalente a poner un solo guión y los valores de las opciones a partir del guión como una cadena de caracteres. Así la orden anterior es equivalente a escribir:

```
$ df -hl
```

<sup>1</sup>Si ejecutamos

```
$man free
```

veremos la sintaxis de este comando y comprobamos que todas las opciones y parámetros son opcionales.

<sup>2</sup>En este comando (`write`), al menos el argumento `user` es necesario.

### 16.1.2. Comodines

De igual manera que en sistemas DOS<sup>3</sup>, en Linux se puede hacer uso de comodines para hacer referencia a nombres de archivos, las posibilidades son:

\* igual que en sistemas DOS, el comodín se sustituye por cualquier cadena de caracteres

? la interrogación también tiene el uso habitual, se sustituye por cualquier carácter, pero sólo uno.

[..] El uso de corchetes permite hacer referencia a un solo carácter, las posibilidades son:

- hacer referencia a un solo carácter pero con la obligatoriedad de estar comprendido en los valores listados entre corchetes:

```
$ ls ed[89]linux
```

en este caso se mostrarían los ficheros cuyo nombre sea de la forma ed9linux o ed8linux

- hacer referencia a un rango de valores separados por un guión:

```
$ ls ed[7-9]linux
```

en esta caso se mostrarían todos los ficheros cuyo nombre fuese de la forma ed7linux, ed8linux o ed9linux.

- invertir el rango anteponiendo el signo !

```
$ ls ed[!1-8]linux
```

en este caso se mostrarán todos los ficheros con tercer carácter arbitrario y distinto de los números 1 al 8 (ambos inclusive)

Se pueden mezclar entre ellos, así:

```
$ ls ed?[7-9]*
```

mostraría todos los ficheros cuyo nombre de fichero verifique:

1. Sus dos primeros caracteres son “ed”
2. El tercer carácter puede ser cualquiera
3. El cuarto carácter es un número comprendido entre 7 y 9
4. El resto de caracteres pueden ser cualesquiera

## 16.2. Resumen comandos

### 16.2.1. Ayuda

**apropos** Busca las páginas de ayuda que contienen la clave que especifiquemos

**info** Permite el acceso a la ayuda *online* de un comando

**man** Para visualizar las páginas man

**whatis** Busca palabras completas en la base de datos *whatis*

viaturas ya definidas

**type** Indica cómo interpretaría la shell el comando pasado como argumento

**unalias** Para eliminar las abreviaturas que previamente hemos definido con *alias*

### 16.2.2. “Construir” comandos

**alias** Se usa para definir abreviaturas para los comandos largos. También nos muestra una lista con las abre-

### 16.2.3. Gestión de usuarios y grupos

**chgrp** Cambia el grupo de un archivo

**chmod** Cambia los permisos de acceso de ficheros

<sup>3</sup>Más bien DOS lo tomó de Unix. Recordemos que Unix es de finales de los años 60.





**chown** Cambia el usuario y grupo propietarios de ficheros

**groups** Muestra los grupos en los que está un usuario

**groupadd** Crea un nuevo grupo

**groupdel** Borra un grupo

**newgrp** Para pasar a tener los derechos de un grupo

**passwd** Para asignarle la contraseña a un usuario

**umask** Establece la máscara de creación de ficheros

**useradd** Para añadir un usuario

**userdel** Permite eliminar un usuario

#### 16.2.4. Manipulación de archivos y directorios

**cd** Cambia el directorio de trabajo

**cp** Copia ficheros y directorios

**file** Determina el tipo de un fichero

**ls** Nos muestra el contenido de un directorio (**dir**, **vdir** son versiones de **ls**)

**ln** Permite crear enlaces entre ficheros

**mkdir** Crea directorios

**mv** Mueve (renombra) ficheros

**rm** Borra ficheros o directorios

**rmdir** Borra directorios vacíos

**pwd** Muestra el nombre del directorio de trabajo actual

**touch** Actualiza la fecha de un archivo a la actual

#### 16.2.5. Localización de archivos

**find** Busca ficheros en un árbol de directorios

**locate** Permite localizar archivos basándose en una base de datos que se va actualizando periódicamente

**whereis** Localiza los ficheros binarios, fuentes y páginas del manual correspondientes a un programa

**which** Muestra el **path** del archivo de comandos pasado como argumento

#### 16.2.6. Procesamiento de archivos

**cat** Concatena archivos y también muestra su contenido usando la salida estándar

**cmp** Compara dos archivos

**csplit** Divide un archivo en secciones determinadas por líneas de contexto

**cut** Imprime secciones de líneas de un archivo de entrada

**dd** Convierte y copia un fichero

**diff** Busca diferencias entre dos archivos o directorios

**expand** Convierte las tabulaciones en espacios

**fold** Permite ajustar las líneas de texto al ancho que especifiquemos

**grep**, **egrep**, **fgrep** Muestran líneas de ficheros que concuerdan con un patrón

**head** Muestra la parte inicial de un archivo (por defecto 10 primeras líneas)

**less** Muestra archivos en pantalla de una vez paginando la salida, permite volver atrás

**more** Filtro que muestra un archivo pantalla a pantalla (es mejor **less**)

**nl** Numera las líneas de un archivo que no estén en blanco

**paste** Combina líneas de ficheros

**patch** Aplica el comando **diff** actualizando el archivo original. Aplica un "parche"

**sed** Editor de texto no interactivo

**sort** Ordena las líneas de archivos de texto

**split** Divide un archivo en varias partes (por defecto de 1000 líneas en 1000 líneas)

**tac** Invierte el orden de las líneas de un archivo. Cat al revés.

**tail** Muestra las últimas líneas (10 por defecto) de un documento

**tr** Cambia unos caracteres por otros

**uniq** Borra las líneas duplicadas de un archivo ordenado

**wc** Muestra el número de bytes, palabras y líneas de un archivo

**xargs** Construye y ejecuta órdenes desde la entrada estándar

**zcat** Igual que **cat** pero sobre ficheros comprimidos

**zless** Actúa como **less** pero sobre archivos comprimidos

**zmore** Igual que **more** pero sobre ficheros comprimidos

### 16.2.7. Guardar y comprimir ficheros

**compress** Comprime (o expande) archivos

**gunzip** Expande ficheros

**gzip** Comprime/expande ficheros

**tar** Para empaquetar y desempaquetar archivos y directorios

**uncompress** Expande archivos

**bzip2** Comprime ficheros con una ratio mejor que los anteriores

**bunzip2** Descomprime ficheros comprimidos con bzip2

### 16.2.8. Procesos de control

**at** Permite planificar la ejecución de tareas

**bg** Permite ejecutar un proceso interrumpido que está en segundo plano

**cron** Para planificar órdenes o procesos de forma periódica en el tiempo

**fg** Sigue con un proceso interrumpido anteriormente, pero en primer plano

**free** Muestra la cantidad de memoria libre y usada en el sistema

**halt** Cierra el sistema

**jobs** Lista la tabla de trabajos en ejecución

**kill** Termina un proceso

**ldd** Nos muestra las librerías compartidas que necesitamos para ejecutar un programa

**nice** Ejecuta un programa con la prioridad de planificación modificada

**ps** Informa del estado de los procesos

**printenv** Imprime parte o todo el entorno

**pstree** Proporciona un árbol de los procesos en ejecución

**reboot** Reinicia el sistema

**shutdown** Cierra el sistema

**sync** Vuelca a disco los *buffers* del sistema de archivos

**uname** Imprime información del sistema

### 16.2.9. Control de usuarios

**chfn** Cambia los datos de un usuario

**chsh** Cambia el shell

**groups** Imprime los grupos en los que está un usuario

**id** Muestra los identificadores de usuario y de grupo

**last** Muestra los últimos accesos al sistema

**passwd** Cambia contraseñas

**su** Ejecuta una shell con identificadores de grupo y de usuario distintos

### 16.2.10. Administrar ficheros

**df** Informa de la utilización del espacio de disco en sistemas de ficheros

**du** Lista el espacio ocupado por los archivos o directorios

**fdformat** Formatea un disquete

**fdisk** Manipulador de tablas de particiones para Linux

**fsck** Chequea y repara un sistema de archivos de Linux

**mkfs** Construye un sistema de ficheros de Linux

**mknod** Crea ficheros especiales de bloques o caracteres

**mkswap** Construye un área de intercambio para Linux

**mount** Monta un sistema de ficheros

**swapoff** Deshabilita dispositivos o ficheros de intercambio

**swapon** Habilita dispositivos o ficheros de intercambio

**tty** Imprime el nombre del fichero del terminal conectado a la entrada estándar

**umount** Desmonta sistemas de ficheros

### 16.2.11. Comunicaciones y redes

Se dejan para la última entrega completar y estudiar: ftp, telnet, ssh, route, ifconfig, netstat, etc.

**finger** Proporciona información sobre los usuarios conectados al sistema

**mail** Programa destinado al envío y recepción de correo

**mesg** Permite permutar la posibilidad de recibir mensajes de otros usuarios

**talk** Permite establecer una “charla” con otro usuario

**wall** Manda un mensaje o un archivo a todos los usuarios que admitan mensajes con `write`

**w** Muestra qué usuarios están conectados y qué están haciendo

**who** Muestra información de los usuarios conectados al sistema

**write** Manda un mensaje a la pantalla de un usuario

### 16.2.12. Comandos de Impresión

**lpq** Muestra los trabajos en la cola de impresión

**lpr** Envía un trabajo a la impresora o pone en cola un trabajo de impresión

**lprm** Elimina un trabajo de la cola

**lpstat** Permite comprobar el estado de los trabajos de impresión

### 16.2.13. Módulos del kernel

**depmod** computa las dependencias entre módulos

**lsmod** lista los módulos activos

**insmod** carga un módulo en el kernel

**rmmod** descarga un módulo cargable

### 16.2.14. Varios

**cal** Calendario

**clear** Borra la pantalla

**date** Proporciona o ajusta la fecha y hora del sistema

**dmesg** Permite ver los mensajes de inicio del sistema

**echo** Muestra el texto/contenido de la variable

**env** Muestra el entorno actual de trabajo con todas sus variables

**exit** Cierra el shell actual

**nohup** Permite que un comando se ejecute aunque se cierre la sesión, y sin salida a un `tty`

**time** Tiempo que tarda en ejecutarse un comando

## 16.3. Algunos ejemplos

En esta sección vamos a ver varios ejemplos de cómo se utilizan algunos de los comandos anteriores. Hay grupos de comandos del resumen anterior que ya se han visto, y, por tanto, no se ponen de nuevo aquí; entre ellos estarían los comandos de impresión, los de gestión de usuarios, ayuda, etc.

Para algunos comandos hemos seguido el convenio de poner:

**comando:** `sintaxis_usual`

### 16.3.1. “Construir” comandos

En el capítulo sobre la Shell Bash ya se ha visto y comentado el funcionamiento de estos comandos. Retomemos algunos aspectos más sobre ellos.

#### alias

```
alias [-p] [nombre[=valor] ...]
```

#### ➡ Para practicar

Un alias es la posibilidad que tenemos de usar un nombre diferente para cualquier comando. Uno de los usos más “típicos” del comando `alias` consiste en definir en el fichero `/etc/bashrc` la serie de “alias”

```
alias ls='ls --color -sF'
alias dir='dir --color -Fs'
alias vdir='vdir --color -s'
alias rd='rmdir'
alias md='mkdir'
alias cd..='cd ..'
```

así, cuando ejecutemos el comando `ls`, `dir` o `vdir` veremos los ficheros/directorios de distintos colores, podremos usar ahora `rd` como sinónimo de `rmdir`, `md` de `mkdir` y `cd..` de `cd ..`<sup>4</sup>. Antes de ponerlos en el fichero `/etc/bashrc` podemos practicar con estos alias (y con otros) desde la línea de comandos, así escribiríamos:

```
$ alias ls='ls --color -sF'
```

si ahora ejecutamos

```
$ ls
```

comprobaremos que, dependiendo de qué tipo de fichero estemos considerando, se ve de distinto color:



```
[root@novo /ppp]# ls
total 16
1 Thalesa          1 ip-up*          1 options
1 arrakis          1 ip-up.local*    0 pap-secrets@
1 averroes         1 a_Thalesa*      1 /
1 chap-secrets     1 a_arrakis*      1 uni2
2 connect-errors   1 a_averroes*
1 ip-down*         1 a_uni2*
```

## type

type comando

Si ahora ejecutamos

```
$ type ls
```

obtendremos:

```
ls is aliased to `ls --color -Fs`
```

## unalias

unalias nombre\_alias...

Por último, con `unalias` podemos quitar alias, así si ejecutamos

```
$ unalias ls
```

y después

```
$ type ls
```

obtendremos:

```
ls is hashed (/bin/ls)
```

es decir, `ls` se quedaría con las opciones que tiene por defecto.

## 16.3.2. Manipulación de archivos y directorios

La mayoría de los comandos que aparecen en este grupo son conocidos ya por los que venimos del MSDOS, lo que ocurre es que puede que se nos haya olvidado su nombre completo<sup>5</sup>. Otros tienen un uso tan inmediato que con el pequeño resumen de su función consideramos que es suficiente.

### cd

```
cd [directorio]
```

Retomemos a nuestro linuxero THALES, que se encuentra trabajando en su directorio de usuario `/home/Thales`. Nuestro usuario THALES tiene que moverse por el árbol de directorios y desplazarse al directorio raíz, para ello ejecuta:

```
$ cd /
```

<sup>4</sup>Puede que sobre el papel no se vea muy claro, lo que ocurre es que en Linux hay que dejar un espacio entre el `cd` y los dos puntos, esto en DOS no importa. Al usar este alias da igual que dejemos el espacio o no.

<sup>5</sup>¿quién se acuerda ya del `edlin`?

Después se mueve a

```
$ cd /etc/sysconfig
```

para ver el contenido de un fichero. Una vez terminada la labor, vuelta a casa

```
$ cd
```

y listo, el sistema lo lleva a /home/Thales.

Pero siempre se olvida algo, necesita volver al directorio en el que se encontraba anteriormente (/etc/sysconfig) y ejecuta:

```
$ cd -
```

## cp

```
cp [opciones] fuente destino
```

Una vez en el raíz, recuerda que tiene que copiar el fichero /home/Thales/curso/entrega\_3.sgml al subdirectorio /ed03linux, para hacer esto escribe:

```
$ cp /home/Thales/curso/entrega_3.sgml /ed03linux
```

## file

```
file archivo...
```

Como no recuerda con qué aplicación lo hizo escribe:

```
$ file /ed03linux/entrega_3.sgml
```

y ve en el terminal que:

```
entrega_3.sgml: gzip compressed data, deflated, original filename,  
last modified: Sun Feb 20 20:46:19 2000, max compression, os: Unix  
con lo que recuerda que ese fichero lo comprimió y que antes de hacer nada con él debe descomprimirlo,
```

tras hacerlo (véase 16.3.5), ejecuta de nuevo:

```
$ file /ed03linux/entrega_3.sgml
```

y el resultado ahora es:

```
entrega_3.sgml: exported SGML document text  
es decir, es un documento en formato sgml.
```

## ls

```
ls [opciones] [archivo, directorio]
```

Quizás, junto con `cd`, el comando más usado en Linux sea `ls` (o alguna de sus variantes), `ls` muestra el contenido de un directorio en un listado que por defecto está ordenado alfabéticamente. La sintaxis básica es:

```
$ ls [opciones] [archivo, directorio]
```

donde:

**opciones** las más importantes son:

- a** Muestra todos los archivos (hasta los “ocultos”, los que empiezan por “.”)
- f** Muestra el contenido de los directorios en el orden en el que están almacenados en el disco.
- i** Muestra el inodo de los archivos listados.
- m** Lista los directorios separando los nombres por comas.
- r** Invierte el orden usual de mostrar el directorio
- s** Muestra el tamaño de los archivos.
- t** Ordena los archivos por fecha de creación, primero los más recientes.

**R** Muestra recursivamente el directorio y sus subdirectorios.

Ya hemos visto anteriormente algunos ejemplos de este comando.

### **mkdir**

```
mkdir [-p6] directorio...
```

Continuemos con THALES. Ahora tiene que crear un nuevo directorio en el subdirectorio /ed03linux; tras situarse en él escribe:

```
$ mkdir entrega_4
listo, ya tiene su flamante directorio,
```

### **mv**

```
mv [-i7] origen destino
```

pero necesita mover el fichero anteriormente copiado a este nuevo directorio y entonces usa:

```
$ mv entrega_3.sgml entrega_4/
```

### **rm**

```
rm [opciones] archivo
```

Como ya no necesita el fichero original (estaba en /home/Thales/curso) decide borrarlo:

```
$ rm /home/Thales/curso/entrega_3.sgml
```

### **rmdir**

```
rmdir directorio...
```

Se da cuenta de que ni ese directorio (/home/Thales/curso/) ni su contenido los va a usar más y decide borrarlo, para esto escribe:

```
$ rmdir /home/Thales/curso/
```

y recibe un error del sistema, y es que `rmdir` sólo borra directorios vacíos (¿qué se la va a hacer? a grandes males...), así que escribe:

```
$ rm -r /home/Thales/curso/
```

y listo, ha borrado el directorio curso y todos los archivos, directorios y subdirectorios contenidos en él.<sup>8</sup>

➡ **Para practicar: Probar el uso de los comandos anteriores.**

## **16.3.3. Localización de archivos**

Con `mc` o con `gnome-search-tool` la búsqueda de ficheros está “tirada”.

---

<sup>6</sup>Con esta opción crea los directorios intermedios en caso de que no existan

<sup>7</sup>Pregunta antes de sobrescribir un archivo de destino que ya exista.

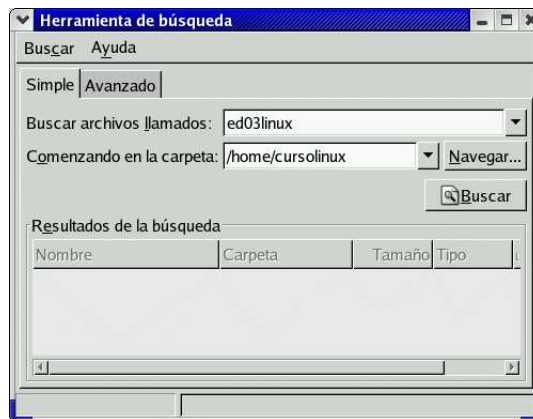
<sup>8</sup>Cuidado con esta orden. La “r” viene de Recursivo ¿os imagináis qué pasaría si como root escribís?:

```
# rm -r /
```

Incluso si en un directorio cualquiera ejecutamos como root la siguiente orden, pensando en eliminar los ficheros ocultos que empiezan por punto:

```
# rm -r .*
```

Recordaremos la película del aprendiz de brujo en la que las escobas no paraban de traer cubos de agua y desearemos pararlo como sea, aunque casi siempre demasiado tarde. Lo que ha ocurrido es que una de las expansiones de `.*` será el fichero `..` que es precisamente el directorio superior. ¡¡¡Socorro, ayuda!!!



## locate

```
locate patrón
```

Otra forma de buscar ficheros es usar el comando `locate` junto con `updatedb`. `updatedb`<sup>9</sup> actualiza una base de datos de nombres de ficheros, para actualizar la base de datos de ficheros escribiremos:

```
# updatedb &
```

ejecutamos el comando como root y además en segundo plano, ya que tiene que localizar y almacenar todos los nombres de todos los ficheros y esto hace que su finalización no sea inmediata. Al ejecutar el comando en segundo plano, podremos seguir trabajando en nuestro terminal mientras que se ejecuta. Tan sólo tendremos que actualizar la base de datos de ficheros cada vez que se haya producido un cambio sustancial.

Cuando queremos buscar cualquier fichero que esté en la base de datos generada usaremos

```
$ locate patron
```

y en pantalla se nos mostrarán todas las concordancias con ese `patron`.

Por ejemplo, si ya hemos actualizado la base de datos de ficheros:

```
$ locate fracta
/usr/share/icons/mini/mini-fractal.xpm
/usr/share/wallpapers/fractal_nebula.jpg
```

## find

```
find [camino...] [expresión]
```

Veamos algunos más. Con `find` podemos encontrar archivos basando su búsqueda en distintas características de los mismos. Su sintaxis básica es:

```
find [camino...] [expresión]
```

El número de opciones de `find` es muy elevado (`$man find`)

### ➡ Para practicar: `find` y `locate`

#### 1. Uso de `find`

- Encuentra todos los archivos que hay en el directorio actual y en sus subdirectorios con extensión `.txt`

---

<sup>9</sup>Este comando se ejecuta una vez al día. El comando `cron` (se ve un poco más adelante) se encarga de que a diario la base de datos esté actualizada (véase el fichero `/etc/cron.daily/slocate.cron`)

Este es uno de los motivos de que a veces (sobre todo si no usamos linux tanto como debiéramos) el sistema parezca lento. Es que se están ejecutando procesos en segundo plano. Podemos notar que el disco duro comienza a trabajar sin saber por qué.

```
$ find . -name "*.txt"10
```

- Busca los archivos que hay en el directorio actual y en sus subdirectorios a los que se haya accedido en los últimos 10 minutos

```
$ find * -amin 10
```

- Encuentra los ficheros con permisos 777

```
$ find * -perm 77711
```

2. Localizar con `locate` los archivos de nombre `internet/Internet`, para eso hemos de añadir el parámetro `-i`  

```
$locate -i internet
```

### which

`which` comando

Si queremos conocer el path completo de un determinado comando, como por ejemplo de `startx`, usaremos:

```
$ which startx
```

la respuesta sería:

```
/usr/X11R6/bin/startx
```

### whereis

`whereis` comando

si no nos basta con esta información y, además, queremos saber qué páginas del manual acompañan al programa escribiremos:

```
$ whereis startx
```

el resultado es:

```
startx: /usr/X11R6/bin/startx /usr/X386/bin/startx /usr/bin/X11/startx
```

```
/usr/X11/bin/startx /usr/X386/man/man1/startx.1x /usr/X11/man/man1/startx.1x
```

### ➡ Para practicar

1. Buscar el path del comando `ls` y las *páginas de manual* de este comando.
2. Como un usuario normal hacer lo mismo con el comando `fdisk` ¿por qué `which` no lo encuentra?
3. Usando el comando `su` pasar a ser root y hacer de nuevo el apartado anterior ¿por qué `which` sigue sin encontrarlo?
4. Por último, entrar como root (o ejecutar el comando `$su -12`) y comprobar que ahora sí que “funciona” bien.

## 16.3.4. Procesamiento de archivos

Con respecto a las órdenes para procesar archivos lo idóneo es saber que están ahí y las posibilidades que tenemos con ellas. Algunas son bastante especializadas y en general su uso puede ser escaso. Os remitimos a las manpages para profundizar en ellas en el caso de necesitarlas.

<sup>10</sup>Las dobles comillas son necesarias para que no interprete el asterisco. Una explicación sobre el uso de las distintas comillas está en 17.5.1 en la página 252.

<sup>11</sup>En estos dos casos, el asterisco significa el camino y si queremos que se expanda, por eso no va entre comillas.

<sup>12</sup>El guión (-) hace que se cargue el entorno completo del usuario root. Como ya vimos, los ficheros `/etc/profile`, `/etc/bashrc...`



**cat, less**

Las que más se usan son `cat`, `less` (ya ha aparecido en su uso más estándar). La forma más común de usar `cat` es para mostrar en pantalla un fichero de texto (como el `type` del DOS). Así, si queremos ver un fichero de pocas líneas escribiremos:

```
$ cat fichero
```

`Less` es bastante más interesante y como ya se ha dicho la forma más usual de utilizarlo es en una tubería, ya ha salido varias veces en el curso. Por ejemplo si el fichero de antes es largo lo mejor sería escribir:

```
$ less fichero.txt
```

y tendremos la posibilidad de verlo página a página y de echar hacia atrás.

**dd**

Con respecto a `dd` comentar que nos permite (entre otras cosas) crear discos de instalación de Linux sin tener que usar `rawrite` ni MSDOS, la forma de usarla es:

```
# dd if=/mnt/cdrom/images/fichero_img.img of=/dev/fd0
```

claro está, en el supuesto de que el fichero `.img` se encuentre en el subdirectorio `/mnt/cdrom/images` y que se llame `fichero_img.img`



**➡ Para practicar.**

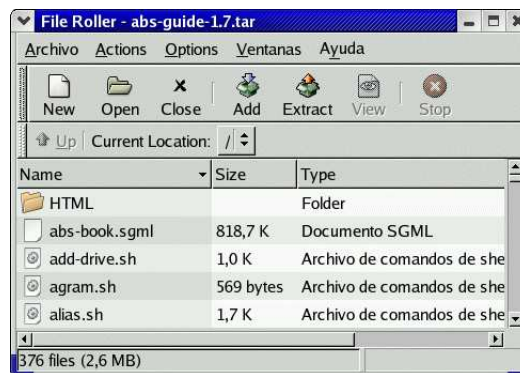
Crear un disco de instalación vía red usando el CD1:

```
#dd if=/mnt/cdrom/images/bootnet.img of=/dev/fd0
```

**16.3.5. Empaquetando y comprimiendo ficheros.**

Linux dispone de múltiples utilidades y programas para comprimir<sup>13</sup> y descomprimir ficheros, desde el clásico `compress` hasta el más moderno `bzip2`<sup>14</sup>.

 Recordemos que en modo gráfico disponemos de `File Roller` ( → **Accesorios** → **File Roller**)



<sup>13</sup>

- En los CDs de la distribución hay dos paquetes que permiten comprimir y descomprimir ficheros con compatibilidad con MSDOS, son:

```
unzip-5.50-5.i386.rpm
```

```
zip-2.3-14.i386.rpm
```

Se instalan por defecto.

- Los paquetes `rar` y `unrar` (disponibles en <http://rpmfind.net>) permiten trabajar con este formato de compresión de ficheros (`rar`).

<sup>14</sup>Se instala por defecto



Su uso es inmediato y no debe suponer ningún problema.

Sin embargo los más utilizados (hoy por hoy) son:

**tar** para empaquetar

**gzip** para comprimir

**tar**

`tar` permite empaquetar o desempaquetar ficheros. El concepto de empaquetar aquí es el de meter varios ficheros y/o directorios en un solo fichero. Posteriormente podremos recuperar esa estructura de ficheros y directorios en el lugar donde queramos. Su sintaxis básica es<sup>15</sup>

```
$ tar opciones archivo.tar [origen]
donde:
```

**opciones** vamos a analizar:

**c** para crear archivos empaquetados

**x** para expandir archivos empaquetados

**t** para mostrar el contenido de un fichero tar empaquetado

**v** almacenamos/visualizamos la información en forma detallada

**f** para indicar que `archivo.tar` es un fichero.

**z** filtrar el archivo a través de `gzip` (tanto para comprimir como descomprimir).

**M** para crear/desempaquetar usando varios discos.

**archivo** nombre del archivo

**origen** nombre<sup>16</sup> del directorio/fichero o directorios/ficheros a empaquetar separados por espacios

Supongamos que deseamos empaquetar dos ficheros llamados `ed03linux1.txt` y `ed03linux2.txt`, en un fichero tar de nombre `ed03linux.tar`, escribiremos:

```
$ tar -cvf ed03linux.tar ed03linux1.txt ed03linux2.txt
```

también podíamos escribir:

```
$ tar -cvf ed03linux.tar ed03linux?.txt
```

Si lo que queremos es empaquetar un subdirectorio, todo igual. Supongamos que deseamos empaquetar el contenido de la 3ª entrega del curso y que esta entrega se encuentra en el directorio `entrega_3`, la sintaxis sería:

```
$ tar -cvf entrega_3.tar entrega_3
```

Si lo que deseamos es desempaquetar un fichero tar, en vez de escribir la opción `-c` escribiremos `-x`, así para desempaquetar el contenido de la entrega anterior escribiremos:

```
$ tar -xvf entrega_3.tar
```

Si solamente queremos ver el contenido del fichero empaquetado (tar), ejecutaremos.

```
$ tar -tvf entrega_3.tar
```

---

<sup>15</sup>Para ver todas la opciones de la orden `tar`:

```
$ man tar
```

<sup>16</sup>Podemos usar comodines

## gzip

Gzip permite comprimir ficheros, la sintaxis básica es:

```
$ gzip [opciones] archivo
```

dónde:

**opciones** vamos a analizar:

**d** para descomprimir archivos

**t** para comprobar que la compresión se ha realizado con éxito

**1-9** nivel de compresión, el 1 indica menor ratio y mayor rapidez, el 9 daría como resultado un archivo más pequeño pero un mayor tiempo de compresión. El nivel por defecto es 6

**archivo** nombre del archivo

Ya tenemos empaquetado nuestro archivo `entrega_3` y deseamos comprimirlo al máximo y verificar que todo está bien, usaremos:

```
$ gzip -9 entrega_3.tar
```

el resultado es el fichero `entrega_3.tar.gz`, si queremos comprobar la “integridad” del fichero escribiremos:

```
$ gzip -tv entrega_3.tar.gz17
```

```
entrega_3.tar.gz: OK
```

Si ahora queremos descomprimir este fichero tenemos dos opciones, bien usamos:

```
$ gzip -d entrega_3.tar.gz
```

bien escribimos directamente

```
$ gunzip entrega_3.tar.gz
```

**Los dos a la vez: tar y gzip** Cabe la posibilidad de empaquetar y comprimir un directorio directamente sin tener que usar un comando tras otro, una posibilidad consiste<sup>18</sup> en escribir:

```
$ tar -czvf nombre_fichero.tar.gz origen
```

o bien

```
$ tar -czvf nombre_fichero.tgz origen
```

en ambos casos, la opción `-z` es la que señala que vamos a comprimir. Como podemos comprobar, en este caso la extensión es `.tgz`, y es que ambas son equivalentes: `.tar.gz` y `.tgz`.

También podemos descomprimir y desempaquetar un fichero `tar.gz` o un `tgz` usando sólo la orden `tar`, la sintaxis sería:

```
$ tar -xzvf fichero.tar.gz
```

o bien

```
$ tar -xzvf fichero.tgz
```

## ➡ Para practicar: “targz”

El objetivo de esta práctica reside en aprender a empaquetar/comprimir directorios. Vamos a trabajar con el subdirectorio `/usr/share/doc/HTML` (documentación del KDE).



Notar que se supone que estamos trabajando como un usuario normal y no como el root. Un error en la sintaxis del último comando puede ser desastrosa para el sistema.

<sup>17</sup>La `v` de “verbose”, es decir, para que me muestre más información en el terminal de cómo ha ido el proceso de testeo.

<sup>18</sup>Sin usar tuberías

1.

```
$tar -cvf ~/kde_html.tar /usr/share/doc/HTML
$ll kde_html.tar                (es enorme)
$gzip kde_html.tar
$ll kde*                        (ya de tamaño mejor)
```

2. Ahora de un tirón

```
$tar -czvf ~/kde_html.tgz /usr/share/doc/HTML
$ll kde*
¿Salen del mismo tamaño?
```

3. Descomprimirlo de una pasada:

```
$tar -xzf kde_html.tgz
Comprobar que todo ha salido bien
```

4. Ahora con formato zip:

```
$zip -r ~/kde_html.zip /usr/share/doc/HTML
¿Cuál comprime mejor?
```

5. Por último, borremos el directorio creado:

```
$rm -r ~/usr
```

**Copias de seguridad multivolumen** Podemos usar `tar` para hacer copias de seguridad en varios discos, en ese caso la sintaxis sería:

```
$ tar cvfM /dev/fd0 origen
```

hay que hacer notar que en este caso no se puede empaquetar y comprimir a la vez y no podemos poner el guión delante de las opciones, no funciona.

Para desempaquetar los discos así obtenidos tan sólo tendremos que introducir el primer disco y, sin montarlo, ejecutar:

```
$ tar xvfM /dev/fd0
```

se nos pedirán los discos uno a uno hasta que termine el proceso.

### ➡ Para practicar: split

1. Crear, usando `tar`, una copia multivolumen del fichero `/usr/X11R6/bin/Xprt`
2. Veamos otra forma de partir un fichero grande y poder llevárnoslo en disquetes, para ejemplificarlo trabajaremos con el fichero del CD2 `gimp-1.2.3-9.i386.rpm`.

Lo primero que haremos será copiarlo a nuestro `$HOME` de usuario:

```
$ mount /mnt/cdrom
$ cp /mnt/cdrom/RedHat/RPMS/gimp-1.2* ~/
```

Trabajemos el fichero, lo vamos a hacer “cachitos” de 1400K de nombre `gimp`

```
$split -b 1400k gimp-1.2.3-9.i386.rpm gimp-
```

Podemos ver que hemos hecho con:

```
$ ll gimp*
total 15460
-rw----- 1 cursolinux cursolinux 8431167 mar 1 21:07 gimp-1.2.3-9.i386.rpm
```

```
-rw-rw-r-- 1 cursolinux cursolinux 1433600 mar 1 21:07 gimp-aa
-rw-rw-r-- 1 cursolinux cursolinux 1433600 mar 1 21:07 gimp-ab
-rw-rw-r-- 1 cursolinux cursolinux 1433600 mar 1 21:07 gimp-ac
-rw-rw-r-- 1 cursolinux cursolinux 1433600 mar 1 21:07 gimp-ad
-rw-rw-r-- 1 cursolinux cursolinux 1433600 mar 1 21:08 gimp-ae
-rw-rw-r-- 1 cursolinux cursolinux 1263167 mar 1 21:08 gimp-af
```

tenemos 6 ficheros (gimp-aa .. gimp-af) que juntos forman el paquete original. Para copiarlos a discos podemos:

o bien, usar el comando `mcopy`

```
#mcopy gimp-aa a:
```

o bien, tras montar el floppy, usar el comando `cp`

```
#mount /mnt/floppy
#cp gimp-aa /mnt/floppy
#umount /mnt/floppy
```

Una vez en la máquina de destino y tras copiar los 6 ficheros en algún directorio, con

```
$cat gimp-a* >gimp-1.2.3-9.i386.rpm
```

obtenemos el fichero original.

### compress, uncompress

Los ficheros comprimidos con `compress` tienen extensión `.Z`, para comprimirlos se usa el programa `compress` y para descomprimirlos `uncompress`. Es un formato poco utilizado en este momento y, además, la utilidad `gunzip` permite descomprimirlos. Así, para descomprimir un fichero comprimido con `compress` podemos usar:

```
$ uncompress archivo.Z
o bien
$ gunzip archivo.Z
```

### bzip2, bunzip2

Para comprimir y descomprimir ficheros existen más herramientas que las ya comentadas pero la única que merece mención especial es `bzip`. Con la instalación básica del sistema RedHat se nos instaló el paquete:

```
bzip2-1.0.2-5.i386.rpm
```

con él disponemos de dos utilidades que permiten comprimir y descomprimir ficheros con este formato<sup>19</sup>. La extensión de este tipo de ficheros es `.bz2`, para comprimir un fichero escribiremos<sup>20</sup>

```
$ bzip2 fichero
```

y para descomprimir un fichero:

```
$ bunzip2 fichero.bz2
```

---

<sup>19</sup>Es un formato con mejores ratios de compresión que los que ofrece `gzip`. `bzip2` comprime ficheros utilizando el algoritmo de compresión de texto por ordenación de bloques de Burrows-Wheeler.

<sup>20</sup>Para ampliar es mejor consultar la manpage que acompaña al programa

### ➡ Para practicar

Comprimamos otra vez el subdirectorio `/usr/share/doc/HTML`, pero ahora con:

```
$tar -cjvf ~/kde_html.tar.bz2 /usr/share/doc/HTML21
```

¿Cuál es el más comprimido?, ¿el `.zip`, el `.tgz` o el `.tar.bz2`?

Para descomprimirlo desempaquetarlo:

```
$bunzip kde_html.tar.bz2
```

```
$tar -xf kde_html.tar
```

¿Cómo se haría con un sólo comando?

Borremos toda la “basura” generada:

```
$rm kde*
```

```
$rm -r HTML
```

### 16.3.6. Procesos de control

Los comandos `jobs`, `bg` y `fg` se comentan en el capítulo sobre la Shell Bash II, sección Control de Trabajos ( 17.1 en la página 243 ).

#### at

```
at hora [fecha] lista_comandos
```

Comenzaremos con el comando `at`. `at` permite planificar la ejecución de tareas, permite que le especifiquemos tanto la fecha como la hora para activarse. Una vez activo `at` se encargará de hacer ejecutar las órdenes programadas (órdenes no interactivas). Por ejemplo supongamos que son las 3 h pm y hemos quedado a las 4 h pm, somos tan despistados que cuando nos ponemos con el ordenador se nos olvida todo, en ese caso podemos decirle a `at` que nos avise dentro de una hora escribiendo:

```
$ at now +60 minutes22
```

tras pulsar intro podremos escribir aquello que consideremos oportuno, por ejemplo:

```
at>echo "No te despistes, tienes una cita"
```

```
at>cdp23
```

cuando terminemos de escribir pulsaremos `[Ctrl]+[d]`. Introduciremos un cd de música en nuestro lector de cd y a trabajar.

A las cuatro `at` nos enviará un correo con el texto anterior que podremos visualizar con la orden `mail`. Y, además, si tenemos un cd de música comenzará a sonar la música.

`at` permite distintas formas para especificar la fecha y hora en que debe activarse. Así, el tiempo se puede especificar en la forma HHMM o HH:MM para llevar a cabo una tarea en el mismo día. Por ejemplo la orden anterior es equivalente a:

```
$ at 16:00
```

Con `at` es posible usar `midnight` (medianoche), `noon` (mediodía), `teatime` (4 de la tarde) o `tomorrow` (mañana). También podemos anteponer a la hora `am` o `pm`.

Si queremos que `at` se ejecute en un día distinto al que estamos, pondremos la fecha en la forma ‘mes día’ por ejemplo, May 12.

Asociado al comando `at` tenemos los comandos:

**atq** muestra un listado de los trabajos en espera de ejecución.

**atrm** para eliminar trabajos en espera.

<sup>21</sup>es equivalente a usar:

```
$tar -cvf ~/kde_html.tar /usr/share/doc/HTML | bzip2
```

<sup>22</sup>También podemos dar el tiempo de espera como un incremento de un número de `weeks` (semanas), `days` (días) u `hours` (horas).

<sup>23</sup>`cdp` es una utilidad de texto que permite reproducir cd de música.

## ➡ Para practicar: at

Usando el comando `at` programar un trabajo para dentro de 2 minutos (o un tiempo razonable) que:

1. Te mande un correo con el texto que te parezca, una idea: “Curso Linux te saluda”
2. Cree un directorio en tu `$HOME` de usuario, y dentro un fichero de texto el de antes:

```
mkdir ~/uso_at
echo "Curso Linux te saluda" >~/uso_at/at.txt
```

Comprobar/leer el correo con (este comando se ve después en la página 241, pero la práctica mejor ahora):

```
$mail
&l
Para salir
&q
```

## cron

Para conocer esta importante herramienta:

- `$man cron`
- Capítulo 27: Tareas automáticas de *The Official Red Hat Linux Customization Guide*<sup>24</sup>.

Veamos el contenido de un fichero de configuración de `cron` básico:

```
$ cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Explicuemos un poco este fichero, las líneas tienen la forma:

fecha nombre\_usuario comando

- Con las 4 primeras líneas asignamos las variables de entorno con que se va a ejecutar `cron`.
- ¿Qué hacen las 4 últimas?
  - Con `run-parts` indicamos que se ejecuten los scripts pasados como parámetro en las fechas seleccionadas al inicio de la línea. El formato en que se codifican las fechas es:

CAMPOS	minuto	hora	día	mes	día de la semana
rango	0-59	0-23	1-31	1-12	0-7

- El 0 y el 7 del día de la semana corresponde al domingo (se puede usar *sun*, *sat*, ...).
- 3-6 equivale a la lista de números 3, 4, 5, 6

<sup>24</sup>En ese capítulo, se explican a su vez dos comandos más relacionados con el control de trabajos: `anacron` y `bathc`. `anacron` es una diferencia de `cron` en que no necesita estar en ejecución para realizar las tareas encomendadas. Por otro lado, `bathc`, es un `at` “inteligente” en el sentido de que ejecuta la tarea sólo si el sistema está poco “cargado de trabajo”.

También se explica la forma en que se puede permitir usarlo a los usuarios del sistema.

- El asterisco significa cualquier valor del rango permitido
- Podemos incrementar un valor con el formato `/numero`. Por ejemplo, podemos conseguir que un comando se ejecute cada 8 horas escribiendo `*/8` en el campo hora.
- Con ellas se controlan las acciones a realizar cada hora (`/etc/cron.hourly`), día (`/etc/cron.daily`), semana (`/etc/cron-weekly`) y mes (`/etc/cron.monthly`). Así, por ejemplo, el efecto diario de ejecutar los ficheros:
 

<code>/etc/cron.daily/tmpwatch</code>	borra los ficheros “viejos” de los subdirectorios temporales
<code>/etc/cron.daily/slocate.cron</code>	ejecuta updatedb con ciertos parámetros

## shutdown

```
shutdown [opciones] tiempo [mensaje]
```

De este grupo hemos visto varios comandos y su uso más corriente: `halt`, `reboot` y `shutdown`. En la primera entrega vimos que para apagar el sistema se tiene que escribir:

```
# shutdown -h now
```

o bien

```
# halt
```

y que para reiniciar el sistema la orden es:

```
# shutdown -r now
```

o bien

```
# reboot
```

Los parámetros que se han pasado son claros, `h` para “halt” y `r` para “reboot”, y la opción “now” por ahora mismo. Comentar que podemos pasarle como argumento el tiempo antes de cerrar/reiniciar el sistema, así en vez de `now` podemos escribir:

```
# shutdown -h +5
```

con lo que el sistema se cerrará dentro de cinco minutos.

## ps, kill

De los comandos asignados a este grupo sigamos con los más “importantes”. Para gestionar procesos disponemos fundamentalmente de dos órdenes: `ps` y `kill`.<sup>25</sup>

Linux es un sistema operativo multitarea y, por tanto, tiene que llevar un control permanente de cada uno de los programas que hay en ejecución, se llama *proceso* a una instancia de ese programa en ejecución. Cada proceso tiene un número único que lo identifica llamado **PID** (Process Identificator), el PID es la entrada de ese proceso en la tabla de procesos del kernel. Para listar los procesos en ejecución podemos usar el comando:

```
$ ps
```

una posible salida puede ser

```
PID TTY TIME CMD
```

```
3846 pts/1 00:00:00 bash
```

```
3899 pts/1 00:00:00 ps
```

con ella se nos muestran los procesos de este usuario junto al PID de los procesos.

La sintaxis básica de este comando es:

```
$ ps [opciones]
```

donde las opciones más usuales son:

## I Formato grande

---

<sup>25</sup>También `top`



**u** Da el nombre de usuario, la hora de comienzo y el uso de los procesos de este usuario de la máquina.

**a** Muestra también procesos de otros usuarios.

**x** Muestra los procesos sin terminal de control.

**r** Muestra sólo procesos que estén activos.

**txx** Muestra los procesos controlados por el terminal txx

Si queremos ver los procesos de todos los usuarios escribiremos:

```
$ ps -a
```

si además queremos los procesos sin terminal de control usaremos:

```
$ ps -ax
```

si queremos una salida más completa y que se actualiza de forma continua podemos usar<sup>26</sup>:

```
$ top
```

Con **kill** podemos terminar un proceso, la sintaxis más usual es:

```
kill [señal27] pid...28
```

donde **señal** es opcional y en general toma dos valores

**-15** (SIGTERM) es la señal por defecto y no siempre es capaz de “matar” todos los procesos

**-9** (SIGKILL) es el “Rambo” de las señales, acaba con cualquier proceso.

Si no especificamos ninguna señal, estamos mandando la señal 15 y de una manera no del todo formal, le estamos diciendo al proceso que se muera por las buenas. Es deseable que sea así, porque si hace caso el proceso, puede cerrar los ficheros, descargar los datos de memoria a disco y decirle a sus hijos (en caso de que los tuviera) que también se mueran por las buenas.

Si nos vemos obligados a utilizar la señal 9, lo matamos bien muerto, sin tiempo a que cierre ficheros ni descargue datos de memoria a disco. Moraleja, intentaremos mandarle primero un **kill** normal y si no hay manera pasaremos a la artillería pesada.

Supongamos que hemos cerrado **mozilla** y que notamos que el sistema “está lento”, escribimos:

```
$ps -ax
```

y ¡date!, **mozilla** sigue en ejecución con el PID

```
...
```

```
3940 tty1 S 0:01 /usr/lib/mozilla-1.0.1/mozilla-bin
```

```
...
```

ejecutamos entonces:

```
$ kill -9 3940
```

y listo, se acabó **mozilla** (en sentido figurado, claro está). Si ahora ejecutamos de nuevo:

```
$ ps -ax
```

no debería aparecer **mozilla** por ningún lado.

Si la lista de procesos es muy larga también podemos filtrar la salida con **grep**:

```
$ ps -ax | grep mozilla
```

---

<sup>26</sup>Para ampliar sobre **top** ejecutar:

```
$ man top
```

<sup>27</sup>Con

```
$ kill -l
```

podemos visualizar todas las señales posibles (es una **ele** minúscula).

<sup>28</sup>En este caso no hemos puesto ni **\$** ni **#** ya que el **root** podrá “matar” procesos de todos los usuarios, pero un usuario tan sólo podrá “matar” los suyos.

### ➡ Para practicar: Un asesino de procesos.

1. Ejecutar el comando

```
$top
```

y ordenar la salida por uso de la memoria (**M**) y después de la CPU (**P**). Para salir **q** (quit)

2. Uso conjunto de `ps` y `kill`

- a) En modo gráfico abrir una xterm y ejecutar en segundo plano el programa `gedit`

```
$gedit &
```

Comprobar la ID del proceso anterior con

```
$ps -a
```

y la diferencia de usar:

```
$ps -ax
```

```
$ps -aux
```

Matar el programa con:

```
$kill -15 PID_gedit
```

3. Acceder al sistema como root y como otro usuario del sistema (llamémosle INVITADO). Después de ver qué `tty` se corresponde con la sesión abierta de INVITADO con

```
#ps -G invitado
```

Pero el INVITADO es un poco gorrón y deseamos mandarlo a tomar viento, para eso matamos esa sesión con el PID del proceso `login`

```
#kill -9 PID_login
```

Comprobar que ya no hay “invitados”

### uname

```
uname [opciones]
```

Por último, con

```
$ uname -a
```

conseguimos toda la información de `uname`, una posible salida es:

```
(Sistema Hosts Versión del núcleo y fecha de la compilación)
```

```
Linux novo 2.4.18-14 #1 Thu Jan 17 13:35:24 EST 2002 i586 unknown
```

Si no ponemos ninguna opción sólo se nos muestra el nombre del sistema operativo.

## 16.3.7. Administrar ficheros

Analicemos algunos de los comandos de los que disponemos para administrar ficheros. Entre éstos, destacar:

### df

```
df [opciones] [sistema_archivos]
```

```
$ df
```

cuya salida podría ser:

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hdc1	2392739	1523644	770136	66 %	/
/dev/hdc5	1585200	765078	738193	51 %	/datos

es autoexplicativa. Si deseamos tener una salida más “comprensible” podemos escribir:

```
$ df -h
```

en cuyo caso la información se nos mostrará como sigue:

```
Filesystem Size Used Avail Use% Mounted on
/dev/hdc1  2.3G 1.5G 752M  66%  /
/dev/hdc5  1.5G 747M 721M  51%  /datos
```

## du

```
du [opciones] [nombre_archivo...]
```

```
$ du
```

Lista el espacio ocupado por los archivos o directorios que cuelgan desde donde se invoca. Si se ejecuta sin argumentos es poco práctico. Uno de los modos más corrientes de uso es:

```
$ du -sh directorio
```

da el total de espacio ocupado por ese directorio en formato “humano”, es decir, añade una letra indicativa del tamaño, como M para megabytes:

```
104M directorio
```

## fdformat

```
fdformat device
```

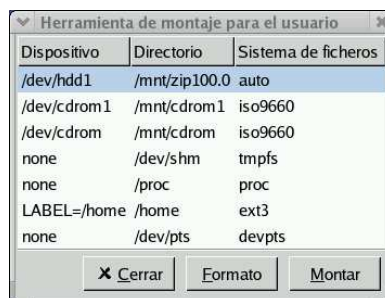
Cuando queramos formatear un disco escribiremos (sin tener el disco montado):

```
$ fdformat /dev/fd0H144029
```

Recordar que si desde una xterm ejecutamos

```
usermount
```

se nos abrirá una ventana parecida a



Este programa es una interfaz gráfica para montar, desmontar y formatear sistemas. En GNOME se encuentra en el menú de **Herramientas del Sistema**.

<sup>29</sup>En el supuesto de que nuestro disco flexible sea /dev/fd0. Normalmente H1440 no es necesario. Para ver los parámetros para otros formatos ejecutar:

```
man fdformat
```

## fdisk

Desde Linux podemos ejecutar el `fdisk` de Linux, es el mismo al que podemos acceder si instalamos Linux en modo experto. La sintaxis es:

```
# fdisk device
```

es en modo texto y un poco más árido que el `fdisk` del DOS. Es más potente que el `diskdruid` que nos aparece al instalar Linux. Por ejemplo, si queremos cambiar las particiones del segundo disco duro (esclavo del primer IDE) escribiremos:

```
# fdisk /dev/hdb
```

`fsck`, `mkfs`, `mount` y `umount` se vieron ya. El resto son de un uso muy específico<sup>30</sup> y en caso de necesitar usarlas es conveniente consultar en la manpages correspondiente.

### ➡ Para practicar:

Comprobar el espacio ocupado/disponible:

```
$df -h
```

Espacio que ocupa nuestro directorio de trabajo:

```
$du -sh ~/
```

Particiones del disco:

```
#!/sbin/fdisk -l /dev/hda
```

## 16.3.8. Comunicaciones y redes.

Se deja para después completar y estudiar: `ftp`, `telnet`, `ifconfig`, `route`, `netstat`, `ssh`, ...  
Veamos algunos ejemplos de cómo usar el resto.

## finger

Proporciona información sobre los usuarios conectados al sistema. Su sintaxis es:

```
finger [-lmsp] [usuario...] [usuario@host...]
```

si usamos `-l` la información es aún más amplia que con `-s`, por ejemplo:

```
$ finger
```

muestra entre otros datos: el directorio de conexión, el nombre completo, la fecha de conexión, etc. Si todavía queremos que la información sea más detallada escribiremos:

```
$ finger -l
```

Si sólo queremos información del usuario THALES escribiremos:

```
$ finger Thales
```

## who

```
who [opciones]
```

Con `who` podemos ver los usuarios conectados a nuestro sistema. Además, nos muestra el terminal en el que están conectados y el tiempo de conexión. `who` imprime la siguiente información por cada usuario que actualmente está conectado al sistema:

- nombre de la cuenta (login name)
- terminal
- tiempo que dura la sesión
- nombre de ordenador remoto o terminal X

---

<sup>30</sup>Salvo `tty`.

Si sólo queremos información sobre el usuario que ejecuta la orden escribiremos

```
$ who -m
```

**w**

```
w [usuario]
```

w nos da información sobre los usuarios que están conectados en ese momento y sobre sus procesos.

```
$ w
 4:59pm up 4:05, 3 users, load average: 0.00, 0.01, 0.01
USER      TTY  FROM LOGIN@  IDLE   JCPU   PCPU   WHAT
paco      tty1  -    12:54pm  4:04m  11.85s 0.03s  startx
cursolin  tty3  -    4:59pm  7.00s  0.06s  ?      -
paco      pts/0 -    4:53pm  0.00s  0.08s  0.03s  w
```

En la primera línea nos informa de

- el tiempo actual
- cuánto lleva el sistema funcionando
- cuántos usuarios están conectados y
- las cargas medias en los anteriores 1, 5 y 15 minutos.

A continuación para cada usuario se muestra:

- nombre de login,
- nombre de tty,
- nodo remoto,
- tiempo de conexión,
- tiempo inactivo,
- JCPU<sup>31</sup>,
- PCPU<sup>32</sup> y
- la línea de comando del proceso en curso.

**write**

```
write usuario [terminal]
```

El comando `write` nos permite mandar un mensaje a otro usuario. Previamente ese usuario debe tener `mesg`<sup>33</sup> en “y”, en el caso de que ese usuario tenga `mesg` en “n” el sistema nos avisará con un error:

```
write: "usuario" has messages disabled
```

Supongamos que deseamos enviar un mensaje al usuario THALES, y que tiene activa la opción de que le envíen mensajes, en ese caso escribiríamos

<sup>31</sup>JCPU es el tiempo usado por todos los procesos bajo el terminal tty.

<sup>32</sup>PCPU es el tiempo usado por el proceso en curso.

<sup>33</sup>

**mesg** Permite permutar la posibilidad de recibir mensajes de otros usuarios, funciona como un “interruptor” con sólo dos estados y o n. Estos estados se le tienen que pasar como parámetros.



```
$ write Thales
```

se nos avisará de que estamos en modo de edición, una vez escrito el texto que deseemos enviar como mensaje:

```
Hola Thales, que no se te olvide la cita.
pulsaremos [Ctrl]+[d] y el mensaje será enviado.
```

### wall

```
wall [archivo]
```

Si queremos enviar un mensaje no a un solo usuario, sino a todos los usuarios conectados, usaremos `wall`. Si lo que queremos es mandar un fichero escribiremos:

```
$ wall <fichero.txt
```

y el contenido de este fichero será enviado al terminal de todos los usuarios conectados al sistema.

Además de `write` y `wall`, para comunicarnos con el resto de usuarios disponemos de al menos dos comandos que permiten comunicarnos con ellos:

### talk

```
talk usuario [terminal]
```

`talk` permite la “conversación”<sup>34</sup> con otro usuario conectado al sistema, si deseamos establecer una conversación con `THALES`<sup>35</sup> escribiríamos:

```
$ talk Thales36
```

ante este comando, `talk` se queda en espera de que el usuario `THALES` responda a la llamada. Para salir de la “conversación” ejecutaremos `[Ctrl]+[c]`.

Para que funcione, deben existir los permisos correspondientes en el servidor.

### mail

```
mail [usuario]
```

En el caso de que nuestro usuario no esté conectado el mejor comando para comunicarnos con él es `mail`, si queremos enviar un mensaje a `THALES` escribiremos:

```
$ mail Thales
Subject: Cita
Te recuerdo que tenemos una cita
Cc:
```

tras escribir el texto pulsamos `[Ctrl]+[d]` y cuando `THALES` se conecte al sistema, éste le avisará de que tiene un correo:

```
Last login: Thu Mar 2 14:40:00 from Thales.cica.es
You have mail.
```

ejecutando `mail` podrá visualizarlo, borrarlo, etc:

```
$ mail
Mail version 8.1 6/6/93. Type ? for help. "/var/spool/mail/Thales":
1 message 1 new >N 1 paco@thales.cica.es Thu Mar 2 15:24 13/370 "Cita"
```

---

<sup>34</sup>Tipo irc

<sup>35</sup>Tiene que tener `mesg` en “y”

<sup>36</sup>En el caso de que nuestro usuario de destino esté conectado en dos terminales diferentes hay que especificar el `tty` donde queremos establecer la conexión.

Si es un usuario de otra máquina escribiremos:

```
$ talk usuario@nombre_máquina
```

&

Por ejemplo, para visualizar el correo tan sólo tiene que pulsar sobre el número que hay antes del correo:

& 1

para borrarlo ejecutar

& d 1

para salir

& q

Si queremos enviar un mensaje a un usuario de otra máquina escribiremos:

```
$ mail usuario@nombre_maquina
```

Para ampliar sobre este comando lo mejor es mirar en la ayuda. Otro agente de correo en modo consola es `pine`.

### ➡ Para practicar.

Comprobar los ejemplos de los comandos de esta sección (salvo `talk`)

## 16.3.9. Varios

De la selección de comandos de este grupo algunos son de uso trivial: `cal`, `clear`, `date`, `dmesg`<sup>37</sup>, `exit`. Tan sólo vamos a comentar `echo`<sup>38</sup>.

### echo

```
echo [cadena...]
```

Con `echo` podemos ver el contenido de una variable. Así con:

```
$ echo $HOSTNAME
```

veríamos en el terminal el nombre de nuestra máquina. Con

```
$ echo $PATH
```

veríamos el path de búsqueda, o si desde un terminal gráfico escribimos:

```
$ echo $DISPLAY
```

nos saldrá algo parecido a:

```
thales.cica.es:0.0
```

Remarcar que con `exit` cerramos el shell actual y también una `xterm`, así si estamos como usuario `THALES` y ya no queremos seguir trabajando como ese usuario escribiremos:

```
$ exit
```

y saldremos a:

```
localhost login:
```

El mismo efecto se consigue si pulsamos a la vez `[Ctrl]+[d]`.

El resto de comandos no presentan dificultad<sup>39</sup> y se trata de “jugar” con ellos.

<sup>37</sup>Nos muestra información del arranque y posterior funcionamiento de nuestro sistema.

<sup>38</sup>Otra forma de usar `echo` se ha visto en la sección de la `bash`.

<sup>39</sup>Salvo quizás `nohup`: con él podemos ejecutar comandos que no terminan al cerrar el terminal que se está usando. Lo podemos usar, por ejemplo, para dejar un trabajo ejecutándose en una máquina después de que nos desconectemos de ella (usando `telnet` o `ssh` por ejemplo)

## Capítulo 17

# La Shell Bash II

**Script:** Lenguaje de programación cuyo código no necesita ser compilado para ser ejecutado. Se suele llamar script a un programa o fragmento de código escrito en algún lenguaje de scripts. (De Wikipedia, la enciclopedia libre. <http://es.wikipedia.org/>)

### 17.1. Control de trabajos

El control de trabajos se refiere a la capacidad de parar selectivamente (suspender) la ejecución de procesos y continuar (reanudar) su ejecución posteriormente. Un usuario emplea esta facilidad típicamente a través de una interfaz interactiva suministrada conjuntamente por el controlador de terminal del sistema y bash.

La shell asocia un trabajo con cada tubería. Mantiene una tabla de trabajos ejecutándose actualmente, que pueden listarse con la orden `jobs`. Cuando bash arranca un trabajo asincrónicamente (en segundo plano), imprime una línea con un aspecto como ésta:

```
$[1] 25647
```

indicando que este trabajo es el número 1 y que el *PID*<sup>1</sup> del último proceso en la tubería asociada con él es 25647. Todos los procesos en una misma tubería son miembros del mismo trabajo. Bash emplea la abstracción del trabajo como la base para el control de trabajos.

Podemos simular esto ejecutando un sencillo proceso en segundo plano.

```
$sleep 10 &
```

Lanzamos un proceso, que lo que hace es dormir 10 segundos, en segundo plano.

Para facilitar la implementación de la interfaz del usuario al control de trabajos, el sistema mantiene la noción de un identificador (ID) de grupo de procesos del terminal actual. Los miembros de este grupo de procesos (procesos cuyo PGID es igual al PGID del terminal) reciben señales generadas por el teclado como SIGINT. Se dice que estos procesos están en primer plano. Los procesos en segundo plano son aquéllos cuyo PGID difiere del de nuestro terminal; tales procesos son inmunes a señales generadas desde el teclado. Sólo los procesos en primer plano tienen permitido leer o escribir en el terminal. A los procesos en segundo plano que intenten leer de (o escribir en) el terminal, el controlador de terminal les manda una señal SIGTTIN (SIGTTOU) que, a menos que sea capturada, suspende el proceso.

Para que podamos disponer del control de trabajos, el sistema operativo en el que la shell bash se está ejecutando ha de admitirlo. Y claro está, Linux lo admite, por supuesto.

Teclear el carácter:

**suspender** (típicamente `^Z`, `[Ctrl]+[Z]`) mientras que un proceso se está ejecutando, hace que ese proceso se pare y nos devuelva al indicador de la shell bash<sup>2</sup>.

---

<sup>1</sup>El PID es el número de proceso. Cada proceso tiene un número que lo identifica en el sistema.

<sup>2</sup>Podemos simularlo mediante:

```
$ sleep 15
```

y seguidamente pulsar **Control-z**



**suspensión diferida** (típicamente `^Y`, `[Ctrl]+[Y]`) hace que el proceso se pare cuando intente leer entrada desde la terminal, y el control se devuelve a `bash`. El usuario puede entonces manipular el estado de este trabajo, empleando la orden `bg` para continuar con él en segundo plano, la orden `fg` para continuar con él en primer plano, o la orden `kill` para matarlo. Un `^Z` tiene efecto inmediatamente, y tiene el efecto adicional colateral de que la salida pendiente y lo que haya en el búfer de entrada del teclado se descartan.

Hay varias formas de referirse a un trabajo en el shell.

- El carácter `%` introduce un nombre de trabajo. El trabajo número `n` puede ser referenciado como `%n`.
- Un trabajo puede ser referenciado utilizando un prefijo del nombre empleado para arrancarlo, o usando una subcadena que aparezca en su línea de órdenes. Por ejemplo. `%ce` se refiere a un trabajo `ce` parado. Si un prefijo concuerda con más de un trabajo, `bash` informa de un error. Con `??ce` nos referimos a cualquier trabajo que contenga la cadena `ce` en su línea de órdenes. Igual que antes, si la subcadena concuerda con más de un trabajo se nos notificará un error.
- Los símbolos `%%` y `%+` se refieren a la noción que tiene el shell del trabajo en curso, que es el último trabajo parado mientras estaba en primer plano o se arrancó en segundo plano. El trabajo anterior puede referenciarse usando `%-`. En la salida relativa a trabajos (e.g., la salida de la orden `jobs`), el trabajo actual se marca siempre con un `+`, y el anterior con un `-`. Para verlo, podemos ejecutar 5 ó 6 comandos como el siguiente

```
$ sleep 60 &
```

Seguidamente, ejecutamos

```
$ jobs
```

y los veremos a todos.

Si simplemente damos el nombre de un trabajo, esto puede traerlo a primer plano: `%1` es un sinónimo de `"fg %1"`, que trae el trabajo número 1 desde el segundo plano al primero. Similarmente, `"%1 &"` reanuda el trabajo 1 en el segundo plano, equivalente a `"bg %1"`.

El shell se entera inmediatamente de cuándo un trabajo cambia de estado. Normalmente, `bash` espera hasta que está a punto de mostrar un indicador antes de informar de cambios en el estado de un trabajo, para no interrumpir cualquier otra salida<sup>3</sup>.

Si se intenta salir de `bash` mientras hay trabajos parados, la shell muestra un mensaje de aviso. La orden `jobs` se puede usar entonces para inspeccionar sus estados. Si se hace un segundo intento de salir sin ninguna otra orden intermedia, el shell no muestra ningún otro aviso, y los trabajos parados se hacen terminar.

### ➡ Para practicar

Ejecutar en 2º plano los comandos:

```
#updatedb &
```

```
#tar -czf ~/doc.tgz /usr/share/doc
```

Comprobar que tenemos dos trabajos en segundo plano con:

```
#jobs
```

Pasar a primer plano el comando `tar` con:

```
#fg %2
```

<sup>3</sup>Se nos informará de tales cambios inmediatamente si la opción `-b` de la orden interna `set` está activada.

## 17.2. Expresiones

### 17.2.1. Expresiones condicionales

#### El comando test.

Este comando evalúa expresiones. Si la expresión (*expr*) es verdadera, *test* devuelve un cero. Si la expresión evaluada es falsa, devuelve un valor distinto de cero. La sintaxis es:

```
test expr
o de forma alternativa
[expr].
```

Como se observa, el comando *test*, puede ser reemplazado por corchetes.

Las expresiones condicionales son empleadas para comprobar los atributos de ficheros y realizar comparaciones de cadenas y aritméticas.

Las expresiones se forman a partir de las primarias monarias o binarias siguientes.

- a fichero  $\mapsto$  Verdad si fichero existe.
- b fichero  $\mapsto$  Verdad si fichero existe y es un fichero especial de bloques.
- c fichero  $\mapsto$  Verdad si fichero existe y es un fichero especial de caracteres.
- d fichero  $\mapsto$  Verdad si fichero existe y es un directorio.
- e fichero  $\mapsto$  Verdad si fichero existe.
- f fichero  $\mapsto$  Verdad si fichero existe y es un fichero regular.
- g fichero  $\mapsto$  Verdad si fichero existe y tiene el bit *SGID*.
- k fichero  $\mapsto$  Verdad si fichero existe y tiene el bit “pegajoso” (*STicky*).
- p fichero  $\mapsto$  Verdad si fichero existe y es una tubería con nombre (*FIFO*).
- r fichero  $\mapsto$  Verdad si fichero existe y se puede leer.
- s fichero  $\mapsto$  Verdad si fichero existe y tiene un tamaño mayor que cero.
- t fd  $\mapsto$  Verdad si el descriptor de fichero *fd* está abierto y se refiere a una terminal.
- u fichero  $\mapsto$  Verdad si fichero existe y tiene el bit *SUID*.
- w fichero  $\mapsto$  Verdad si fichero existe y se puede modificar.
- x fichero  $\mapsto$  Verdad si fichero existe y es ejecutable.
- O fichero  $\mapsto$  Verdad si fichero existe y su propietario es el *UID* efectivo.
- G fichero  $\mapsto$  Verdad si fichero existe y su grupo es el *GID* efectivo.
- L fichero  $\mapsto$  Verdad si fichero existe y es un enlace simbólico o blando.
- S fichero  $\mapsto$  Verdad si fichero existe y es un zócalo (*socket*).
- N fichero  $\mapsto$  Verdad si fichero existe y ha sido modificado desde que se leyó la última vez.
- fichero1 -nt fichero2  $\mapsto$  Verdad si fichero1 es más reciente (según la fecha de modificación) que fichero2.

fichero1 -ot fichero2  $\mapsto$  Verdad si fichero1 es más antiguo que fichero2.

fichero1 -ef fichero2  $\mapsto$  Verdad si fichero1 y fichero2 tienen los mismos números de nodo-í y de dispositivo.

-o nombre-opción  $\mapsto$  Verdad si la opción del shell nombre-opción está activada.

-z cadena  $\mapsto$  Verdad si la longitud de cadena es cero.

-n cadena  $\mapsto$  Verdad si la longitud de cadena no es cero.

cadena1 == cadena2  $\mapsto$  Verdad si las cadenas son iguales. También se puede emplear = en vez de ==.

cadena1 != cadena2  $\mapsto$  Verdad si las cadenas no son iguales.

cadena1 <cadena2  $\mapsto$  Verdad si cadena1 se ordena lexicográficamente antes de cadena2 en la localización en curso.

cadena1 >cadena2  $\mapsto$  Verdad si cadena1 se clasifica lexicográficamente tras cadena2 en la localización en curso.

arg1 OP arg2  $\mapsto$  OP es uno de -eq, -ne, -lt, -le, -gt, o -ge. Estos operadores aritméticos binarios devuelven verdadero si arg1 es igual a, distinto de, menor que, menor o igual a, mayor que, o mayor o igual a arg2, respectivamente. arg1 y arg2 pueden ser enteros positivos o negativos.

## Evaluación Aritmética

La shell permite que se evalúen expresiones aritméticas, bajo ciertas circunstancias.

La evaluación se hace con enteros largos sin comprobación de desbordamiento, aunque la división por 0 se captura y se señala como un error. La lista siguiente de operadores se agrupa en niveles de operadores de igual precedencia. Los niveles se listan en orden de precedencia decreciente.

`-`, `+`  $\mapsto$  menos y más unarios  
`!`, `~`  $\mapsto$  negación lógica y de bits  
`**`  $\mapsto$  exponenciación  
`*`, `/`, `%`  $\mapsto$  multiplicación, división, resto  
`+`, `-`  $\mapsto$  adición, sustracción  
`<<`, `>>`  $\mapsto$  desplazamientos de bits a izquierda y derecha  
`<=`, `>=`, `<>`  $\mapsto$  comparación  
`==`, `!=`  $\mapsto$  igualdad y desigualdad  
`&`  $\mapsto$  Y de bits (AND)  
`^`  $\mapsto$  O exclusivo de bits (XOR)  
`|`  $\mapsto$  O inclusivo de bits (OR)  
`&&`  $\mapsto$  Y lógico (AND)  
`||`  $\mapsto$  O lógico (OR)  
`expr?expr:expr`  $\mapsto$  evaluación condicional  
`=`, `*`, `/`, `%`, `+`, `-`, `<=`, `>=`, `&=`, `^=`, `|=`  $\mapsto$  asignación

Las constantes con un 0 inicial se interpretan como números octales. Un 0x ó 0X inicial denota un número en hexadecimal. De otro modo, los números toman la forma `[base#]n`, donde base es un número en base 10 entre 2 y 64 que representa la base aritmética, y n es un número en esa base. Si base se omite, entonces se emplea la base 10. Los dígitos mayores que 9 se representan por las letras minúsculas, las letras mayúsculas, `_`, y `@`, en este orden. Si base es menor o igual que 36, las letras minúsculas y mayúsculas pueden emplearse indistintamente para representar números entre 10 y 35.

Los operadores se evalúan en orden de precedencia. Las sub-expresiones entre paréntesis se evalúan primero y pueden sustituir a las reglas de precedencia anteriores.

## Combinación de expresiones

Las expresiones se pueden combinar usando los siguientes operadores, listados en orden de precedencia decreciente.

`--( expresión )`

Devuelve el valor de expresión. Puede ser utilizado para saltarse el orden normal de precedencia de operadores.

`! expresión1`

Esta expresión es verdadera si `expresion1` es falsa.

`expresión1 && expresión2`

Es una expresión verdadera si ambas, `expresión1` y `expresión2` son verdaderas.

`expresión1 || expresión2`

Verdadera si `expresión1` o `expresión2` es verdadera.

Los comandos `&&` y `||` no ejecutan `expresión2` si el valor de `expresión1` es suficiente para determinar el valor de verdad de la expresión condicional.

**➡ Para practicar**

Comprobar el efecto de los comandos:

```
$cd ~ ; ls; date
$test -a ~/.Xclients-default && ls
$test -a ~/.Xclients-defaults || ls
```

## 17.3. Comandos de la Shell II

### 17.3.1. Bucles

La shell Bash permite las siguientes construcciones de bucles.

**until**

La sintaxis del comando `until` es:

```
until condicion; do lista_de_comandos; done
```

Ejecutará la `lista_de_comandos` hasta que la condición devuelva un código de salida distinto de cero.

**while**

Su sintaxis es la siguiente:

```
while condicion; do lista_de_comandos; done
```

Ejecutará los comandos mientras la condición devuelva un código de cero; es decir, sea verdadera.

**for**

La sintaxis es:

```
for nombre [in palabras ...]; do lista_de_comandos; done
```

Su funcionamiento es el que sigue. Se expande `palabras` y ejecutará la `lista_de_comandos` una vez por cada miembro de la lista resultante, con `nombre` conteniendo el valor actual. Si el elemento `in palabras` no está presente, se asume la lista de parámetros (`$@`)<sup>4</sup>.

**➡ Para practicar**

Comprobar la salida de:

```
$num=10;until [ $num -lt 0 ];do echo número=$num; let num-=1; done
$num=0;while [ $num -lt 10 ];do echo número=$num; let num=num+1; done
$for fichero in *; do echo "$fichero"; done
```

---

<sup>4</sup>Tranquilos, su explicación viene un poco más adelante.



### 17.3.2. Sentencias condicionales

#### if

La sintaxis es la siguiente:

```
if condicion; then
    comandos_si_condicion;
[elif otra_condicion; then
    otros_comandos;]
[else comandos_alternativos;]
fi
```

Se evalúa la condición y si devuelve un estado de cero (verdadero), se ejecutan `comandos_si_condicion`. Si la condición es falsa (devuelve distinto de cero), se evalúa si existiera `otra_condicion`, que siendo verdadera permitiría la ejecución de `otros_comandos`. Si ninguna de las condiciones es verdadera y, existe la cláusula `else`, se ejecutarían los `comandos_alternativos`.

#### ➡ Para practicar

Comprobar los efectos de:

```
$ if [ -f .Xclients-default ]; then echo existe; else echo "No está";
fi
```

#### case

La sintaxis es la siguiente:

```
case palabra in [ ({} patron1 [| patron2]...) comandos ;;]... esac
```

La sentencia `case` ejecutará selectivamente los comandos que correspondan con la primera coincidencia de palabra con un patrón. Esto se ve mejor con un sencillo ejemplo:

Si el valor de `ANIMAL` coincide con alguno de caballo, perro o gato, se ejecutará el comando `echo -n "cuatro"` e irá al final para mostrar "patas". Los otros patrones son similares, pero en el caso de que no coincida con ninguno de los patrones anteriores, se ejecutarán los comandos de la entrada `"*)"`, que corresponda con una entrada que no ha coincidido con ninguno de los anteriores.

```
echo -n "Introduzca el nombre de un animal: "
read ANIMAL
echo -n "Un $ANIMAL tiene "
case $ANIMAL in
    caballo | perro | gato) echo -n "cuatro";;
    hombre | mono | pato ) echo -n "dos";;
    *) echo -n "un numero desconocido de";;
esac
echo " patas."
```

## 17.4. Ficheros de órdenes (Shell Scripts)

Una de las mayores ventajas que Linux ofrece a sus usuarios respecto de otros sistemas operativos es el control casi total sobre todas y cada una de las partes del sistema. En cada una de ellas, los «Shell Scripts» juegan un papel decisivo, ya que todos los procesos a llevar a cabo en estas tareas están controlados por ellos. Como una primera aproximación, podemos entender un *shell script* como un fichero `.bat` en



MS-DOS<sup>5</sup>: se trata de un fichero en el que hay un conjunto de órdenes del sistema operativo que se van ejecutando secuencialmente, de igual manera que si las vamos escribiendo una detrás de otra desde la línea de comandos<sup>6</sup>.

Un shell script es un fichero de texto que contiene comandos de la shell. Cuando un fichero de estas características es invocado desde la shell, su contenido es leído y ejecutado. Debemos darle los correspondientes permisos de ejecución con el comando `chmod`:

```
$ chmod +x fichero_script
```

Si la primera línea de un script comienza con los dos caracteres `'#!'`, el resto de la línea especifica el intérprete para el resto del contenido. Por ejemplo, para llamar a un intérprete de Perl (es un lenguaje de script), comenzaríamos la primera línea con

```
#!/usr/bin/perl
```

Si no especificamos ningún intérprete, tomará la shell por defecto, que es `/bin/sh`.

Como documentación recomendable el Howto *Programación en BASH-COMO de introducción y cómo no*

```
$ man bash
```

Además, en el directorio `/usr/share/doc/bash-2.05/` podemos ver una gran cantidad de script de ejemplo.

Cuando iniciamos una sesión en un sistema Linux, se inicia un shell «padre». Cada vez que deseamos ejecutar un comando externo, éste no se ejecuta por la shell en uso sino que ésta lanza otra shell<sup>7</sup> para ejecutar ese comando. Una de las ventajas de hacer esto así, consiste en que si ese comando produce algún desastre, lo más que puede pasar es que afecte a la shell que lo ejecutó y no a la shell padre.



- Si queremos que la shell padre ejecute (sin abrir una shell “hija”) los comandos guardados en un script tendremos que escribir un punto, dejar un espacio y escribir después el nombre del script.
- Si queremos verificar un script al tiempo que lo ejecutamos debemos usar la opción `-x`, así:  

```
$ sh -x nombre_script
```

con esta orden veremos en pantalla las instrucciones paso a paso.

### 17.4.1. Argumentos para los Procedimientos Shell

Un tipo diferente de variable de la shell es la que se transmite al procedimiento shell cuando se le invoca. Es un argumento para el procedimiento. A veces se les llama parámetros posicionales, porque se accede a ellos por su número de posición en la lista de argumentos. Por ejemplo, si tenemos un procedimiento que se invoca con la orden:

```
$ semana lunes martes miercoles jueves viernes sabado domingo
```

"lunes", "martes", "miercoles", "jueves", "viernes", "sabado" y "domingo" son parámetros posicionales, y se referencian como `$1`, `$2`, `$3`, `$4`, `$5`, `$6` y `$7`, respectivamente. `$0` será el nombre del propio procedimiento, en este caso "semana". Por último, `$@` se refiere a todos los argumentos, sin contar el nombre del procedimiento (`$0`).

Veamos un ejemplo:

```
$ cat mes
```

```
#####
```

<sup>5</sup>La cantidad de posibilidades que nos ofrecen los «scripts» en Linux/Unix es superior con diferencia a los lenguajes de comandos disponibles en otros sistemas operativos.

<sup>6</sup>Esto se llama *ejecución interpretada* de programas ya que cada línea se interpreta antes de su ejecución.

<sup>7</sup>Shell “hija”. Merece la pena comentar que la shell “hija” hereda las variables de entorno de la shell “padre” y que todos los cambios que la shell “hija” haga en las variables de ambiente no afectarán a las de la shell padre. Así, al terminar el comando, las variables de entorno serán las de la shell inicial: ¿recordáis la práctica sobre la variable path, los comandos `which` y su de la página ?, esta nota es explica las preguntas allí planteadas.



```
# Script: mes
# Descripcion: Este script me devolvera solo los tres primeros
# argumentos posicionales.
#####

echo $1 $2 $3

Con:
$ mes enero febrero marzo abril
enero febrero marzo
```



Si nos contesta bash: mes: "comand not found", quiere decir que en los directorios especificados por la variable PATH no se encuentra el ejecutable (ya sea un binario o un shell script) de nombre mes. Para poderlo ejecutar, en este caso podemos incorporar el carácter "." (punto) al PATH para indicar el directorio en el que nos encontremos en un momento determinado, o ejecutarlo de la forma:

```
$ ./mes
```

con lo que decimos: ejecuta el comando mes que está en el directorio actual, porque te lo digo yo.

Analicemos el script. Las líneas que comienzan por el símbolo almohadilla (en inglés se llama *number sign* #) son comentarios y su función es informativa y de documentación. Como la primera línea no indica un intérprete específico, se cogerá `/bin/sh` (que en Linux coincide con `/bin/bash`). O lo que es lo mismo, como si en la primera línea apareciera: `#!/bin/sh`

La línea `echo $1 $2 $3` coge los tres primeros argumentos con los que llamamos al script y los muestra por la salida estándar.

### 17.4.2. Entrada/Salida de Datos

A veces se desea escribir un procedimiento shell que pueda aceptar entradas interactivas del usuario.

Esto puede lograrse usando la orden `read` (*built-in* de la shell), que toma una línea de entrada de usuario y la asigna a una variable.

```
$ cat procesos
#####

# Script: procesos
# Descripcion: Me pregunta por el nombre de un usuario y a continuacion

# me lista sus procesos, que estan corriendo actualmente
# en el sistema.
#####

echo "Introduce nombre de usuario: \c"
read usuario
echo "Los procesos de $usuario son:"
ps -aux | grep $usuario
```

La parte interesante de este script es que podemos introducir valores interactivamente que serán recogidos por la orden `read`.

### 17.4.3. Bucles y condiciones

Mostramos a continuación un ejemplo simple pero que nos muestra muchas de las posibilidades de los shell script.

```
$ cat noescribir
#####

# Script: noescribir
# Descripcion: Este programa mostrara los ficheros, del directorio
# donde se ejecuta, en los que no tenemos permiso de escritura.
# Este programa, no funcionara para el root, ya que aun
# no teniendo para un fichero permiso de escritura, el root
# siempre podra escribir.
#####

for fichero in *
do
    if [ ! -w $fichero ]
    then
        echo "No puedes escribir el fichero $fichero"
    fi
done
```

Comienza con un bucle `for` en el que toma como valores para su recorrido los ficheros del directorio actual (con el metacarácter `*`). Van pasando uno a uno por el bucle, incorporándose los nombres de los ficheros a la variable “fichero”.

Para cada uno de los ficheros, se comprueba mediante la sentencia de evaluación de expresión (`[ ]`) si se puede escribir en el fichero. En este caso, comprobamos si no (!) podemos escribir (`-w`) en el fichero (`$fichero`).

Dependiendo de esta expresión se ejecutará condicionalmente el mensaje que nos muestra por pantalla que no tenemos permiso de escritura.

## 17.5. Ejemplos de scripts.



En nuestro ordenador hay bastantes ejemplos de scripts en los subdirectorios:

```
/usr/share/doc/bash-2.05b/scripts*
```

Abordaremos el estudio de varios casos prácticos, con el fin de comprobar sobre el terreno la aplicación palpable del mayor número posible de conceptos e ideas explicados hasta el momento, introduciendo de paso algunos conocimientos nuevos.

Para facilitar el comentario de los scripts las líneas están numeradas

### 17.5.1. Conversión a minúsculas de los nombres de archivos.

Un problema con el que nos encontramos cuando trabajamos con entornos Windows es que los nombres de los archivos están en mayúsculas. Veamos cómo poder arreglar esto:

```
$cat -b mayusculas8
```

```
1  #!/bin/bash
2  #####
```

<sup>8</sup>Con `-b` nos numera las líneas del fichero.



```

3  # Script: mayúsculas
4  # Descripcion: Este script convierte los nombres de archivos de
un
5  # subdirectorio que se pasa como argumento a minúsculas.
6  #####
7  directorio=$1
8  for nombre_M in `ls $directorio`
9      do
10         nombre_m=`echo $nombre_M | tr A-Z a-z`
11         mv $directorio/$nombre_M $directorio/$nombre_m
12     done
13     exit 0

```

Con la 1ª línea le decimos a la shell que use la shell bash como intérprete para este script. Las líneas 2 a 6 son comentarios que nos indican el nombre y cuál es la función del script. Veamos las otras.

La shell bash puede manejar variables sencillas. No hace falta definir las previamente sino que basta colocar un literal (su nombre), un signo igual y el valor que queremos darle. Con `directorio=$1` creamos una variable de nombre `directorio` a la que le asignamos como valor el primer parámetro que se le ha pasado al script como argumento, así si lo ejecutamos:

```
$ ./mayusculas trabajos
```

la variable `directorio`<sup>9</sup> almacenará el valor “trabajos”. A partir de ahora, `directorio` es el nombre de la variable, y `$directorio` es el contenido de la misma. Si queremos ver el contenido de una variable por pantalla podemos usar el comando

```
$ echo $directorio10
```

A partir de ahí comienza el bucle `for` que recorre todos los nombres de archivos del directorio pasado como argumento y en cada pasada se almacena el nombre del fichero en la variable `nombre_M`. Pero, cuidado con las comillas, después volveremos sobre esto, por ahora sólo nos interesa saber que recorreremos todos los ficheros del directorio. En la línea 10 creamos la variable `nombre_m` en la que almacenamos el nombre del fichero pero ya en minúsculas (para eso usamos la orden `tr` y una “tubería”). El proceso consiste en pasarle a `tr` el nombre del fichero en mayúsculas (`echo $nombre_M`) y “decirle” que sustituya las mayúsculas (A-Z) por minúsculas (a-z).

Ya está casi, ahora sólo tenemos que mover el contenido del fichero viejo (en mayúsculas) al nuevo (en minúsculas) y se acabó. Cuando hagamos esto con todos y cada uno de los ficheros de nuestro directorio terminará el bucle `for`.

Con `exit 0` salimos inmediatamente del script.

## Comillas

Disponemos de tres tipos de comillas:

“ Comillas dobles

’ Comillas simples

` Comilla inversa

con comportamiento diferente. Veámoslo con un ejemplo. Sea el script:

```
$ cat comillas
```

```
1  #!/bin/bash
```

<sup>9</sup>Si deseamos usar los nombres de las variables como partes de directorios, se nos puede plantear un problema al mezclar otros caracteres y los nombres de las variables. Cuando ocurra esto la shell no podrá interpretar bien el nombre y nos dará un mensaje de error. La solución consiste en encerrar el nombre de la variable entre llaves `${directorio}`. Así la shell leerá el contenido de la variable `directorio` y lo pondrá justo antes del siguiente carácter.

<sup>10</sup>Si lo que queremos es ver la lista actual de todas las variables podemos utilizar el comando `set`

```

2 #####
3 # Script: comillas
4 # Descripcion: Este script ejemplifica el uso de los distintos
5 # tipos de comillas
6 #####
7 directorio=$1
8 var1="ls $directorio"
9 var2='ls $directorio'
10 var3=`ls $directorio`
11 echo $var1
12 echo $var2
13 echo $var3

```

Analicemos la salida de este script pasándole como parámetro el directorio `/etc/rc.d`

```
$ ./comillas /etc/rc.d/
```

```
ls /etc/rc.d/
```

```
ls $directorio
```

```
init.d rc rc.local rc.sysinit rc0.d rc1.d rc2.d rc3.d rc4.d rc5.d rc6.d
```

Es decir:

- El texto que se encuentra dentro de unas comillas pierde su significado especial (el comando `ls` se interpreta como el texto “ls”) pero con la salvedad de las variables que sí que se expanden a su valor. Es por eso que el valor de la variable `var1` es el texto `ls` y a continuación el valor de la variable `$directorio (/etc/rc.d)`
- Si usamos comillas simples todo es texto y no se expanden las variables ni los comandos.
- Por último, con la comillas inversas el shell lo interpreta como la salida del comando que está dentro de las comillas inversas. Es por esto que esa variable contiene el listado del directorio.

### 17.5.2. Pasar varios argumentos

Con este ejemplo pretendemos que se vea que no estamos limitados a pasar un solo argumento:

```
$ cat -b argumento
```

```

1 #!/bin/bash
2 #####
3 # Script: argumento
4 # Descripcion: Este script permite contar el número de argumentos,
5 # además de mostrarlos en pantalla
6 #
7 #####
8 contador=0
9 for i in $*
10 do
11     echo $i
12     let contador=contador+1
13 done
14 echo Has pasado $contador argumentos

```

Observemos la salida del script, pasándole:

```
$ ./argumento 1 dos 3 y cuatro
1
dos
3
y
cuatro
Has pasado 5 argumentos
```

### 17.5.3. Script para compilar un fichero $\text{\LaTeX}$

Continuemos con un ejemplo sencillo. Usando este fichero (en modo gráfico) podemos compilar un fichero en formato  $\text{\LaTeX}$ <sup>11</sup> (sin la extensión) pasado como argumento y visualizar cómo quedaría finalmente en formato ps (PostScript). Además, antes de visualizarlo se borran algunos de los ficheros auxiliares generados en el proceso de compilación.

```
$ cat -b lat_dvi

1  #!/bin/bash
2  #####

3  # Script: lat_dvi
4  # Descripcion: Permite compilar/visualizar un fichero TeX.
5  #####
6  fichero=$1
7  latex ${fichero}.tex
8  latex ${fichero}.tex
9  dvips ${fichero}.dvi -o ${fichero}.ps
10 rm ${fichero}.aux
11 rm ${fichero}.toc
12 rm ${fichero}.log
13 exec gv ${fichero}.ps
14 exit 0
```

La única cosa a destacar del script (se puede mejorar mucho más) es que hemos usado `${variable}` en vez de `$variable`. Esto se hace cuando existen otros caracteres a continuación de la variable que pueden ser interpretados como parte de dicho nombre.

### 17.5.4. Texto en movimiento por la pantalla.

Este script permite hacer “scroll” con el texto que se le pasa como argumento, si no se le pasa nada mostrará el texto “CursoLinux”:

```
$ cat -b movertexto
1 #!/bin/bash
2 #####
3 # scrollbar - texto en movimiento #
4 #####
5 #Si la variable esta definida toma ese valor. Si no, le asigna 80
6 COL=${COLUMNS:-80}
7 #Si no se pasa nada, el texto es CursoLinux
8 [ $# -lt 1 ] && set -- CursoLinux
9 #Formatea el texto: como cadena (s) y alineado
```

<sup>11</sup>En la próxima entrada veremos un poco sobre qué es  $\text{\LaTeX}$ .

```

10 #a la izquierda (-) dentro del valor
11 # $* todos los parametros separados por espacios
12 Text=$(printf "%-${COL}s" "$*")
13 while :
14 do
15     #como antes pero para la variable Texto y
16     #añadiendo un retorno de carro /r
17     printf "%-${COL}s\r" "$Text"
18     #Desplazamos el texto un espacio
19     Text=" $Text"
20     #Mejor un poco más lento
21     sleep 1
22 done

```

### 17.5.5. Adivinar el número de palabras de un fichero.

En este ejemplo tenemos un script que nos va a permitir jugar con él. Se trata de adivinar el número de palabras de un fichero pasado como argumento. En él se han introducido algunas cuestiones nuevas, entre ellas:

- Otra forma de escribir comentarios  $\mapsto$  líneas 2 a 6. Para iniciar un comentario de varias líneas podemos usar

```

<<etiqueta

    texto del comentario
    ....

etiqueta

```

- Manejo de contadores  $\mapsto$  línea 17 en la que aparece el comando `let` que se emplea para evaluar expresiones aritméticas.
- El resto de líneas son inmediatas y/o aparecen comentadas en el propio fichero.

```

1  #!/bin/sh
2  <<COMENTARIO
3      Script : palabras
4      Descripcion : este script permite que juguemos a adivinar
el
5          número de palabras de un fichero pasado como parámetro
6  COMENTARIO
7  #Ponemos a cero el contador de intentos
8  intentos=0
9  #Antes de nada comprobamos que el número de argumentos
10 #es el adecuado (un sólo fichero)

11 if [ $# -ne 1 ]
12 then
13     echo "Número de argumentos no válido"
14 else
15     #almacenamos en la variable tamanno el número de palabras
del fichero
16     #para esto usamos la orden wc a la que le pasamos el fichero
17     tamanno=`wc -w <$1`

```

```
18      #Inicializamos la variable para entrar la primer vez en el
bucle
19      numero=-1
20      # iniciamos el bucle que se ejecuta mientras que no acertemos

21      until [ $tamanno -eq $numero ]
22      do
23          echo ¿Cuántas palabras tiene el fichero $1?
24          read numero
25          # aumentamos la variable intentos en 1
26          let intentos+=1
27          # comprobamos si nos hemos quedado cortos o nos hemos
pasado
28          if [ $numero -gt $tamanno ]
29          then
30              echo El fichero $1 tiene menos palabras.
31          else
32              if [ $numero -lt $tamanno ]
33              then
34                  echo El fichero $1 tiene más palabras.
35              fi
36          fi
37      done
38      #cuando acertamos salimos del bucle
39      echo  LO CONSEGUISTE, EL FICHERO TIENE $numero PALABRAS
40      echo  Y HAS NECESITADO $intentos INTENTOS
41  fi
```

### 17.5.6. Conectar a Internet con varios servidores de acceso

En el caso de que hayamos optado por usar los scripts para conectar a internet de la primera entrega, vamos a modificarlos de manera que podamos conectarnos a Internet con ellos en los supuestos:

- de que tengamos más de un proveedor de Internet y
- que además los números de teléfono no tienen por qué ser los mismos para todos ellos.

Para el desarrollo de esta práctica vamos a suponer que disponemos de tres servidores (ficticios): UNO, DOS y TRES. Como información añadida os remito al tema de conexión a Internet .

Con respecto a las DNS de todos los proveedores hay que ponerlas en `/etc/resolv.conf`, añadiendo nuevas líneas del tipo

```
nameserver xxx.xxx.xxx.xxx
```

Lo mejor es poner primero los DNS primarios de todos los proveedores y después los secundarios, además, es deseable poner en primer lugar los de aquellos que sepamos que vamos a usar con más frecuencia.

Supongamos que los datos de UNO (como es obvio necesitaríamos los de todos) son

#### Uno

Nombre de usuario: cursolinux@uno

Contraseña: linux00t

Número nodo local: 123456789

DNS primario: 195.76.42.101

Los ficheros necesarios son (los permisos los mismos que los de la conexión con scripts):

## **/etc/ppp/options**

Del fichero que aparecía en los scripts para un solo proveedor debemos eliminar las líneas `connect /etc/ppp/conecta` y `user usuario@servidor`

## **m\_servidor y servidor**

Para cada servidor de acceso necesitamos dos scripts como los que creamos a continuación para el servidor UNO. Los permisos y dueños de ambos ficheros al lado entre paréntesis.

### **/etc/ppp/m\_uno (- rwX r-x r-x root root)**

```
ABORT "BUSY"
ABORT "NO DIALTONE"
ABORT "NO CARRIER"
ABORT "ERROR"
""
"AT& F" TIMEOUT 5 OK
"ATDTW123456789" TIMEOUT 100 CONNECT
```

### **/etc/ppp/peers/uno (- rw- r- - r- - root root)**

```
connect "/usr/sbin/chat -v -f /etc/ppp/m_uno"
name cursolinux@uno
```

Para cada servidor crearíamos un par de script como los anteriores, hay que hacer notar que como es obvio los nombres son sólo orientativos.

### **chap-secrets o pap-secrets (- rw- — — root daemon)**

Con respecto a estos ficheros lo mejor es eliminar el fichero `pap-secrets` y realizar un enlace simbólico desde `pap-secrets` al fichero `chap-secrets`.

```
#rm /etc/ppp/pap-secrets
#ln -s /etc/ppp/chap-secrets /etc/ppp/pap-secrets
```

En el supuesto de que usemos autenticación CHAP el fichero tendrá una línea para cada servidor de acceso. Además, es conveniente proteger este archivo de “intrusos”.

### **/usr/local/bin/conecta (- rwX r-x r-x root root )**

Con el script que sigue podemos

- Ver qué proveedores de acceso disponemos (los tres antes mencionados).
- Conectar a Internet en función del que optemos.
- Comprobar que la conexión se ha realizado con éxito

```
1  #!/bin/sh
2  clear
3  echo "          *****"
4  echo "          *          Script para conectar a Internet          *"
```

```
5  echo "          *****"

6  echo "          Introduce el nombre del servidor de acceso:"
7      echo "          uno "
8      echo "          dos "
9      echo "          tres "
10     echo "          "
11     echo "          Salir"
12 #inicializamos la variable para entrar en el bucle
13 servidor=0
14 while [ -n $servidor ]
15 do
16     read servidor
17     case $servidor in
18         "Salir")
19         exit 0;;
20         *)
21         if [ -e "/etc/ppp/peers/$servidor" ]; then
22             /usr/sbin/pppd call $servidor
23             while !(/sbin/ifconfig | grep ppp0) >/dev/null
24             do
25                 sleep 1
26                 echo -ne "."
27             done
28             echo .
29             sleep 10
30             echo Conectado
31             break
32         else
33             echo "No te equivoques tecleando, el nombre tiene
que ser exacto"
34         fi
35         ;;
36     esac
37 done
```

**/usr/local/bin/desconectar (- rwx r-x r-x root root)**

Para poder desconectar:

```
#!/bin/sh
killall pppd
```

**Parte V**

# **Programas para Linux**



## Prólogo

En esta entrega vamos a enumerar algunas utilidades para Linux. Nos vamos a centrar en aquellas que confieren a Linux un aspecto más ofimático aunque no debemos olvidar que si hay algo en que Linux está completo es en lenguajes de programación, están todos (todos los de verdad).

También remarcar que no se nos puede olvidar que este curso parte de Thales y el CICA, y que Thales es la Sociedad de Profesores de Matemáticas de Andalucía, así que, además de los programas puramente ofimáticos hemos puesto una reseña de software específico de Matemáticas y de Astronomía

Hay multitud de lugares de internet desde donde localizar aplicaciones para Linux, algunos que os recomendamos son:

```
http://rpmfind.net/linux/RPM/ByName.html
http://freshmeat.net/
http://sourceforge.net
Loads of Linux Links http://loll.sourceforge.net/linux/links/
```

Específica de software científico:

```
http://ceu.fi.udc.es/SAL/sall.shtml
```

En castellano hay una que merece mención especial, es:

```
http://glub.ehu.es/soft/
```

Como es de esperar, en esta entrega nos vamos a centrar más en los procesos de instalación y en dar algunas pinceladas de lo que se puede hacer con cada programa que en dar un manual de uso de cada uno de ellos; hay que tener en cuenta que cada uno de ellos abarcaría por sí solo un curso.

Siempre que hemos podido, hemos puesto alguna dirección con manuales en castellano sobre los programas tratados. El motivo de hacerlo así y de no “meter el manual en la entrega” es que si se pusiesen todos, la entrega sería “enorme”. Además, entendemos que no a todos os interesan todos los programas y es mejor que cuando tengáis la “necesidad” de información dispongáis de una referencia que os facilite iniciaros con ellos.

Es conveniente destacar también, que ni es obligatorio, ni conveniente, instalarlos todos. El sentido de poner algunos programas “repetidos” es que si bien algunos son más completos que otros no todos son los más adecuados para todos los equipos. Así, no parece lógico intentar trabajar con OpenOffice en un 586 de primera generación, con no demasiada memoria RAM y poco disco duro. En este caso, lo mejor sería no instalarlo o desinstalarlo si está ya instalado, y puede ser más interesante trabajar con un programa más “viejo” y con menos necesidad de recursos (por ejemplo WordPerfect o LyX)

Vamos a seguir un esquema general en la instalación de todos los programas: consiste en que aquellos programas que no están en formato .rpm<sup>12</sup> los vamos a instalar en un subdirectorio, que cuelga del raíz, llamado /opt . Está claro que esto no es obligatorio y que se podrían poner en otro sitio (por ejemplo en /usr/local) pero hemos optado por el convenio anterior.

Por tanto, si váis a seguir este mismo esquema lo primero sería crear ese subdirectorio, si no lo tenemos ya<sup>13</sup>:

```
# cd /
# mkdir opt
```

Por último, confiamos en que no se haya pasado ningún problema de dependencias ya que al instalar/desinstalar todos estos programas entra dentro de lo posible que en algún momento se haya dejado instalado un paquete que sea necesario para alguno de ellos y que no se haya detectado el problema.

---

<sup>12</sup>Los que están en este formato no son ningún problema

<sup>13</sup>Se crea por defecto.

## Capítulo 18

# Instalación de programas

A partir de este momento la responsabilidad de elegir los programas está en tus manos. (*Manual Avanzado de linux*, RAÚL MONTERO RIVERO)

Si bien ya se ha visto el proceso de instalación de algunos programas, éste puede ser buen momento para resumir y aclarar conceptos que pueden haber quedado dispersos en temas anteriores. Comentemos, por tanto, algunos aspectos más que nos pueden ser útiles y nos pueden aclarar cuestiones dadas por “explicadas”.

Para instalar un nuevo programa podemos hacerlo de diferentes formas.

- A partir de código fuente y compilarlo nosotros mismos.
- A partir de binarios ya compilados y que se nos presenta en formato `.tgz`.
- En formato de paquetes RPM.

La instalación en formato RPM la hemos visto en entregas anteriores, con lo que no entraremos en ella.

### 18.1. Instalación a partir del código fuente.

Normalmente los ficheros en código fuente vienen en formato `.tgz` (o `tar.bz2`). Pasemos a explicar lo fundamental sobre este tipo de ficheros.

#### 18.1.1. Comando tar

Mediante este comando podemos “empaquetar” muchos ficheros (un árbol de directorios o árboles) en un solo fichero. Este fichero ya es más fácil de manejar, podemos comprimirlo y transferirlo a otra máquina o lugar y posteriormente recuperar los ficheros originales. La extensión normal para un fichero empaquetado con `tar` es `.tar`, que proviene de *tape archive* (archivo en cinta) que era el propósito primigenio, el conservar la copia en cintas en caso de desastre. Recordemos algunas cuestiones ya estudiadas sobre empaquetar y comprimir ficheros.

Para crear un fichero `tar` de un directorio utilizamos el comando:

```
tar -c -f fichero.tar directorio
```

Con esto creamos el fichero `fichero.tar` conteniendo todo lo que cuelgue del árbol que empieza en `directorio`. Por ejemplo,

```
$tar -cf home.tar /home
```

Estamos empaquetando (opción `c`) en el fichero `home.tar` (opción: `f home.tar`) el directorio `/home` con todo su contenido. En este caso hemos puesto las dos opciones (`-c` y `-f`) seguidas y con un solo guión por delante de ellas. Es necesario que tras la `f` esté el nombre del fichero que vamos a crear. Si quisiéramos restaurar el contenido del directorio usaríamos

```
$tar -xf home.tar
```



y todo el árbol de directorios guardado se restaurará en la ubicación donde estemos. Una vez que tenemos un archivo en formato tar, normalmente lo comprimimos para que no ocupe tanto espacio. Si realizamos

```
$gzip home.tar
obtenemos el fichero home.tar.gz.
```



Las extensiones `tgz` y `tar.gz` son “equivalentes”. Para conseguir los ficheros `home.tgz` y `home.tar.gz` hemos tenido que hacer lo mismo, primero empaquetar con el comando `tar` y después comprimir. `tgz` es una abreviatura de `tar.gz`.

### ➡ Para practicar: Instalación de Nightfall

Para ver un ejemplo, instalaremos el programa `nightfall`, <http://www.lsw.uni-heidelberg.de/~rwichman/Nightfall.html> es un programa interactivo de astronomía que puede generar animaciones de eclipses, curvas de velocidad radial, etc. El fichero siguiente contiene el código fuente de la aplicación:

```
nightfall-1.32.tar.gz
```

Lo bajaremos de:

```
http://www.lsw.uni-heidelberg.de/~rwichman/nightfall-1.32.tar.gz
```

Una vez en nuestra máquina, lo desempaquetamos con<sup>1</sup>

```
$gunzip -c nightfall-1.32.tar.gz | tar -xvf -
```

y vemos que crea un directorio `nightfall-1.32`

Las opciones del `tar` son:

**x** expandir el fichero en un árbol de directorios

**z** utilizar `gunzip` para descomprimir (si el fichero está comprimido)

**v** modo verbose (parlanchín) que nos da más detalles del proceso

**f** seguida del fichero que queremos expandir

Si nos vamos a ese directorio

```
$cd nightfall-1.32
```

y hacemos `ls`, vemos que hay un fichero llamado `README`. Normalmente, buscaremos ficheros como `README-algo`, `INSTALL` o `LEAME` para buscar las instrucciones de instalación. Si leemos `README` veremos los requerimientos que debemos tener en nuestro sistema y los pasos para compilar e instalar el programa. Estas serían las dependencias de paquetes que necesitamos en caso de que lo instalásemos en formato paquete, como RPM. Nos aseguramos de que tenemos instalado todo, en especial `gnuplot`, y pasamos a la acción:

```
$ rpm -q gnuplot
package gnuplot is not installed
```

Lo primero, mirar si lo tenemos en nuestros CDs (es del CD1), así que lo montamos e instalamos:

```
# rpm -ivh gnuplot-3.7.2-1.i386.rpm
Preparing... ##### [100 %]
1:gnuplot ##### [100 %]
```

La instalación de `nightfall` está bastante automatizada y posee un script que irá haciendo todos los pasos y si es necesario nos preguntará opciones.

```
$ ./DoInstall.sh
```

<sup>1</sup>Equivalente sería `$tar -xvzf nightfall-1.32.tar.gz`



El proceso manual sería

```
$ ./configure2  
$ make
```

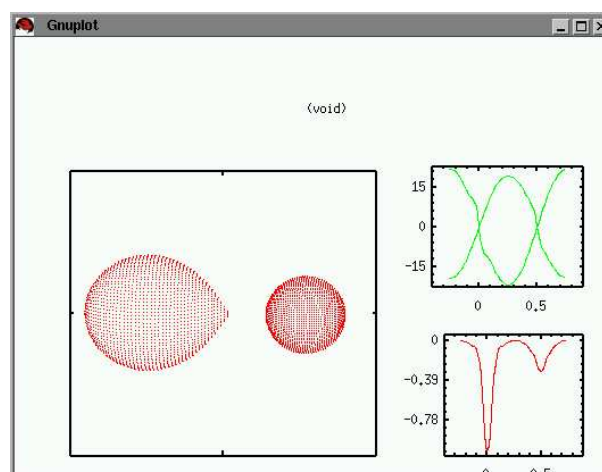
Realiza la compilación del programa<sup>3</sup>.

Hasta ahora hemos podido realizar todas las tareas como un usuario no autorizado, pero para colocar los ficheros en un directorio del sistema y que esté disponible para todos los usuarios, debemos tener permisos especiales. En este caso nos convertimos en superusuario y ya podemos realizar la instalación, que por defecto la hace en `/usr/local/bin` (para los ejecutables) y `/usr/local/share/nightfall` (para datos y documentación). Otra opción es colocarlos en el directorio del usuario `/home/usuario/`, donde no se necesitan permisos especiales.

```
$ su  
password:  
# make install
```

Realiza el proceso de instalación. Si en nuestra variable `PATH` se encuentra `/usr/local/bin`, podemos ejecutar

```
$ nightfall -A -Db 0.9 80 1.0 0.7 4000. 5000.
```



Como, por ahora, no nos interesa mantener en nuestra máquina este programa, nos situaremos en el subdirectorio en donde descomprimos las fuentes del programa y, para desinstalarlo, ejecutaremos:

```
# make uninstall
```

Todos estos problemas los pretenden solucionar las instalaciones mediante paquetes, del tipo RPM o debian (`.deb`). Por ello, recomendamos que se instalen programas en formato paquete. En la página principal del programa había otro llamado `nightfall-1.32-1.i386.rpm`, con el cual podríamos haber hecho simplemente:

```
# rpm -Uhv nightfall-1.32-1.i386.rpm
```

<sup>2</sup>Recoge opciones de nuestro sistema. Normalmente estos programas vienen preparados para detectar qué tipo de sistema es el nuestro Linux, Sun, HP... y atender a sus particularidades

<sup>3</sup>En ambos casos antes de proceder a compilar/installar el programa es necesario tener instalado el paquete `gcc-g77` (Da soporte para compilar programas de Fortran 77 con el compilador `gcc`). El comando `make`, recoge del fichero `Makefile` las dependencias y al no especificar ningún objetivo, éste será `all`.

```
warning: nightfall-1.32-1.i386.rpm: V3 DSA signature: NOKEY, key ID 0f571f6c
```

```
error: Failed dependencies:
    libglut.so.3 is needed by nightfall-1.32-1
    libgtkgl.so.5 is needed by nightfall-1.32-1
Suggested resolutions:
    glut-3.7-8.i386.rpm
    gtkglarea-1.2.2-13.i386.rpm
```

Instalemos ambos paquetes (CD3) y después `nightfall`.



Recordar que si deseamos que el comando `rpm` nos informe de qué paquetes hay que instalar, necesitamos tener instalado el paquete `rpmdb-redhat-8.0-0.20020910.i386.rpm` contenido en el CD3.

## 18.2. Instalación a partir de binarios

La instalación a partir de binarios es similar a la anterior, pero ya no tenemos que compilar, sólo instalar en su sitio los ejecutables y ficheros requeridos por la aplicación.



Lo vamos a ejemplificar en los parrafos que siguen. No os recomendamos que instaléis el binario en formato `tgz`, sólo que se entienda cómo se hace. Para comprender el porqué de este comentario no hay más que leer los problemas que presentan este tipo de instalaciones un poco más adelante.

El fichero `gnotepad.tgz`<sup>4</sup> lo hemos encontrado por la web en:

```
http://public.planetmirror.com/pub/slackware/slackware-7.1/slakware/gtk1/
```

Sabemos que es un binario para linux y además para las versiones de librerías `glibc` (nuestra versión de RedHat las tiene). No está de más que echemos un vistazo a su contenido, pero sin llegar a descomprimirlo<sup>5</sup>.

```
$tar -tzvf gnotepad.tgz6
drwxr-xr-x root/root      0 2000-06-10 19:15:19 ./
drwxr-xr-x root/root      0 2000-06-10 19:14:50 usr/
drwxr-xr-x root/bin       0 2000-06-10 19:15:02 usr/bin/
-rwxr-xr-x root/bin 192832 2000-06-10 19:15:02 usr/bin/gnp
drwxr-xr-x root/root      0 2000-06-10 19:14:50 usr/doc/
...
drwxr-xr-x root/root      0 2000-06-10 19:14:52 usr/share/locale/ja/LC_MESSAGES/
-rw-r--r-- root/root 43849 2000-06-10 19:11:10 usr/share/locale/ja/LC_MESSAGES/gnotepad+.mo
```

Con la opción `t`, el `tar` nos muestra el contenido del fichero, pero no lo expande en nuestro sistema de ficheros. Vemos que están los binarios (`/usr/bin/gnp`, por ejemplo), ficheros que necesita para configuración y documentación. Al guardarse el `tar` se ha hecho con un camino (path) relativo, con lo cual para expandirlo, debemos irnos al sitio de donde queremos que cuelgue el árbol (en este caso el raíz). Antes tenemos que hacernos superusuario:

```
$su
password:

#cd /
#tar -xzvf gnotepad.tgz
```

<sup>4</sup>Se trata de un editor de páginas HTML.

<sup>5</sup>También podemos hacerlo con `midnight commander`

<sup>6</sup>No se ha puesto la salida completa del comando

y nuestro programa se colocará en su sitio. Para ejecutarlo:

```
$ /usr/bin/gnp &
```

Problemas de este tipo de instalaciones:

- Sin darnos cuenta hemos podido machacar un fichero perteneciente a otro programa, que puede dejar de funcionar.
- Puede que necesitemos algún programa y de una determinada versión para que éste funcione (dependencia), pero no lo sabemos hasta que desgraciadamente el programa no nos funcione.
- Si ya no nos interesa este programa por ejemplo, porque hemos encontrado otro que nos gusta más, ¿cómo lo borramos para que no nos ocupe espacio innecesario?
- No sabemos cuántos programas tenemos instalados

#### ➡ Para practicar

Al igual que antes, es recomendable instalar los programas en formato RPM (o .deb) para evitar estos problemas. En:

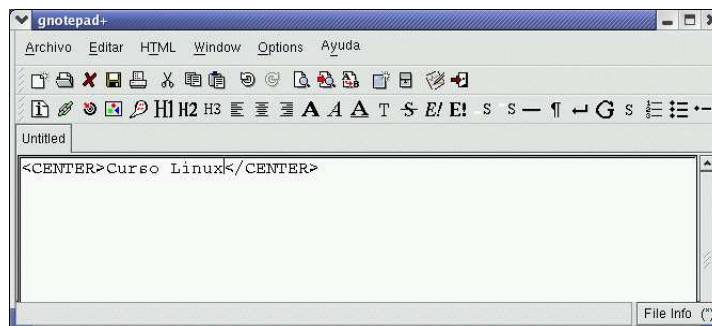
<http://gnotepad.sourceforge.net/#download>

tenemos `gnotepad+-1.3.3-1.i386.rpm`, con el cual podríamos haber hecho simplemente:

```
# rpm -Uhv gnotepad+-1.3.3-1.i386.rpm
```

y ya estaría instalado. Para ejecutarlo:

```
$ gnp
```



## 18.3. En formato RPM

Ya se ha visto bastante sobre este tema pero nunca está de más comentar algún programa más. Utilicemos esta sección como excusa para hablar del programa `Xwpe`. Se trata de un software de desarrollo que nos agrada bastante, sobre todo en su versión para las X, ya que ocupa muy poco y está muy logrado.

La apariencia de este programa es similar a los productos de desarrollo de Borland pero con la ventaja de que no está asociado a ningún lenguaje de programación, es decir, cuando se compila o se aplica `make` para el programa podemos escoger el compilador que deseemos. Podemos conseguirlo desde <http://www.identicalsoftware.com/xwpe/>.

Para disponer de la versión “completa” del programa tendremos que instalar los paquetes: `xwpe-1.5.29a-1.i386.rpm` y `xwpe-X11-1.5.29a-1.i386.rpm`

```
# rpm -ivh xwpe-1.5.29a-1.i386.rpm xwpe-X11-1.5.29a-1.i386.rpm
```

Desde una `xterm` con la orden<sup>7</sup>:

```
$ xwpe
```

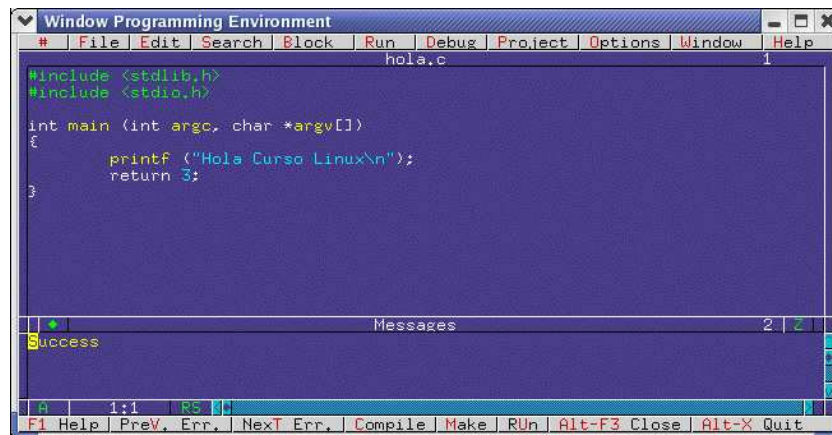
o con

---

<sup>7</sup>Con `xwe` y `we` para el modo gráfico y texto respectivamente ,accedemos sólo al editor de textos.

```
$ wpe
```

en modo texto, dispondremos de una ventana similar a la que sigue, desde la que podremos escribir/compilar/depurar nuestros programas escritos en distintos lenguajes de programación. En particular, si escribimos las líneas de código anteriores, lo guardamos con el nombre `hola.c` y en el menú **Run**



pulsamos de nuevo sobre **Run**, veremos que él solito compila el programa y además nos muestra en una xterm la salida de éste:



# Capítulo 19

## Gráficos

Una imagen vale más que mil palabras. (¿Es GPL?)

En este apartado vamos a intentar mostrar que Linux dispone de todo lo necesario para trabajar con gráficos. Cuando se habló de OpenOffice se puso de manifiesto que con ese programa podemos hacer casi todo lo que se nos ocurra y que trae herramientas más que suficientes para el usuario medio en su “pelea” diaria con el ordenador.

Aparte de los que se van a exponer, pueden ser de utilidad:

**xwd** capturador de pantallas (el de toda la vida)

**giftrans** para transformar gráficos en formato gif87 a gif89<sup>1</sup>.

**cycas** es un programa arquitectura 2D/3D (<http://www.cycas.de>).

No se van a ver aquellos cuyo uso es más “evidente” y que se instalan con los escritorios GNOME/KDE de forma automática.



Dos aplicaciones que si bien no son de “gráficos” es interesante conocer:

**xine** reproductor de video para Linux (con soporte de video mpeg-2 y mpeg-1, DVD's, video CDs, SVCDs y ficheros AVI<sup>2</sup>). Se puede conseguir en: <http://xinehq.de/>.

**cinelerra** programa de edición de vídeo para Linux <http://heroinewarrior.com/cinelerra.php3>

### 19.1. xv

Xv es un programa gráfico que aunque no viene con los CDs sí que está en

<http://rpmfind.net/linux/RPM/contrib/libc6/i386/xv-3.10a-27.i386.html>

Si ya lo hemos bajado, procedamos a su instalación:

```
# rpm -ivh xv-3.10a-27.i386.rpm
#####
```

El programa **xv** es un clásico para manejar gráficos dentro del mundo UNIX y permite trabajar con los formatos gráficos más usuales.

Podemos disponer de un manual de uso que explica bastante bien todas sus funciones. Para poder acceder a él debemos bajarnos el fichero

---

<sup>1</sup>Yo lo uso sobre todo para hacer gif transparentes. No se instala por defecto y está en el CD2.

<sup>2</sup>Hay que instalar el paquete `w32codec`



<http://rpmfind.net/linux/RPM/contrib/libc6/i386/xv-docs-3.10a-27.i386.html>

e instalarlo. Una vez instalado, el manual es el fichero<sup>3</sup>  
`/usr/share/doc/xv-docs-3.10a/xvdocs.ps`

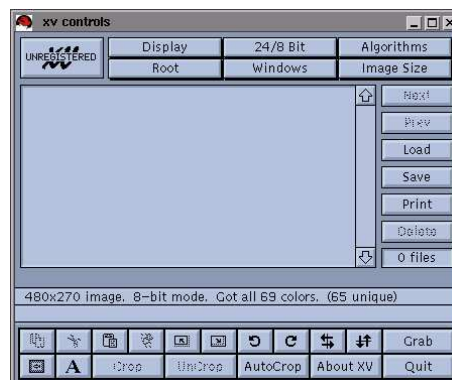
Algunas de sus características más destacables son:

- Incluye un visor gráfico para visualizar todos los gráficos de un directorio, incluso gráficos en formato *ps*<sup>4</sup>
- Permite transformar de unos formatos gráficos a otros.
- Incluye un buen capturador de pantallas<sup>5</sup>
- Permite realizar transformaciones sobre las imágenes tales como: recortarlas, rotarlas, etc.
- Trae una serie de filtros (algoritmos) que permiten modificar las imágenes.
- Etc.

Comentemos cómo acceder a estas funciones. Ejecutemos desde el entorno gráfico:

```
$ xv
```

La ventana inicial del programa nos informa de que para acceder al menú principal de la aplicación debemos pulsar sobre el botón derecho del ratón. Una vez que pulsemos este botón veremos una ventana similar a



Para acceder al visor gráfico tenemos que pulsar sobre el botón **Windows** de la ventana principal del programa, se abrirá el menú en el que seleccionaremos **Visual schnauzer**<sup>6</sup>.



<sup>3</sup>Está en formato PostScript y en inglés.

<sup>4</sup>Los visores gráficos que aparecen en el apartado 19.7 no tienen esta posibilidad.

<sup>5</sup>La mayoría de las capturas gráficas de este curso están hechas con él

<sup>6</sup>También se accede a esta pantalla pulsando `[Ctrl] + [v]`

Usando esta ventana podemos desplazarnos por el árbol de directorios hasta aquel que deseemos visualizar. Por ejemplo desplacémonos hasta el directorio `/usr/share/icons/`, una vez en él pulsaremos sobre el botón **Misc. Commands** y seleccionaremos todos los ficheros (usando la opción **Select All Files** o bien pulsando sobre `[Ctrl]+[a]`), por último pulsaremos `[Ctrl]+[g]` o bien desde el menú emergente que aparece al pulsar sobre **Misc. Commands** pulsaremos sobre **Generate Icons**, el resultado será



Comentemos que para capturar una pantalla o una región de ésta hay que pulsar sobre el botón **Grab** de la ventana principal, veremos la pantalla



Las opciones que podemos controlar usando esta ventana son:

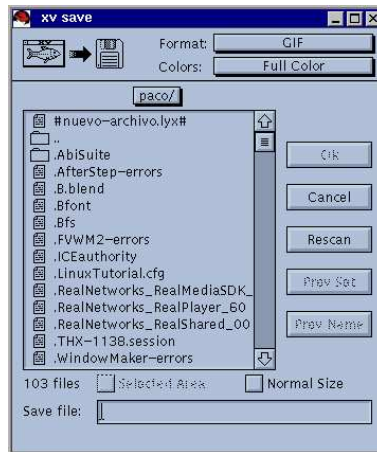
**Grab** para comenzar a capturar una ventana. Podemos, bien seleccionar una ventana completa si pulsamos sobre ella con el botón izquierdo del ratón, bien capturar un área rectangular si pulsamos con el botón central del ratón y manteniéndolo pulsado, lo arrastramos desde la esquina superior hasta la esquina inferior de la captura. Emulamos el efecto de pulsar el botón central del ratón pulsando a la vez los dos botones, el izquierdo y el derecho<sup>7</sup>.

**Hide** Para que se oculten las ventanas del programa `xv` al iniciar la captura

**Delay** Tiempo de espera en segundos antes de comenzar la captura. Es útil si se usa junto con **Autograb**. Si por ejemplo deseamos capturar un menú emergente pondremos en **Delay** 3 ó 4 segundos y al pulsar en **Autograb** nos dará tiempo a abrir el menú que deseamos capturar y el programa solo realizará la captura.

Con respecto a exportar a otros formatos, no presenta mayor dificultad. Tan solo pulsaremos sobre el botón **Save** del menú principal y en la ventana que aparece seleccionaremos el formato gráfico que deseamos

<sup>7</sup>Esto es cierto sólo en el caso de que al configurar el ratón optásemos por la opción de Emular tres botones.



Usando el menú desplegable que aparece al pulsar sobre GIF (es el formato por defecto) podemos optar por uno de los formatos: GIF, JPEG, TIFF, PNG, PostScript, PBM/PGM/PPM (raw), PBM/PGM/PPM (ascii), X11 Bitmap, XPM, BMP, Sun Rasterfile, IRIS RGB, Targa (24 bits), FITS y PM. Además, para cada uno de ellos podremos seleccionar si lo guardamos en color, gris, blanco y negro o reducimos los colores.

Las posibilidades del programa son bastante amplias y os remitimos al manual antes comentado (son 128 páginas).

## 19.2. The Gimp

Gimp (*General Image Manipulation Program*) es un programa de manipulación de imágenes para retoque fotográfico y composición de imágenes. Es muy útil para la creación de logotipos y gráficos para páginas web. Posee la mayoría de herramientas y filtros de otros programas comerciales.

Podemos ampliar sobre él en:

- La web del programa: <http://www.gimp.org>
- Imprescindible y en castellano: <http://gimp.es.gnome.org/>, de esta página comentar
  - Tutoriales sobre GIMP en castellano: <http://gimp.es.gnome.org/manuales.php>
  - Documentación sobre GIMP en castellano: <http://gimp.es.gnome.org/documentacion.php>
  - Todo está bien. Es la página de visita obligada, enlaces, manuales, tutoriales, etc.
- Manual del programa: <ftp://manual.gimp.org/pub/manual/>
- *The Official Red Hat linux Getting Started Guide* en el capítulo 10.
- Un libro en castellano: *GIMP*, ALEX HARFORD. Ed. Prentice Hall

Por el “grosor” del manual<sup>8</sup> es fácil deducir que el uso de The Gimp, igual que PhotoShop no es inmediato. Comentemos algunos de sus usos más frecuentes. Pero antes, para los que queráis introducirlos en el programa, algunas direcciones más de artículos en castellano sobre el uso de Gimp:

- Una Introducción al GIMP  
<http://www.acm.org/crossroads/espanol/xrds3-4/gimp.html>

<sup>8</sup>Son casi 10 MB y más de 900 páginas.

- Fotocomposición Gráfica con ... El Gimp  
<http://www.linuxfocus.org/Castellano/March1998/article34.html>
- Creando Texto con The Gimp  
<http://www.linuxfocus.org/Castellano/May1998/article36.html>
- Crear un efecto de fuego en GIMP  
<http://www.linuxfocus.org/Castellano/November1999/article112.html>
- Creando y Editando Animaciones con GIMP<sup>9</sup>:  
<http://www.linuxfocus.org/Castellano/July1998/article51.html>
- Las utilidades de selección de Gimp y la corrección de color  
<http://www.linuxfocus.org/Castellano/January2001/article119.shtml>

Incorpora gran cantidad de herramientas para manipular las imágenes, incluyendo canal de operaciones y capas, efectos, antialiasing y conversiones. También incluye un capturador de pantallas. La versión que acompaña a la RedHat incluye gran cantidad de *plug-ins* (más de 130) que permiten ampliar las posibilidades de este programa.

En los CD de la distribución disponemos de los paquetes (los marcados con ★ se instalan por defecto):

- CD 1
  - ★ `gimp-print-4.2.1-5.i386.rpm`
  - ★ `gimp-print-utils-4.2.1-5.i386.rpm`
- CD 2
  - ★ `gimp-1.2.3-9.i386.rpm`
  - ★ `gimp-data-extras-1.2.0-6.noarch.rpm`
  - `gimp-devel-1.2.3-9.i386.rpm`
  - `gimp-print-cups-4.2.1-5.i386.rpm`
  - `gimp-print-devel-4.2.1-5.i386.rpm`
  - ★ `xsane-gimp-0.84-8.i386.rpm`
- CD3
  - `gimp-perl-1.2.3-9.i386.rpm`

Cada uno de los paquetes anteriores es:

**gimp** Paquete principal de la aplicación.

**gimp-data-extras** Plantillas, gradientes, etc. Este paquete no es necesario.

**gimp-devel** Librerías estáticas y archivos de cabecera para escribir “plugins” y extensiones GIMP.

**gimp-perl** Contiene extensiones y *plugins* de Perl.

**gimp-print** *Drivers* de impresora.

**gimp-print-devel** Archivos para aplicaciones de desarrollo usadas por el paquete `gimp-print`

---

<sup>9</sup>Este artículo está un poco anticuado con respecto a las posibilidades actuales del programa.

**gimp-print-cups Drivers** para impresoras Canon, Epson, HP, etc para CUPS (*Common UNIX Printing System*, sistema de impresión por defecto en Red Hat 9.0 y ya disponible en esta versión)

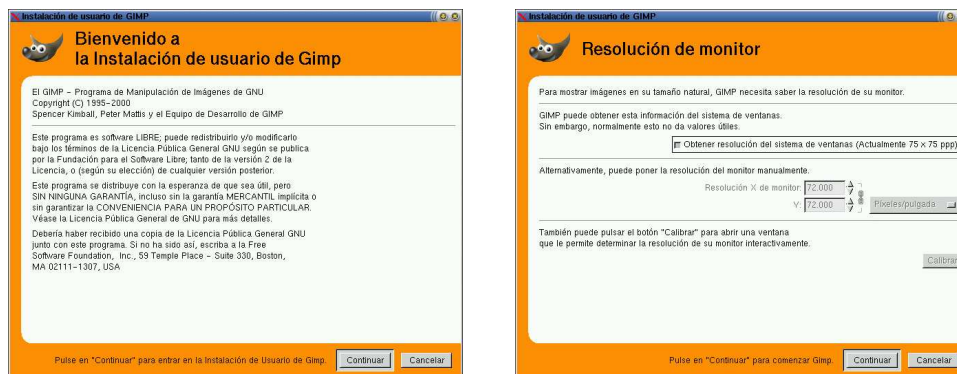
**gimp-print-utils** Programas de utilidades desde `gimp-print`

**xsane-gimp** Para trabajar con el escáner desde Gimp<sup>10</sup>

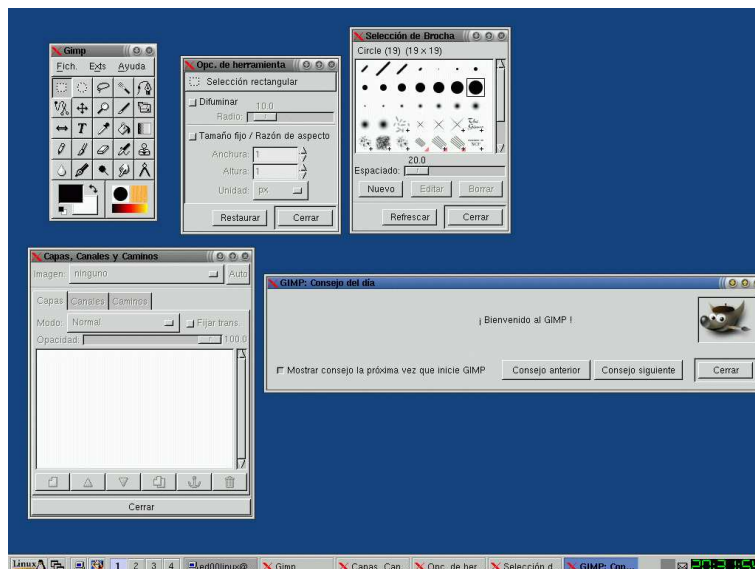
Para ejecutarlo, podemos hacerlo desde  → **Gráficos** → **El Gimp** o desde una xterm escribiendo:

```
$ gimp
```

La primera vez que lo ejecutemos tendremos que aceptar en una serie de ventanas para personalizar el programa al usuario que lo ejecuta,



En general, y salvo que tengamos muy claro lo que hacemos, lo mejor es mantener las opciones por defecto. En cualquier caso siempre podemos modificar estos valores después. Una vez pasado este trámite accederemos a una serie de ventanas:





#### ■ Consejo del día

<sup>10</sup>Además, deberían de estar ya instalados los paquetes `sane-backends-1.0.8-5`, `sane-frontends-1.0.8-4` y `xsane-0.84-8` que son los que nos permiten trabajar con el escáner (siempre que esté soportado)

- Selección del tipo/grosor de la brocha
- Opciones de las herramientas que aparecen en el menú principal.
- Ventana para trabajar con capas.
- y entre ellas la “principal” de la aplicación



La explicación de cada uno de estos botones es la estándar de las herramientas de dibujo, además cuando mantenemos el ratón sobre cualquiera de ellas veremos una pequeña nota sobre su función:

	Seleccionar regiones rectangulares en la capa activa.		Añadir texto a la imagen
	Selección de regiones elípticas en la capa activa.		Escoger un color de la imagen
	Selección de regiones “tiradas a mano”.		Rellenar una selección con un color o un modelo
	Selección de regiones basadas en el color (contiguas)		Rellenar con un “gradiente” de colores
	Selección de regiones usando curvas Bezier		Dibujar trazos muy definidos con lápiz
	Selección automática		Pintar trazados borrosos con pincel
	Para mover capas y selecciones		Borra hasta transparencia
	Zoom de una imagen.		Pincel aéreo con presión variable
	Recortar la imagen		Pintar usando modelos o regiones de imagen
	Transformar la capa o selección		Difusión o afinamiento
	Refleja la capa o selección.		Dibujar con tinta
			Aumentar o disminuir la intensidad



Difuminar una imagen con el pincel seleccionado en ese momento



Medir distancias y ángulos

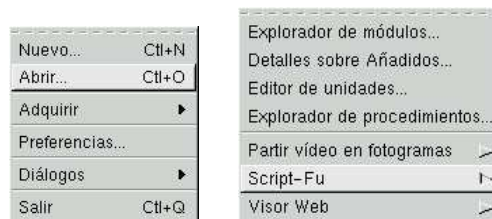


Colores activos

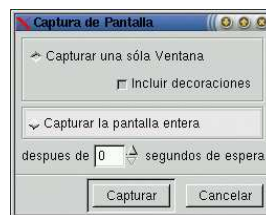


Pincel, el modelo y el gradiente activo

Comentemos algunos aspectos básicos del programa. En primer lugar decir que soporta multitud de formatos gráficos<sup>11</sup> y que su formato nativo es el XCF. Los menús principales de la aplicación son **Fich.** y **Exts**:

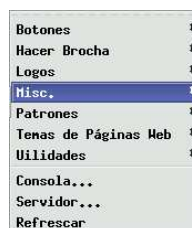


Las opciones del primer menú no presentan problema (Nuevo, Abrir, Adquirir, Preferencias, Dialogos y Salir). Con la opción **Adquirir** se abrirá la ventana que permite capturar pantallas gráficas:



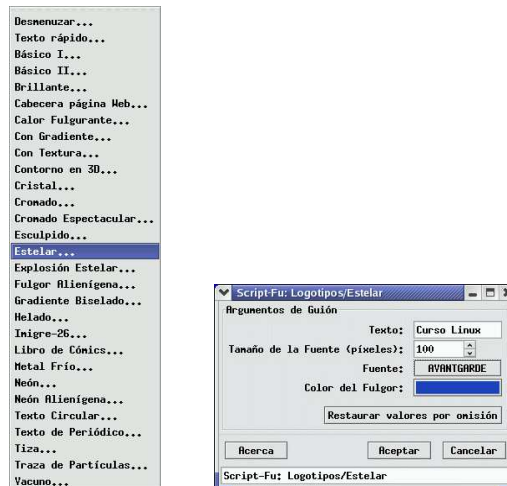
Tan solo comentar que podemos grabar una ventana gráfica, o bien poner un retardo de espera para que el programa capture de forma automática el elemento activo al pasar ese tiempo.

Comentemos el segundo menú (**Exts**). Desde él podemos acceder a algunos de los aspectos iniciales más llamativos de GIMP, para ello pulsemos sobre la opción **ScriptFu** y veremos la ventana:



Mediante este menú podemos crear logotipos, texturas, botones, fondos, etc. Veamos cómo crear un logotipo del tipo **Estelar**: pulsemos sobre **Logos** en el menú anterior y veremos la ventana

<sup>11</sup>Están casi todos los más usuales: BMP, CELS, FITS, FLI, GBR, GIF, GIcon, HRZ, HTML, Header, Jpeg, PAT, PCX, PIX, PSD, PNG, PNM, PostScript, SGI, SUNRAS, TGA, Tiff, XCF, XWD, Xpm, gzip, bzip2

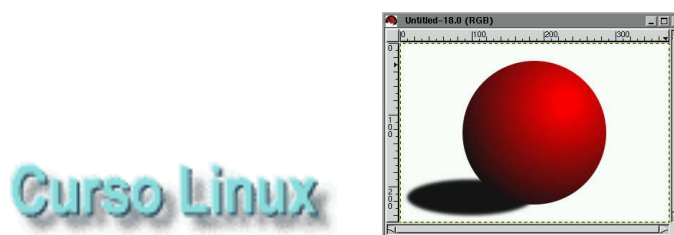


Con cada una de estas opciones podremos realizar un modelo diferente. Si ahora pulsamos sobre el elemento **Estelar** se abrirá una ventana como la anterior que nos va a permitir elegir el texto que vamos a escribir, el tamaño de la letra, el color, la fuente, etc.

Si escribimos *Curso Linux*, seleccionamos la fuente *Avantgarde* y personalizamos el tamaño del texto a 100 pixels, cuando aceptemos obtendremos:



No está mal para ser lo primero que hacemos. Continuemos con el menú **Exts** de la ventana inicial, si deseamos hacer un fondo pulsaremos sobre **Patrones**. Con la opción **Temas de páginas Web** podemos realizar elementos para nuestras páginas Web (flechas, títulos, etc). Por ejemplo, si pulsamos sobre **Gimp.Org** y después sobre **Encabezado Pequeño** podemos conseguir el gráfico



en el que lo único que se ha cambiado ha sido el texto por defecto por *Curso Linux*. Bonito ¿verdad?. Con las demás opciones podremos conseguir: botones, una esfera, etc

Si ahora queremos modificar esa imagen podemos pulsar sobre ella con el botón derecho del ratón y accederemos al menú “principal” de la aplicación





Las distintas opciones que presenta este menú se escapan de nuestras pretensiones y de nuevo os re-mitimos a los manuales/artículos antes comentados. Sí que merece la pena echar un vistazo a la magnífica colección de **Filtros** que acompañan al programa.



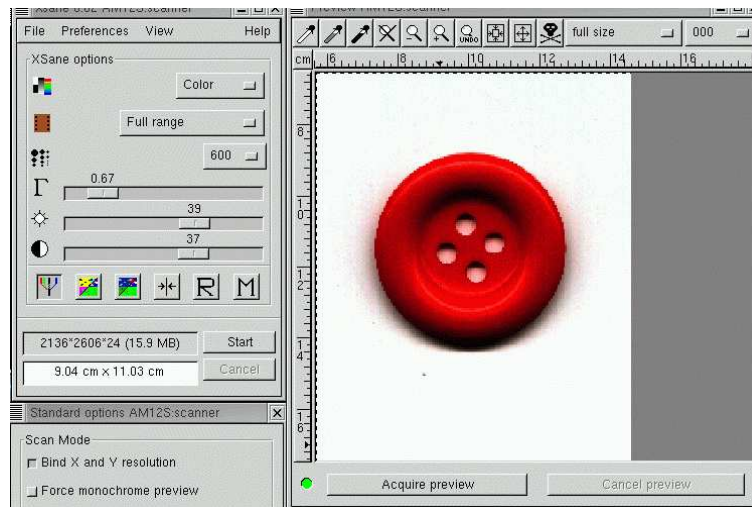
¿Qué os parece nuestro querido Tux pasado por un filtro **Mosaico (Render—Patrón)**?

Es un magnífico programa que a buen seguro entusiasmará a todos aquellos a los que guste el retoque fotográfico o que deseen mejorar sus páginas Web. Con The Gimp podemos modificar una imagen de casi<sup>12</sup> todas las formas posibles.

Si además, tenemos la “suerte”<sup>13</sup> de disponer de un escáner compatible con Linux, podremos “escanear” nuestras imágenes desde el propio Gimp y después modificarlas a nuestro antojo. Sólo si nuestro escáner funciona bien en Linux podremos acceder a esta utilidad desde el menú **Fich—>Adquirir—>Xsane...**


<sup>12</sup>Hay que dejar alguna posibilidad de mejora

<sup>13</sup>Ya se sabe que la suerte hay que buscarla.



desde la primera ventana podremos “previsualizar”/controlar los valores con los que deseamos adquirir nuestra imagen. Como muestra un “botón”.

### 19.3. Image Magick

Es el paquete ImageMagick-5.4.7-5.i386.rpm, se instala por defecto. Se ejecuta con  → **Extras**  
→ **Gráficos** → **ImageMaick** o desde una xterm con  
\$ display



Además de que permite capturar pantallas<sup>14</sup> “es un display de imagen y herramientas de manipulación para el sistema X Window. Puede leer y escribir en formatos JPEG, TIFF, PNM, GIF e imagen Photo CD. Permite cambiar el tamaño, rotar, reducir colores o añadir efectos especiales a una imagen. Permite guardar el trabajo en el formato original o en otro distinto. También incluye comandos para crear gifs animados o transparentes, imágenes compuestas, imágenes detalladas...”

Para desarrollar aplicaciones propias que usen el código ImageMagick o APIs es necesario instalar también ImageMagick-devel.” (RED HAT LINUX 6.1: THE OFFICIAL RED HAT LINUX REFERENCE GUIDE)

Con él podemos también aplicar filtros a las imágenes, efectos, etc. El menú principal del programa es:

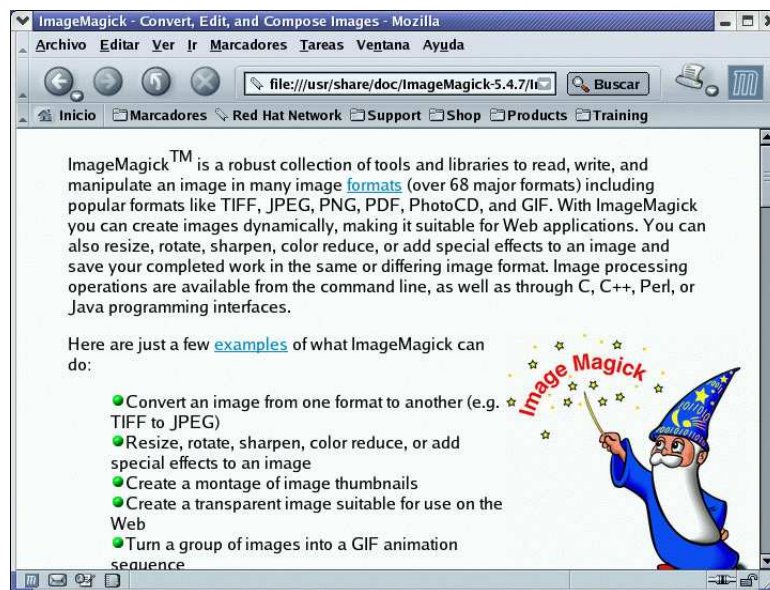
<sup>14</sup>Comando import.



se accede a él cuando tenemos una imagen cargada/capturada o bien pulsando con el botón izquierdo sobre la ventana inicial y su uso es inmediato.

La completa documentación del programa se instala en  
 /usr/share/doc/ImageMagick-5.4.7/

y en los subdirectorios que cuelgan de él. Podemos visualizarla con Mozilla (está en formato html) accediendo al directorio anterior o bien pulsando en el menú principal del programa en **Help**→**Browse Documentation**.



Hacer notar que el programa `convert`, usado en las prácticas forma “parte” de este paquete, y, por tanto, podremos exportar a multitud de formatos cualquier gráfico usando este programa.

## 19.4. xfig

Xfig es una utilidad de dibujo vectorial, es decir, se dibujan objetos y no pixels. Resulta muy útil como complemento de  $\text{LyX/L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  ya que permite guardar las figuras en formato PostScript o directamente en formato  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Está en el CD3 y no se instala por defecto, así que tras montarlo, lo instalamos:

```
# rpm -ivh xfig-3.2.3d-10.i386.rpm
```

la salida que nos aparece es:


```
# rpm -ivh xfig-3.2.3d-10.i386.rpm
warning: xfig-3.2.3d-10.i386.rpm: V3 DSA signature: NOKEY, key ID db42a60e
error: Failed dependencies:
    transfig >= 3.2.3d is needed by xfig-3.2.3d-10
Suggested resolutions:
    transfig-3.2.3d-7.i386.rpm
```

Necesitamos el paquete `transfig` (una versión igual o posterior a la 3.2.3d) para poder instalar `xfig`.

El paquete necesario se encuentra a su vez en el CD3, así que podemos ejecutar

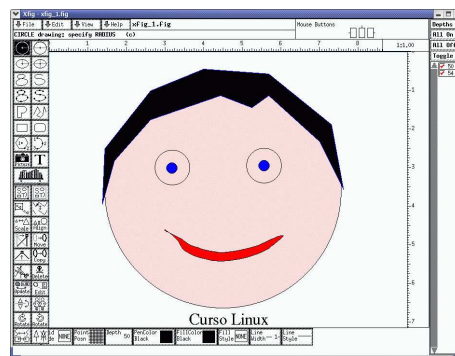
```
# rpm -ivh xfig-3.2.3d-10.i386.rpm transfig-3.2.3d-7.i386.rpm
```

Podemos ejecutarlo con

■  → Extras → Gráficos → Xfig

■ desde una xterm mediante el comando:

```
$ xfig15
```



Cuando instalemos el programa, podemos consultar sobre su uso básico en el fichero

`/usr/X11R6/lib/X11/xfig/xfig-howto.pdf`

no es demasiado extenso (son 20 páginas) y se entiende bastante bien.

Además, en el directorio

`/usr/X11R6/lib/X11/xfig/html`

tenemos un manual bastante amplio<sup>16</sup> de cómo usar el programa, está en formato html. El fichero principal es:

`/usr/X11R6/lib/X11/xfig/html/index.html`

---

<sup>15</sup>Sin el dibujo, claro está

<sup>16</sup>Afortunadamente en versión inglesa. Para los que queráis estudiar la versión en japonés aclarar que también está disponible.



A todos ellos se puede acceder directamente pulsando sobre **Help** en el menú principal del programa.

## 19.5. Qcad

Si queremos hacer CAD, hay un programa que nos puede resolver el problema, se trata de Qcad. La página principal de este programa es:

<http://www.qcad.org>

Desde ella podemos bajar la última versión compilada para nuestra versión de Red Hat, se trata del paquete `qcad-1.5.4-3.redhat.i386.rpm`

Si bajamos a nuestra máquina esta versión y comenzamos su instalación, no se produce ningún problema de dependencias.

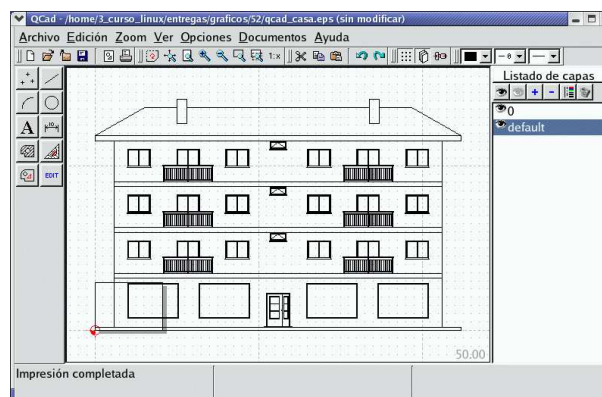
Al ejecutar desde una xterm

```
$ qcad
```

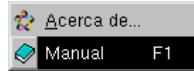
la primera ventana de Qcad nos solicita el idioma



Tras seleccionar el castellano podremos, por ejemplo, crear



El programa viene acompañado de un manual en línea (en inglés) al que se accede desde el menú **Ayuda**, pulsando sobre **Manual** o bien **F1**



➡ **Para practicar:** Podemos conseguir que la ayuda esté en castellano. Para eso, hemos de bajarnos el fichero `qcad_1_5_1_doc_es.zip` que está a nuestra disposición en la sección *downloads* de la página principal del programa. Una vez un nuestra máquina, hemos de desempaquetarlo y descomprimirlo en `/usr/share/qcad/doc`.

```
# cp qcad_1_5_1_doc_es.zip /usr/share/qcad/doc/  
# unzip /usr/share/qcad/doc/qcad_1_5_1_doc_es.zip
```

Cuando ejecutemos de nuevo el programa tendremos la ayuda (para la versión 1.4.8) en castellano.

## 19.6. Blender

Si lo que deseamos es hacer modelado, rendering o animación, disponemos de un magnífico programa libre, se trata de Blender. La página oficial del programa es:

`http://www.blender.org`  
a través de esa misma página podéis bajaros la última versión del programa, en:  
`http://www.blender3d.org/Download/`

Comentar la completa documentación del programa de la página principal. Y sobre todo:

- `http://www.inf-cr.uclm.es/www/cglez/docencia_animacion.html` En esta web hay un manual en castellano sobre su uso.
- `http://3dlibre.hispalinux.es/` Mini guías de cómo trabajar con él.
- `http://difusion.osi.iteso.mx/linux/blender-manual_1_5/` Versión 1.5 del Manual Electrónico 'chapucero' de Blender

Para instalarlo, copiémoslo al directorio de siempre (`/opt`) y descomprimámoslo:

```
#tar -cxvf blender-2.27-linux-glibc2.2.5-i386-static.tar.gz  
como el nombre del directorio es "poco cómodo" para trabajar con él, podemos "abreviarlo":  
#mv blender-2.27-linux-glibc2.2.5-i386-static blender
```

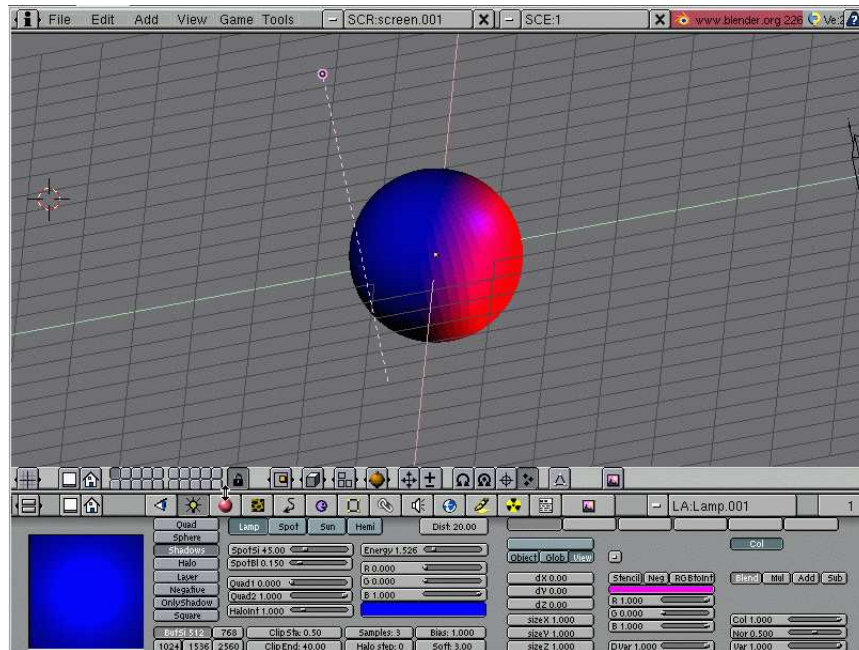
Para acceder al programa ya sólo hay que ejecutar el comando<sup>17</sup>:

```
$/opt/blender/blender
```

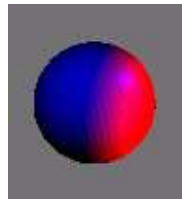
Su uso no es inmediato y para trabajar con él, os remitimos al manual antes comentado, así como a la completa documentación (en Inglés) disponible en la Web del programa.

<sup>17</sup>Lo más lógico es añadir una lanzadera de GNOME






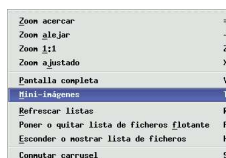
El resultado final del trabajo es una esfera con dos focos de luz:



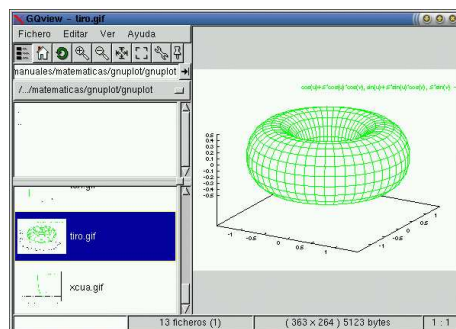
## 19.7. Visor gráfico GQview

Además de las utilidades ya comentadas en esta misma entrega y que incluyen visores gráficos y de la posibilidad de verlos directamente usando Nautilus o Konqueror, hay otra más que merece la pena ver, se trata de GQview

El programa `gqview` se instala por defecto si hemos instalado Gnome (  → **Graficos** → **GQview** ). Permite ver los formatos gráficos más usuales, bien como iconos o bien a tamaño real. Si nos situamos en cualquier directorio y pulsamos en el menú **Ver Mini-imágenes**



veremos iconos de los gráficos en la ventana inferior izquierda de la ventana y si pulsamos sobre uno de estos iconos podremos verlo mejor en el centro de la ventana.



Podemos editar directamente un gráfico desde este programa. Además, desde ese mismo menú podemos configurar todos los editores gráficos que tenemos instalados para elegir en cada momento aquel que más nos interese.



# Capítulo 20

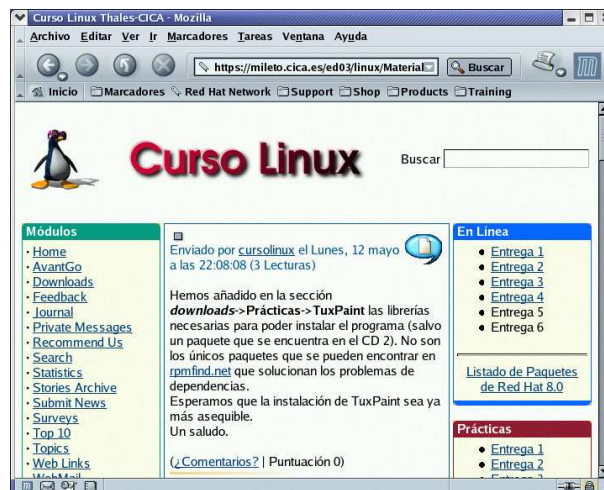
## Internet

Cualquier editor puede crear HTML. No deje que nadie le diga lo contrario. (*HTML 4*, ANNE PHILIPS)

Está claro que Linux se lleva muy bien con Internet, van cogidos de la mano. Los escritorios están bastante bien dotados de herramientas para facilitarnos la labor de conexión<sup>1</sup>. Disponemos de varios navegadores tanto en modo texto (`lynx`) como en modo gráfico (Mozilla, Netscape, Amaya, Konqueror, etc). De hecho, desde que comenzó el curso, seguro que estas herramientas han sido algunas de las más usadas. En este apartado, por tanto, vamos a dar algunas pinceladas que permitan mejorar la “calidad” de nuestra conexión.

### 20.1. Mozilla

De Mosaic y Godzilla sale Mozilla, se trata de un navegador muy configurable reescrito casi entero a partir de la liberación del código fuente de Communicator. Se instala por defecto y se trata de un conjunto de aplicaciones similares al Netscape, y con varias ventajas sustanciales respecto a él: permite múltiples cuentas de correo y, según sus creadores, es un navegador más rápido que Netscape/Explorer.

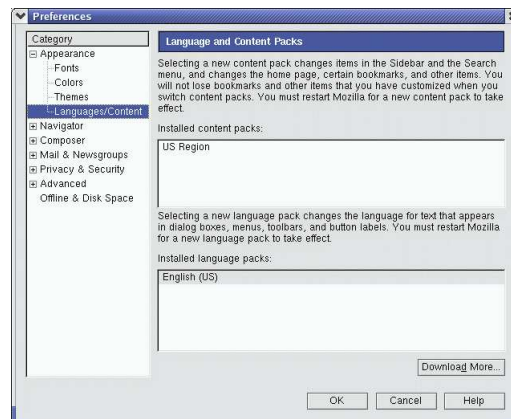


#### ➡ Para practicar

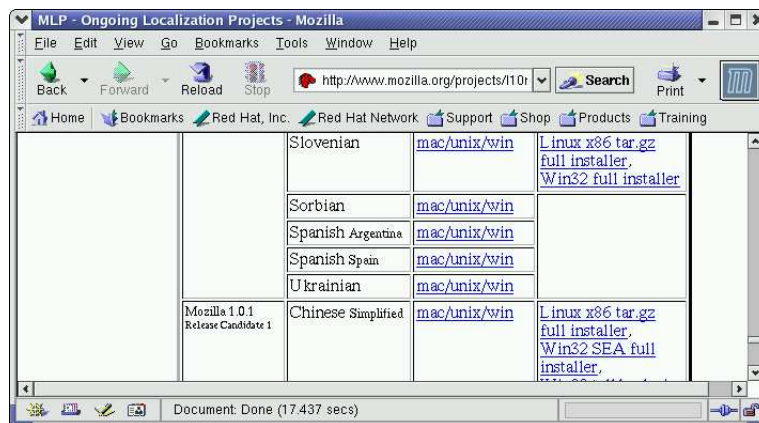
1. Castellanizar Mozilla: una vez conectados a internet (como root), iniciaremos Mozilla y pulsaremos sobre **Edit**→**Preferences**, después optaremos por **Appearance**→**languages/Content**→**Download More..**

---

<sup>1</sup> Se analizaron en la primera entrega



Conectaremos con la Web con módulos de lenguaje para el navegador y nos bajaremos el adecuado para nuestra versión<sup>2</sup>



Una vez bajado el módulo adecuado, tendremos que decirle a Mozilla que lo tome por defecto (esto hay que hacerlo para cada usuario). Reiniciaremos el navegador y de nuevo optaremos por **Edit**→**Preferences** y después **Appearance**→**languages/Content**→**Download More..**. Pero ahora marcaremos el lenguaje Español en ambas casillas.



Para que los cambios sean efectivos tenemos que reiniciar el navegador.

<sup>2</sup>Podemos saber cuál es pulsando sobre **Help**→**About Mozilla**. Si no hemos cambiado nada debe de tratarse de la 1.0.1

2. Añadir *Plugins* a Mozilla: para eso lo mejor es visitar la página Mozilla Plugin Support on Linux (x86) <http://plugindoc.mozdev.org/linux.html>. En ella (y en los foros del curso) están los *plugins* (java, Macromedia Flash, etc) disponibles en la actualidad para este navegador y además se explica la forma de instalarlos.

## 20.2. Editores HTML

Tenemos multitud de editores de HTML para Linux, algunos se han comentado ya al hablar de editores de textos (gnotepad, gXedit), otros son bien conocidos<sup>3</sup> (Composer de Mozilla, Amaya, editor de HTML del OpenOffice, etc). Centrémonos en dos aplicaciones más, específicas para este menester: bluefish y quanta.

### 20.2.1. Bluefish

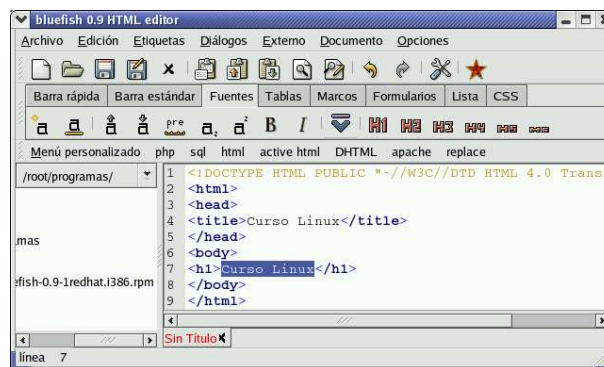
La página principal es

<http://bluefish.openoffice.nl/>

ocupa muy poco y la última versión (bluefish-0.9-1redhat.i386.rpm) la podemos bajar de <http://ftp.europeanservers.net/bluefish/downloads/>

Si instalamos el paquete y ejecutamos el programa en modo gráfico

```
$ bluefish
```



tendremos un entorno para escribir HTML que permite insertar las etiquetas directamente y que nos facilita el trabajo con menús y ventanas para gestionar los atributos de éstas. Además, podemos ver cómo quedaría nuestra página si pulsamos en el menú **Externo** y seleccionamos **Externo Mozilla**<sup>4</sup>. Merece la pena.

### 20.2.2. Quanta

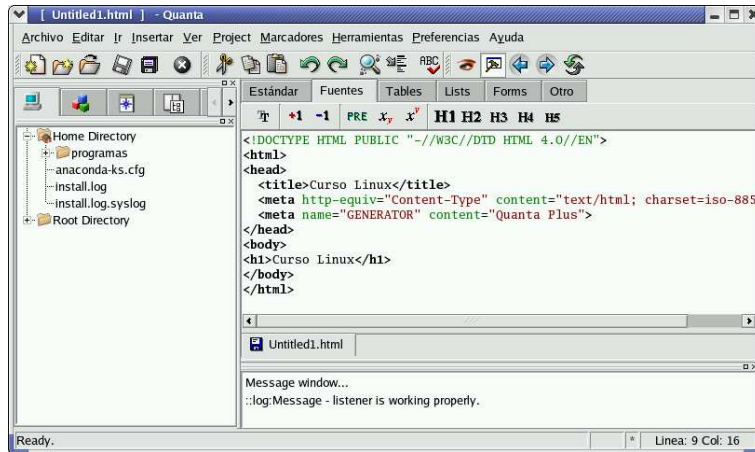
Por fin, uno que viene en los CDs, se trata del paquete del CD3: quanta-3.0-1.pr1.5.i386.rpm

Si lo instalamos y ejecutamos

```
$ quanta
```

<sup>3</sup>Otros se quedan en el tintero: asWedit, screem, etc.

<sup>4</sup>O en el icono correspondiente




podemos disponer de una herramienta valiosa para gestionar proyectos completos de Web. La página inicial del proyecto es: <http://quanta.sourceforge.net>

## 20.3. Varios

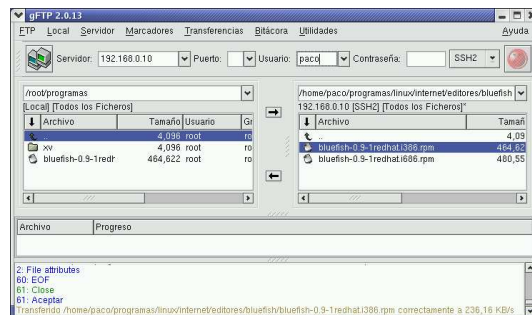
### 20.3.1. FTP: gftp

Con respecto a programas para transferencia de ficheros podemos comentar<sup>5</sup>:

- En modo texto: `ftp`, `rftp` y `ncftp`
- En modo gráfico: sin lugar a dudas `gFTP`

Para acceder al programa<sup>6</sup> podemos pulsar sobre él en el menú  → **Extras** → **Internet** → **Gftp**. También desde una xterm podemos ejecutar:

```
$ gftp &
```



Cuando se cargue veremos que está en castellano y que trabajar con él no es difícil. Para realizar una conexión tan solo tenemos que rellenar los campos indicados: `servidor`, `usuario` y `contraseña`. Cuando se establezca la conexión podremos visualizar/movernos por los directorios del equipo local y del remoto. Además, en las ventanas inferiores podremos ver el estado de la conexión y de las transferencias que estemos realizando.

<sup>5</sup>Recordar que el Midnight Commander nos permite realizar conexiones vía `ftp` de una forma bastante “cómoda” y sin tener que conocer comandos de transmisión de ficheros

<sup>6</sup>Como no es demasiado grande, si trabajáis con él es mejor que lo actualicéis desde:

<http://gftp.seul.org/>

Han salido nuevas versiones que corrigen algunos errores respecto a la que viene con RedHat 8.0.



Permite realizar conexiones seguras usando sftp. Para eso sólo hay que marcar la casilla **Usar sub-sistema SSH2 SFTP** (en la ventana que aparece si pulsamos en **FTP**→**Opciones**→**SSH** del menú principal de la aplicación).

### 20.3.2. wget

wget es un programa que nos permite bajarnos a nuestra máquina todo el contenido de una dirección de Internet. Se instala por defecto (está en el CD1 `wget-1.8.2-3.i386.rpm`):

Veamos con dos ejemplos cómo usarlo. Supongamos que deseamos bajarnos todo el contenido de la URL `http://mileto.cica.es/linux/Material/entrega_4` a nuestro equipo, escribiremos<sup>7</sup>:

```
$ wget -m http://mileto.cica.es/linux/Material/entrega_4
```

Pero claro, ese método tiene un problema y es que hace un rastreo por todos los enlaces y puede que nos bajemos más de lo deseado. Si estamos seguros de que no queremos que suba al directorio padre bajando ficheros escribiremos `-np` (*no parent*), es decir:

```
$ wget -m -np http://mileto.cica.es/linux/Material/entrega_4
```

para ampliar sobre su uso lo mejor es mirar en la *manpage* del programa.

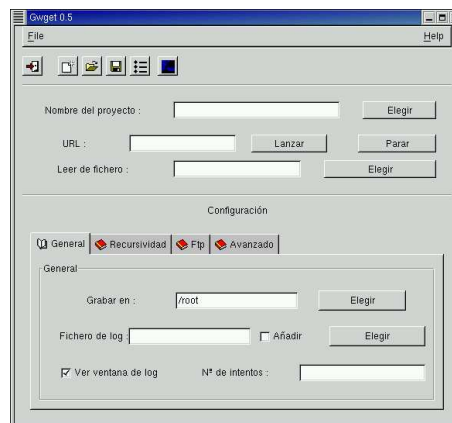
#### gwget

Se trata de un FrontEnd para wget que puede facilitarnos el trabajo, ocupa muy poco y lo tenemos en:

<http://sourceforge.net/projects/gwget/>

Para trabajar con esta versión hay que compilarlo (son las fuentes), por eso trabajaremos con otra versión en formato RPM de Mandrake, más fácil de instalar que la podéis conseguir de:

<http://rpmfind.net/linux/RPM/cooker/contrib/i586/gwget-0.5.2-3mdk.i586.html>



<sup>7</sup>-m es para hacer un *mirror*

# Capítulo 21

## Emuladores

Debe comprender la idea filosófica que subyace bajo el desarrollo del sistema operativo Linux: un mundo mejor donde no importa para qué sistema operativo ha sido desarrollada una determinada aplicación, pues siempre habrá un medio de ejecutarla en Linux. (*Instalación y configuración de Linux*, PATRICK VOLKERDING Y OTROS)

### 21.1. Wine

Wine es un programa que permite ejecutar aplicaciones de Windows desde Linux. La idea de que parte consiste en que, sin salir de Linux, podamos ejecutar cualquier aplicación realizada para Windows. Esto es teoría y, si bien se está desarrollando de forma permanente, todavía le falta bastante para conseguir que ese objetivo sea una realidad.

Para saber qué programas funcionan: <http://www.winehq.com/Apps>

La última versión se puede conseguir en: <http://www.winehq.com/>

De este programa sale una versión nueva casi a “diario”. Si estáis interesados en él lo mejor es bajárselo de la dirección anterior en formato rpm.

Estos apuntes se van a ejemplificar con una versión "arcaica", es el paquete del CD3

wine-20020605-2.i386.rpm

que no se instala por defecto. Así que lo primero es montar el CD3 e instalar el programa. Antes de poder usarlo hemos de adaptarlo a nuestra máquina. Como primer paso debemos editar el fichero `/etc/wine.reg` y tras ponerlo “fino” crear en nuestro directorio de usuario un subdirectorio de nombre `~/.wine`:

```
$mkdir ~/.wine
```

Por ejemplo, veamos cómo quedaría una parte de ese fichero (la que he tenido que tocar en ese equipo) para un sistema con dos sistemas operativos: Windows 98 y Linux. Notad que los destinos de montaje se han puesto tal cual están en el equipo donde se configuró Wine (en `/mnt` casi todos y la unidad C: se monta en `/mnt/win`). De nuevo comentar que no está el fichero completo pero podría ser de la forma:

```
...
[Drive A]
"Path" = "/mnt/floppy"
"Type" = "floppy"
"Label" = "Floppy"
"Serial" = "87654321"
"Device" = "/dev/fd0"

[Drive C]
"Path" = "/mnt/win"
"Type" = "hd"
```

```
"Label" = "MS-DOS"
"Filesystem" = "win95"

[Drive D]
"Path" = "/mnt/cdrom"
"Type" = "cdrom"
"Label" = "CD-Rom"
"Filesystem" = "win95"
; make sure that device is correct and has proper permissions !
"Device" = "/dev/cdrom"

[Drive E]
"Path" = "/tmp"
"Type" = "hd"
"Label" = "Tmp Drive"
"Filesystem" = "win95"

[Drive F]
"Path" = "${HOME}"
"Type" = "network"
"Label" = "Home"
"Filesystem" = "win95"
...
```

La primera vez que ejecutemos wine como ese usuario se copiará el fichero<sup>1</sup> /etc/wine.reg a ~/.wine/config y esa será la configuración con la que trabajaremos. Hacer notar que cada usuario puede modificar después su fichero de configuración a su gusto.

Si montamos la unidad en que la tenemos instalado Windows 9x y ejecutamos desde una xterm  
\$wine /mnt/c/Archivos\ de\ programa/WinZip/winzip32.exe  
veremos que se carga el programa WinZip y que podemos trabajar con él como si de Windows se tratase.



Wine es un programa en constante evolución. Por ahora no funcionan los programas “armatostes” de Windows, pero hay otros que sí<sup>2</sup>.

<sup>1</sup>Entre otros

<sup>2</sup>Por ejemplo, tanto Cabri como Derive funcionan aceptablemente bien.

## 21.2. DosEmu

Por último, para aquellos que sigáis trabajando con MSDOS y programas basados en Dos es obligatorio avisar de la existencia de emuladores (no están en los CDs) de este sistema operativo<sup>3</sup>. No vamos a instalar las últimas revisiones del programa, sino las que aparecieron para la versión 7.2 de Red Hat, se trata de:

dosemu-1.1.1-3.i386.rpm<sup>4</sup> → Emulador de MSDOS para Linux.  
 dosemu-freedos-1.1.1-3.i386.rpm<sup>5</sup> → Necesario si no tenemos instalado el MSDOS<sup>6</sup>.  
 xdosemu-1.1.1-3.i386.rpm<sup>7</sup> → Versión de dosemu para el entrono X.

para saber cómo iniciarse con ellos podéis consultar el HOWTO `Dosemu-Como` y la documentación del programa que se instala en `/usr/share/doc/dosemu-1.1.1`.

Veamos cómo usarlo de una forma “básica”. Para eso primero instalemos los tres programas anteriores:

```
# rpm -ivh *.rpm
Preparing... ##### [100 %]

1:dosemu ##### [ 33 %]

2:dosemu-freedos ##### [ 67 %]

3:xdosemu ##### [100 %]
```

La configuración del programa se realiza con los ficheros

`etc/dosemu.conf` → Configuración del programa para nuestro equipo y necesidades en particular.

`etc/dosemu.users` → Usuarios que pueden ejecutar el emulador.

Para iniciarlo en modo alfanumérico:

```
# dos
```

y dispondremos de un sistema MSDOS en nuestro equipo. Para salir:

```
c:>exitemu
```

Un clásico en acción (WorPerfect 5.1) ejecutado usando el programa `xdos` y con la configuración por defecto<sup>8</sup> del programa:



Otro clásico para algunos centros (Centro2), ejecutado usando `dosemu`

<sup>3</sup>La web del programa es: <http://www.dosemu.org/>

<sup>4</sup><http://rpmfind.net/linux/RPM/rhcontrib/7.2/i386/dosemu-1.1.1-3.i386.html>

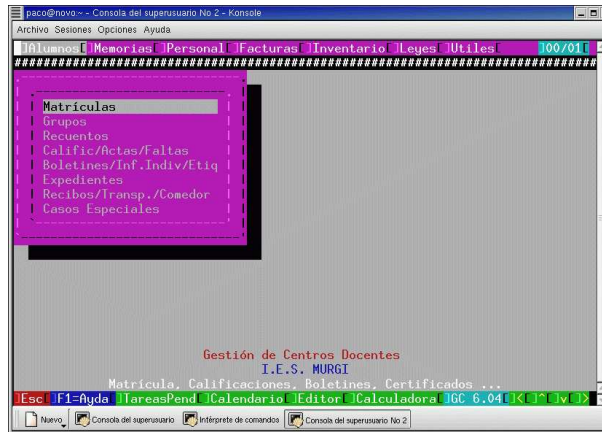
<sup>5</sup><http://rpmfind.net/linux/RPM/rhcontrib/7.2/i386/dosemu-freedos-1.1.1-3.i386.html>

<sup>6</sup>Si queremos podemos usar otra versión de MSDOS, incluso la que trae Windows.

<sup>7</sup><http://rpmfind.net/linux/RPM/rhcontrib/7.2/i386/xdosemu-1.1.1-3.i386.html>

<sup>8</sup>Se ha tenido que modificar el fichero `config.sys` y poner `files=40`.





para conseguir que funcione (además de cambiar el `config.sys`) hay que modificar el fichero de configuración de dosemu (`/etc/dosemu.conf`), en este caso las variables modificadas han sido:

```
$_xms = (10240)
$_ems = (10240)
$_dpmi = (8000)
$_layout = "es"
```

# Capítulo 22

## Textos

*Yo quiero ser escritor*

Hace 20 años le preguntaban a un chaval que qué quería ser de mayor. Él contestaba siempre muy entusiasmado "...yo quiero ser un gran escritor".

Al preguntarle lo que significaba para él exactamente ser "un gran escritor" contestaba : "Es alguien cuyas líneas son leídas por millones de personas. Y no sólo eso, sino que hace reaccionar a la gente, les hace llorar, reír, gritar y emocionarse de muchas maneras".

20 años después este chico ha conseguido su objetivo: Es programador de Microsoft. Escribe los mensajes de error.

Además de las herramientas para escribir textos de los escritorios ya instalados (Gnome<sup>1</sup> y KDE<sup>2</sup>), del editor `mcedit`, etc, vamos a comentar un par de editores fundamentalmente:

**WordPerfect** el de toda la vida en su versión 8

**LyX** una de las sorpresas más gratas que me he encontrado desde que estoy con Linux, es un editor de textos y un procesador de palabras. Lo vemos en el capítulo siguiente.

Antes de comenzar, en el CD de la RedHat disponemos de más editores en modo texto, entre ellos destacar a `jed`, `joe`<sup>3</sup>, `vi`<sup>4</sup> y cómo no `emacs`<sup>5</sup> (los dos últimos se instalan por defecto).

### 22.1. WordPerfect 8

No podía faltar, tenemos a nuestra disposición y de forma gratuita WordPerfect 8. Algunas de las limitaciones que tiene esta versión (al menos que hayamos detectado) es que no tenemos disponible:

- Editor de ecuaciones
- Editor de gráficos
- Manual en Línea

---

<sup>1</sup>`gnp` (Gnotepad), `gedit`.

<sup>2</sup>`kedit`, `kwrite`.

<sup>3</sup>Similar al WordStar. Tanto este editor como el anterior están en el CD3.

<sup>4</sup>El editor UNIX por excelencia

<sup>5</sup>`emacs` es más que un editor de textos creado en lenguaje Lisp. Con `emacs`, además de editar textos, se puede leer y enviar el correo o se puede jugar.

El resto de opciones están todas: corrector ortográfico, fusión, exportación/importación de otros formatos, etc. También disponemos de funcionalidad de Hoja de cálculo: las tablas de WordPerfect permiten trabajar con casi todas las funciones matemáticas más usuales.

Merece la pena destacar que permite leer documentos de Word así como crear documentos en HTML. También que podemos abrir/guardar documentos en otra máquina (publicación remota).

El consumo de recursos de este programa es mucho más reducido que el OpenOffice. Así, en un 486 con 16 MB funciona aceptablemente bien.

### 22.1.1. Instalación

El paquete consta de dos ficheros:

GUILG00.GZ

GUILGES0.GZ

Ambos están a vuestra disposición en la Web del curso. El primero es el programa WordPerfect propiamente dicho y el segundo es el módulo para castellanizarlo.

♠ **Importante** Si intentamos instalar el programa en un sistema RedHat 8.0 nos aparecerán una serie de mensajes de error: se debe a un problema de dependencia de librerías. Para poder llevar a cabo la instalación necesitamos instalar librerías “antiguas”. Los paquetes necesarios son:

```
ld.so-1.9.5-13.i386.rpm
```

```
libc-5.3.12-31.i386.rpm
```

ambos venían en el CD de la RedHat 6.2. En cualquier caso los tenéis en la Web del curso. Una vez instalados:

```
# rpm -ivh ld.so-1.9.5-13.i386.rpm
```

```
# rpm -ivh libc-5.3.12-31.i386.rpm
```

podremos comenzar la instalación de Wordperfect.

Como ya se ha comentado, y para unificar un lugar donde instalar, lo vamos a hacer en el subdirectorio /opt. Para instalar el programa vamos a usar un subdirectorio auxiliar llamado /opt/w, cuando termine la instalación lo borraremos. Por tanto, lo primero que haremos será crear ese directorio:

```
#mkdir /opt/w
```

y copiar ambos ficheros a este subdirectorio. Aunque la extensión no lo indica con claridad son tar.gz y como tales hay que descomprimirlos/desempaquetarlos, es decir, una vez copiados en ese subdirectorio, escribiremos desde /opt/w:

```
# gunzip GUILG*
```

Si después usamos

```
# ls
```

```
GUILG00  GUILGES0
```

Lo siguiente es desempaquetarlos:

```
# tar -xvf GUILG00
```

```
Readme
```

```
Runme
```

```
b_ins00
```

```
b_ins01
```

```
b_req00
```

```
b_req01
```

```
b_us00
g_req00
g_req01
g_req02
g_req03
g_us00
g_us01

# tar -xvf GUILGES0
Readme
Runme
b_es00
b_ins00
b_ins01
b_us00
g_es00
g_us00
g_us01
```

## Sigamos

Tras esto, y desde una xterm<sup>6</sup> ejecutaremos

```
# ./Runme
```

```
Did you unzip and untar the files you downloaded? (y/n) y
```

```
Extracting Files...
```

Tras pulsar “y” ya sólo tendremos que seguir las instrucciones que nos van a ir apareciendo. En primer lugar veremos un gráfico que permite ver la licencia del programa en distintas lenguas.

Comentar que la versión de WordPerfect que estamos instalando es una versión de prueba que “caduca” a los 90 días. Para solventar esta limitación hay que registrarse y obtener una clave en la página de Corel:

<http://venus.corel.com/nasapps/wp8linuxreg/register.html>

tan sólo tenemos que rellenar un formulario y vía e-mail recibiremos el número que nos permitirá registrar el programa y trabajar con él de forma indefinida. Hay que dejar bien claro que este proceso no nos va a costar nada (salvo el tiempo de conexión a Internet). En cualquier caso, siempre tendremos al menos esos 90 días para probar el producto.

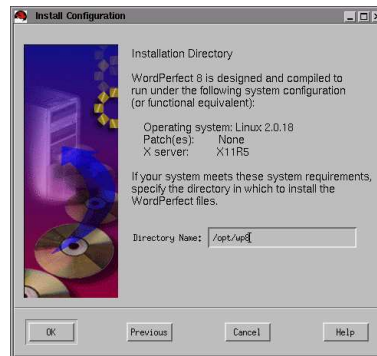
Tras aceptar en la ventanas que nos permiten ver la licencia en castellano



<sup>6</sup>También se puede instalar en modo texto pero no es recomendable ya que aunque las preguntas son las mismas, el “entorno” es mucho más “incómodo”.

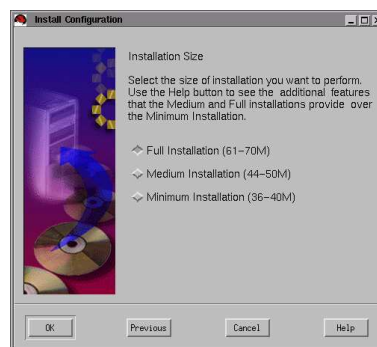
Notar que hemos antepuesto “./” para indicar a la shell que el programa a ejecutar está en el subdirectorio actual. Si no lo ponemos, no encontrará el fichero salvo que tengamos la variable PATH configurada adecuadamente.

veremos una que nos permite seleccionar el lugar donde vamos a instalar el programa y los requisitos mínimos para trabajar con él. En el caso de la RedHat 8.0 y tras instalar los paquetes comentados no hay ningún problema. El directorio elegido para realizar la instalación va a ser `/opt/wp8`<sup>7</sup>



Una vez seleccionado el directorio en donde instalar el programa, elegiremos el tipo de instalación. Si tenemos espacio en disco suficiente lo mejor es seleccionar la instalación completa. Los distintos tipos de instalación se caracterizan por:

- Instalación mínima: configuración mínima para ejecutar el programa
- Instalación media: es como la primera, pero añadiendo el diccionario de sinónimos<sup>8</sup>.
- Instalación máxima: además de lo anterior dispondremos de
  - La ayuda del programa (en castellano)
  - Ejemplos de macros.
  - Más fuentes, etc.

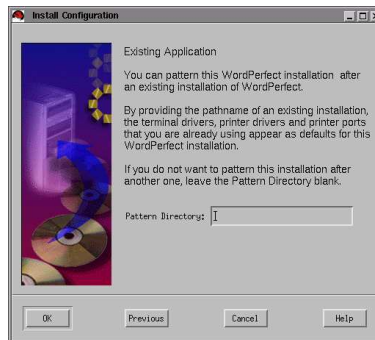


Si ya teníamos una versión anterior de WordPerfect instalada, indicaremos en el campo de la ventana siguiente, el lugar donde está; así podremos aprovechar la configuración de la versión anterior. En el caso contrario, tan solo aceptaremos dejando el campo en blanco:

---

<sup>7</sup>Claro está que esto es opcional

<sup>8</sup>El corrector gramatical no está disponible para esta versión gratuita



Después veremos la ventana:



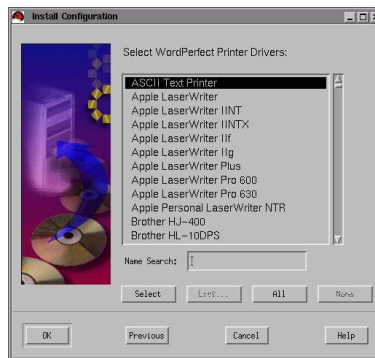
si seleccionamos la opción **Update the magic file** actualizaremos el fichero `/etc/magic`: con este fichero asociaremos las extensiones de los ficheros de WordPerfect a este programa. No es necesario seleccionarlo y podemos seguir sin modificar nada.

Después seleccionaremos el lenguaje en el que vamos a instalar el programa. Por defecto aparece marcado en inglés americano. Podemos marcar que sólo instale en castellano o ambas opciones.

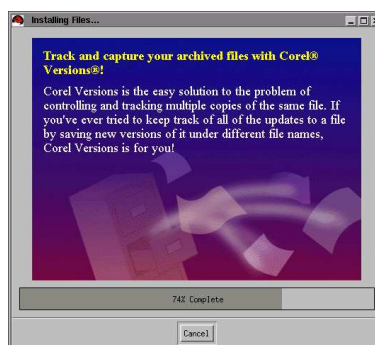


Una vez seleccionado el idioma que más nos interese, proseguiremos con la instalación. Si en algún punto del proceso nos hemos equivocado o deseamos modificar algo ya realizado siempre podemos ir hacia atrás y cambiar esa opción.

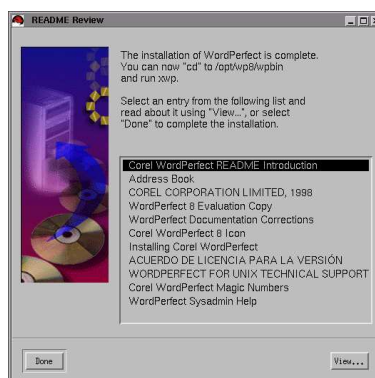
Ahora es el momento de seleccionar la impresora. No deberíamos perder el tiempo aquí ya que las veces que lo hemos instalado<sup>9</sup> este paso no ha servido para nada y hemos tenido que configurarla otra vez después de instalado el programa. Así que lo mejor es continuar el proceso pulsando **OK** y ya la instalaremos.



Continuemos: tras aceptar en un par de ventanas más, comienza el proceso de instalación propiamente dicho. Podremos seguir este proceso por una barra que nos va indicando cómo se desarrolla.



Tras un par de minutos y tras pulsar sobre **Done**



tendremos WordPerfect instalado en nuestro equipo y ya podemos borrar el subdirectorio auxiliar y todos los ficheros que contiene:

```
# rm -r /opt/w
```

<sup>9</sup>Aunque seguro que debe de existir alguna Ley de Murphy que diga que si lo instaláis en vuestro equipo sí que funciona

### 22.1.2. Inicio de WordPerfect. Selección de la Impresora.

El binario que permite ejecutar el programa es `/opt/wp8/wpbin/xwp`<sup>10</sup>, por tanto, para usarlo y salvo que creamos una entrada en el escritorio/gestor de ventanas que usemos tendremos que escribir

```
$ /opt/wp8/wpbin/xwp
```

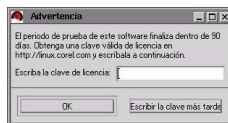
En cualquier caso, no está de más que añadamos el path del programa a la variable PATH del sistema para así poder entrar en él escribiendo sólo

```
$ xwp &
```

esto lo podemos conseguir, por ejemplo, editando el fichero `/etc/bashrc` y modificando o añadiendo la línea

```
export PATH=$PATH:/opt/wp8/wpbin
```

Sea cual sea la forma de entrar en el programa, la primera vez que lo hagamos veremos la ventana

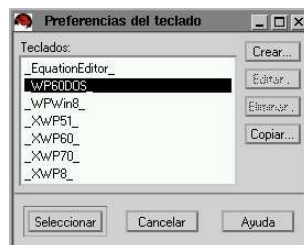


en ella se nos pide que introduzcamos la clave obtenida tal cual se ha explicado antes. Si aún no la hemos conseguido, no pasa nada. Pulsaremos sobre **Escribir la clave más tarde** y comenzaremos a trabajar con WordPerfect.

Lo primero que llama la atención es que se nos abrirán dos ventanas. Con la primera y más pequeña podemos controlar la aplicación,



con ella podemos abrir una nueva ventana para abrir otro documento o configurar en su totalidad la aplicación. Una de las opciones que seguro será del agrado de los que hayáis trabajado con el WordPerfect 5.1 es que podemos optar por seguir trabajando con el teclado del WordPerfect para MSDOS<sup>11</sup>. Para conseguir esto pulsaremos en **Preferencias**, después sobre el icono de **Teclado** y por último optaremos por la combinación de teclado que más nos guste



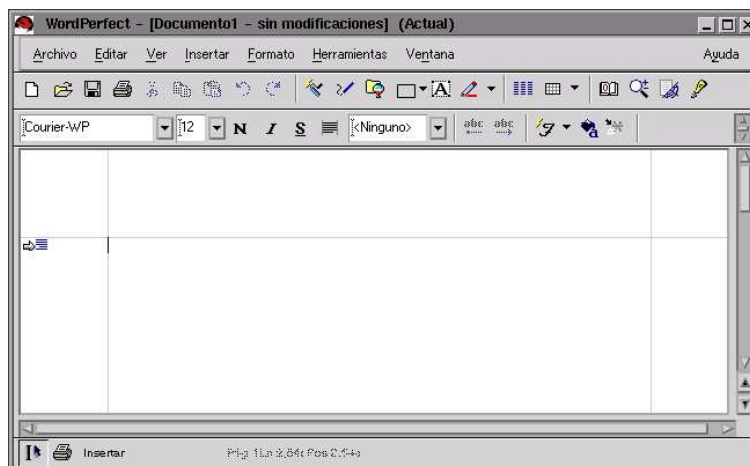
<sup>10</sup>Claro está, en el supuesto de que lo tengamos instalado en `/opt/wp8`, si no es así hay que cambiar la ruta por la que nosotros hayamos elegido

<sup>11</sup>Es muy parecido a las combinaciones de teclas del WordPerfect 5.1




El resto de cuestiones son las típicas de configuración y para saber cómo funcionan os remitimos a la ayuda del programa. Como ya se ha comentado, está en castellano y podemos acceder a ella pulsando en esta misma ventana en **Ayuda** y después en **Contenido**<sup>12</sup>.

La otra ventana es la que nos permite escribir textos, su configuración no difiere en casi nada de su hermano para Windows y dispone de las opciones típicas de cualquier tratamiento de textos:



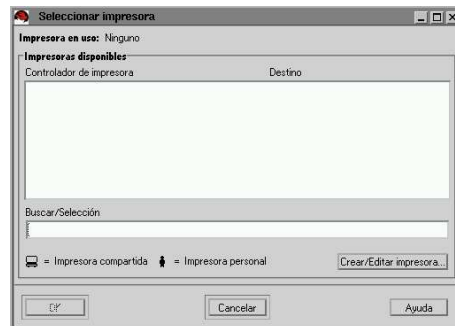
Comentemos una que está a nuestra disposición en toda la familia desde la versión 8 de WordPerfect, consiste en lo que se denomina *shadow cursor* es decir, la posibilidad de situar el puntero del ratón en cualquier zona de la pantalla para poder escribir allí. Tan sólo tenemos que pulsar con el ratón en donde queremos escribir y listo, en ese punto se puede introducir el texto.

Durante el proceso de instalación hemos dejado pendiente la impresora, veamos ahora cómo hacerlo. Para configurar la impresora pulsemos sobre la ventana de edición en el icono  (**Archivo Imprimir**), veremos una ventana como la que sigue

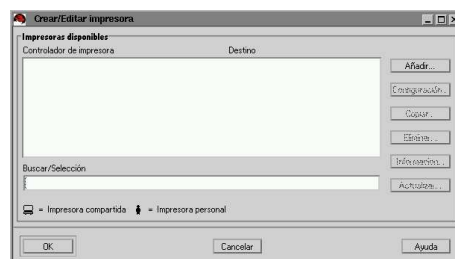


si pulsamos sobre **Seleccionar**

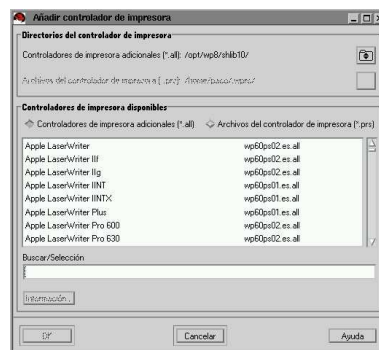
<sup>12</sup>El manual en línea no está disponible en esta versión gratuita



se nos abrirá una ventana en la que pulsaremos sobre **Crear/Editar Impresora**



en la nueva ventana pulsaremos sobre **Añadir**, y tendremos a nuestra disposición un listado con las impresoras soportadas por el programa

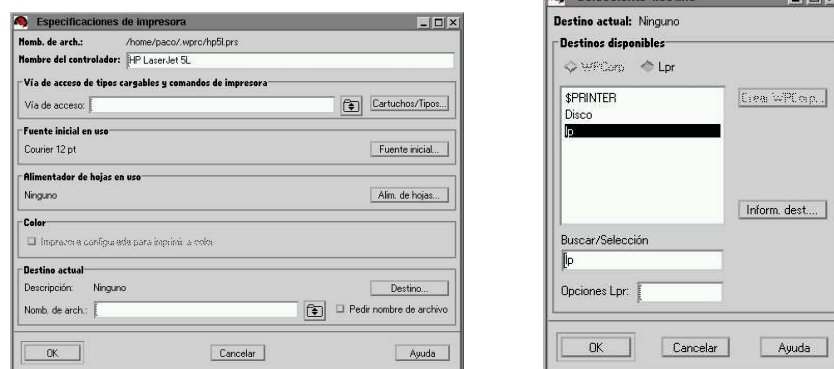


cuando hayamos elegido la nuestra y aceptemos pulsando sobre **OK** tendremos que decir a WordPerfect el destino de la impresora seleccionada. Merece la pena comentar que no hay controlador para las impresoras de la gama *Epson Stylus* y que en este caso tenemos que seleccionar de la lista la opción

### **Passthru PostScript**

para que podamos usar la configuración disponible en el fichero `/etc/printcap`

Una vez seleccionado el controlador de la impresora pulsaremos sobre **Configuración—Destino** para asignar el destino al driver de impresión antes seleccionado



En general, y salvo que tengamos varias impresoras conectadas a nuestro equipo elegiremos la impresora por defecto del sistema seleccionando **lp**. Ya sí, si aceptamos varias veces sobre **OK** hasta cerrar todas las ventanas de configuración de la impresora, la tendremos configurada para WordPerfect y ya podremos imprimir nuestros documentos.

## 22.2. Otros

## 22.3. Abiword

Este programa está ya instalado por defecto, pero si queremos bajarnos la última versión debemos buscarla en:

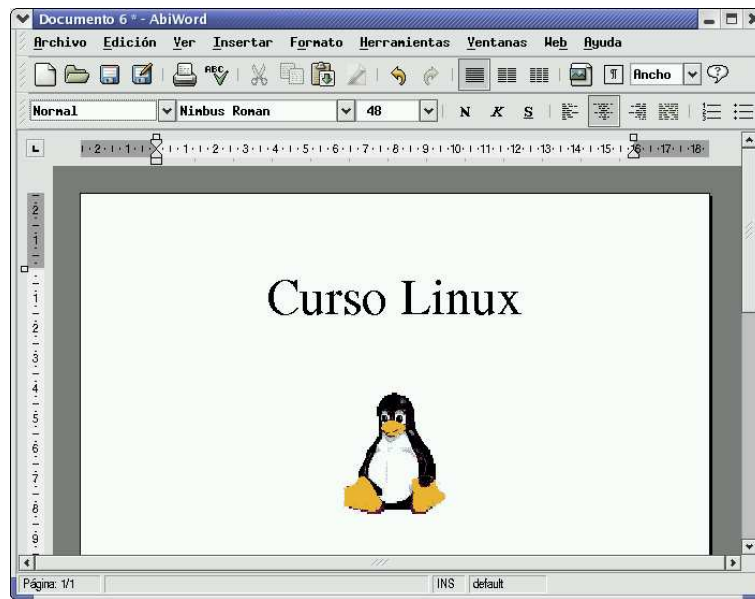
<http://www.abisource.com/>

Entre sus virtudes destacan:

- Está en castellano.
- No necesita grandes recursos para funcionar
- Tiene todo lo básico necesario para escribir textos sin mayores pretensiones.
- Permite importar ficheros de Word, WordPerfect y de  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , aunque todavía no están todo lo depurados que sería de esperar.

Este programa está disponible también para Windows y se espera que en el futuro se complete con una Hoja de Calculo (**AbiCalc**), un gestor de bases de datos (**AbiFile**) y un programa de presentaciones (**AbiShow**).

Si ejecutamos el programa veremos una pantalla como la de la figura



Su manejo es inmediato y si bien aún no está suficientemente documentado, para equipos con pocos recursos puede ser una buena elección.

## 22.4. Vi no, gvim

Otro clásico del mundo UNIX es `vi`, el editor por excelencia, en nuestro caso no vamos a comentar casi nada sobre él. En todos los libros y revistas de Linux hay algunas reseñas sobre su uso. Además de la documentación del programa (página man y ficheros del directorio `/usr/share/doc/vim-common-6.1`), hay una FAQ sobre `vi` en la página de Lucas para los que queráis saber cómo funciona, está en:

<http://lucas.hispalinux.es/htmls/faq.html>

Una demostración de que Linux está cambiando y que se acerca cada vez más a todos los usuarios es la versión para las X de este editor que viene en el CD3, seguro que no será del agrado de los más “puristas” pero merece la pena instalarlo y echarle un vistazo:

```
# rpm -ivh vim-X11-6.1-14.i386.rpm
```

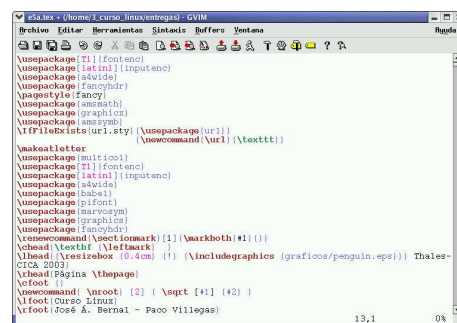
```
vim-X11
```

```
#####
```

El comando a ejecutar desde una xterm es:

```
$ gvim
```

y el resultado es:



De igual manera que con `vi`, para entrar en modo de edición pulsaremos sobre:

**i** para insertar texto antes del cursor<sup>13</sup>

**ESC** para volver al modo de comando

*!Esto no es vi, mi vi me lo han cambiado*¡, permite cortar y pegar, manejar el ratón como ya estamos habituados.

---

<sup>13</sup>Pulsando en la letra “a” se obtiene un efecto parecido, la diferencia reside en que en este caso añadimos texto después del cursor.

## Capítulo 23

# Programas para L<sup>A</sup>T<sub>E</sub>X

Estimado lector, esta es una obra sobre ¿cómo ha dicho?, ¿látex?, ¿qué es L<sup>A</sup>T<sub>E</sub>X? Es una buena pregunta, pero la respuesta ha de ser necesariamente larga y no siempre sencilla.

.....

está usted a las puertas de una hermosa aventura intelectual y científica, técnica y artística, originada por un hombre allá por el año 1977. (*L<sup>A</sup>T<sub>E</sub>X una imprenta en sus manos*, BERNARDO CASCALES SALINAS Y OTROS)

### 23.1. L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X es un sistema de preparación de documentos creado por Leslie Lamport tomando como base el programa T<sub>E</sub>X diseñado por DONALD KNUTH. Su fuerte se centra en la composición profesional de textos matemáticos y/o científicos. Con L<sup>A</sup>T<sub>E</sub>X establecemos qué queremos que aparezca en el documento y no cómo debe aparecer.

Montemos el CD1 de la distribución y situémonos en el subdirectorio donde están los paquetes. Si vemos un listado de los paquetes que empiezan por “tetex” tendremos:

```
$ ls tetex*
tetex-1.0.7-57.i386.rpm      tetex-dvips-1.0.7-57.i386.rpm
tetex-afm-1.0.7-57.i386.rpm tetex-fonts-1.0.7-57.i386.rpm
tetex-latex-1.0.7-57.i386.rpm tetex-xdvi-1.0.6-57.i386.rpm
```

Para tener un sistema con L<sup>A</sup>T<sub>E</sub>X (T<sub>E</sub>X) completo<sup>1</sup> debemos tenerlos instalados todos<sup>2</sup>: la suma de los paquetes anteriores representa aproximadamente 90 MB.

Para poder instalar todos los paquetes (en realidad es por el paquete `tetex-xdvi`) hemos de instalar previamente el paquete del CD1 `w3c-libwww-5.4.0-1.i386.rpm`. Así que, una vez montado el CD1 y tras situarnos en el directorio que contiene los paquetes:

```
# rpm -ivh w3c-libwww-5.4.0-1.i386.rpm
```

Después instalamos L<sup>A</sup>T<sub>E</sub>X

```
#rpm -ivh tetex*
```



Hay dos paquetes que es conveniente actualizar, ya que han salido versiones que corrigen algunos problemas (por ejemplo al crear o visualizar pdfs). Se trata de:

---

<sup>1</sup>No todos son “imprescindibles”.

<sup>2</sup>En el CD3 está el paquete `tetex-doc-1.0.7-57.i386.rpm` que contiene documentación sobre L<sup>A</sup>T<sub>E</sub>X y que no es necesario instalar. Este paquete contiene la ayuda/documentación en inglés y ocupa 15 MB, así que si estamos escasos de disco podemos obviarlo.

```
tetex-1.0.7-57.1.i386.rpm
```

```
tetex-dvips-1.0.7-57.1.i386.rpm
```

o sea que los bajamos (de la Web del curso o de <https://rhn.redhat.com/errata/RHSA-2002-194.html>) y actualizamos las versiones instaladas `rpm -U paquete`.

Para trabajar con  $\text{\LaTeX}$ <sup>3</sup> se escribe el fichero en formato  $\text{\TeX}$  con un editor de texto cualquiera y después se ejecuta (dos veces)

```
$ latex fichero.tex
```

El resultado es un fichero de texto formateado de extensión `.dvi`<sup>4</sup> que se puede visualizar con `xdvi`.

Más información sobre  $\text{\LaTeX}$  en:

- Podemos bajarnos un manual que está bastante bien en:

```
http://carina.fcaglp.unlp.edu.ar/~benaglia/otros/print/latex/ldesc2e.ps.gz
```

es el fichero `ldesc2e.psz`.

- Para ampliar sobre el tema se puede mirar en

```
http://lucas.hispalinux.es/CervanTeX/CervanTeX.html
```

un documento que puede aclarar bastantes conceptos sobre  $\text{\LaTeX}/\text{\TeX}$  es la FAQ de Cervan $\text{\TeX}$ , que está a vuestra disposición en la web anterior.

- Por último comentar un libro muy bueno sobre  $\text{\LaTeX}$  editado por AULA DOCUMENTAL DE INVESTIGACIÓN llamado  $\text{\LaTeX}$ , UNA IMPRENTA EN SUS MANOS.

Existen conversores de los programas más usuales de textos a  $\text{\LaTeX}$ , por ejemplo en:

```
http://www.penguin.cz/~fojtik/wp2latex/wp2latex.htm
```

hay una utilidad para convertir ficheros de WordPerfect a  $\text{\LaTeX}$ , está disponible para varias plataformas.

Veamos sólo un pequeño ejemplo sobre su uso. Para esto con un editor de textos podemos escribir:

```
%% Propuesta de Curso: Thales-CICA 2002-2003

\documentclass[a4paper,10pt]{article}
\usepackage[spanish]{babel}
\usepackage{graphics}

\title{Propuesta de Curso Thales-CICA 2002-2003: \\
{\bf\em Linux como entorno de trabajo en un centro educativo } }

\author{ Jos\'e \'Angel Bernal \and \resizebox*{0.75cm}{!}
{\includegraphics{penguin.eps}} \and Paco Villegas }
\date{}

\begin{document}

\maketitle

\tableofcontents{}

\section{T\'itulo}

\begin{center}
Linux como entorno de trabajo en un centro educativo
```

<sup>3</sup>Os recomendamos el curso que sobre él se está dando este año.

<sup>4</sup>*DeVice Independent*.

```

\end{center}

\section{Profesorado}

\vspace*{0.5cm}

{\em Francisco Villegas Mart}\'{{i}n}
\footnote{Profesor de Matem\'aticas en el I.E.S. Murgi de El Ejido\\
{\tt pvillegas@iesmurgi.org}}
\vspace*{0.5cm}
{\em Jos\'e Angel Bernal Bermejo}
\footnote{Servicio de Coordinaci\'on Inform\'atica de la Junta de Andaluc\'ia
\\
Grupo de Investigaci\'on Julietta de la Universidad de Sevilla\\
{\tt jabernal@thales.cica.es}}

\section{Duraci\'on Docente}
\begin{equation}
\frac{120}{2}=\sqrt{3600}=\left(\sum_{i=1}^{\infty}\frac{1}{n^2}\right)^{-1}
\cdot 10 \cdot \pi^2 = 60 \text{ \mbox{ horas}}
\end{equation}

\section{Objetivos del curso}

La idea de este curso consiste en dotar al profesorado de las
herramientas b\'asicas para trabajar con un
PC que tenga como sistema operativo Linux (Red Hat 8.0) \dots

A un nivel operativo, los principales objetivos del curso son:

\begin{itemize}
\item Romper el {\em miedo inicial} a la instalaci\'on de Linux en un PC
\item \dots
\end{itemize}

\section{Programa del curso}
\begin{enumerate}
\item Sesi\'on inicial
\begin{enumerate}
\item ¿Qu\'e es Linux? Posibilidades de Linux. Estructura del Sistema
Operativo
\item Ap\'endices:
\begin{enumerate}
\item Informaci\'on en l\'{\i}nea
\end{enumerate}
\end{enumerate}
\item \dots
\end{enumerate}
Para terminar este fichero de ejemplo, la funci\'on seno obtenida con el programa
gnuplot:

\input{sin.tex}
\end{document}

```

En este fichero aparecen algunas de las posibilidades que ofrece L<sup>A</sup>T<sub>E</sub>X:

- Tabla de contenidos.
- Notas a pie de página.





- Fórmulas matemáticas.
- Inclusión de gráficos (de dos formas distintas).
- Listas
- ...

Si este texto lo almacenamos con el nombre `fichero.tex` y ejecutamos la serie de comandos:

```
$ latex fichero.tex
```

```
$ latex fichero.tex
```

Tendremos (entre otros) el fichero `fichero.dvi` que podremos visualizar (en modo gráfico) usando:

```
$ xdvi fichero.dvi
```

también podemos pasarlo a otros formatos con:

```
$ dvips fichero.dvi -o fichero.ps
```

```
$ dvi2pdf fichero.dvi fichero.pdf
```

#### ➡ Para practicar.

El texto anterior en  $\text{\LaTeX}$  (así como la gráfica) están a vuestra disposición en la WEB del curso (`latex.tgz`). Si la bajáis a vuestra máquina podréis comprobar el resultado de ejecutar los comandos anteriores.

Para saber cómo pasarlo a HTML sólo hay que seguir leyendo.

## 23.2. Ficheros en formato pdf/ps.

### 23.2.1. Formato PostScript

En nuestra máquina tenemos ya instalados los visores<sup>5</sup>:

- `gv`
- `kghostview`
- `kdvi`

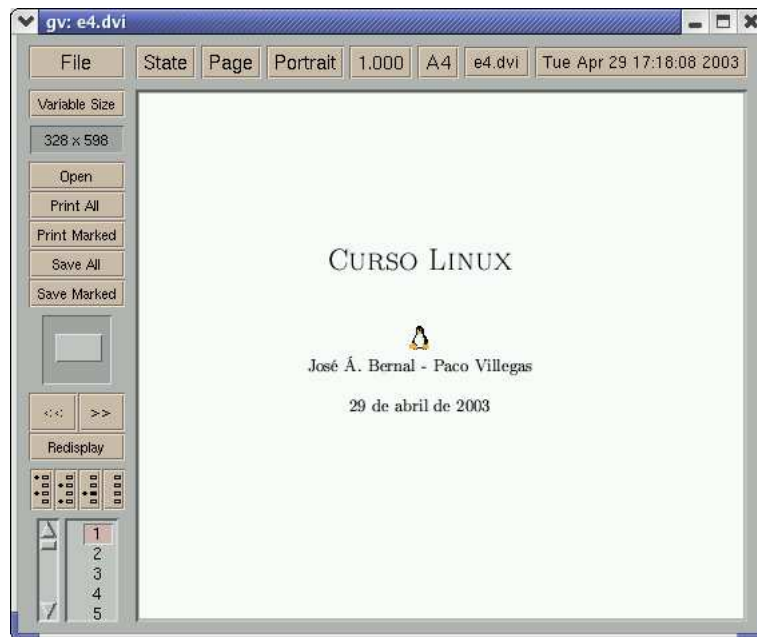
Optar por uno u otro es cuestión de gustos.

El que se instalaba en versiones anteriores como visor PostScript por defecto es otro visor más contenido en el CD3, se trata del paquete `gv-3.5.8-18.i386.rpm`. Una vez instalado podemos ver su aspecto ejecutando en modo gráfico:

```
$ gv
```

---

<sup>5</sup>Los dos últimos se instalan con el KDE



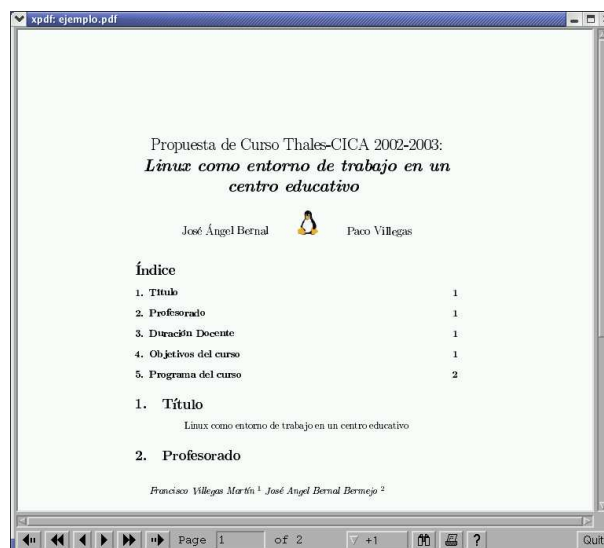
Hay actualizaciones de los visores gv y ggv, podemos bajarlas de: <https://rhn.redhat.com/errata/RHSA-2002-207.html>

### 23.2.2. Formato pdf

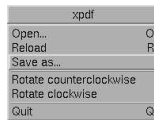
En el CD1 de la distribución tenemos el paquete `xpdf-0.92-5.i386.rpm` que debe estar instalado ya en nuestra máquina. Si ejecutamos en modo gráfico:

```
$ xpdf
```

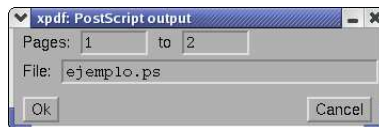
veremos una ventana como la que sigue (sin el fichero)



Para acceder al menú de contexto pulsaremos sobre la ventana del programa con el botón derecho del ratón:



Abrimos nuestro fichero pulsando sobre **Open** y seleccionando el fichero `ejemplo.pdf` obtenido en la práctica anterior podremos visualizar su contenido. Para imprimirlo pulsaremos sobre el icono en el que se representa una impresora y confirmaremos en el menú:



las páginas a “imprimir” y el nombre y ruta del fichero de destino. Con este proceso lo que hemos creado es un fichero PostScript que ya sí podemos imprimir usando el visor `gv`, `ggv`, ...

También podríamos usar directamente la utilidad `pdf2ps` que forma parte del paquete `ghostscript`<sup>6</sup> para transformar este fichero en formato pdf a formato ps:

```
$ pdf2ps ejemplo.pdf ejemplo.ps
```

y después mediante `GostView` imprimirlo<sup>7</sup>.

#### ➤ Para practicar: Acrobat Reader

Otra opción para visualizar e imprimir ficheros en formato pdf es bajarnos la última versión del visor de Adobe para Linux (<http://www.adobe.com/products/acrobat/readstep2.html>), si optamos por un paquete para Red Hat podemos bajar:

```
http://rpmfind.net/linux/RPM/Madeinlinux/packages/RPMS/acroread-5.06-1.2mlx.i386.html
```

Si los bajamos a nuestro equipo y los instalamos, con solo escribir desde una xterm:

```
$ acroread
```

... no funciona. Otra vez las fuentes UTF, y es que el visor de Adobe no puede trabajar con ellas. Para solucionarlo tenemos que cambiar el fichero `/etc/sysconfig/i18n` tal cual se comentaba en la entrega 3, apartado 2.3.1 Ficheros de configuración del sistema, es decir, tenemos que modificar ese fichero como sigue (y reiniciar el equipo):

```
$ cat /etc/sysconfig/i18n
LANG="es_ES@euro"
SUPPORTED="es_ES@euro:es_ES:es"
SYSFONT="lat0-sun16"
SYSFONTACM="iso15"
```

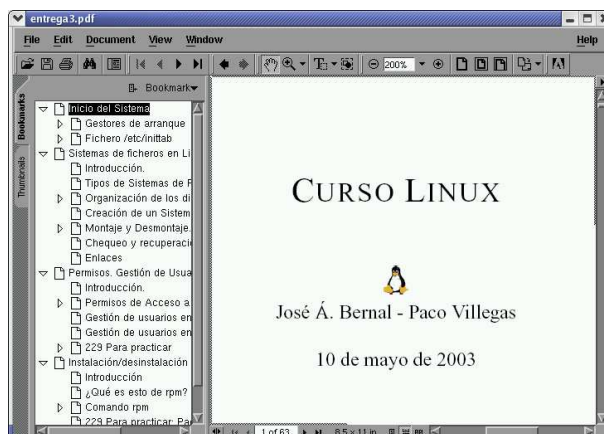
Ya podremos ver e imprimir perfectamente ficheros pdf. Por ejemplo, una entrega del curso tendría la forma:

---

<sup>6</sup>Se instala por defecto.

<sup>7</sup>En realidad no hace falta ningún visor para imprimir un fichero `.ps`, tan sólo hay que usar:

```
$ lpr fichero.ps
y fichero.ps se imprimirá.
```



### 23.3. LyX

Si  $\text{\LaTeX}$  nos parece muy duro, podemos usar una herramienta más intuitiva para escribir nuestros documentos, se trata de LyX.

¿Qué se puede decir de LyX?, ¿qué es LyX?, ¿qué se puede hacer con él?, todas estas preguntas se pueden contestar con una sola respuesta: **todos los apuntes de este curso se han realizado con él**. Con cada entrega estáis viendo las posibilidades de este magnífico programa.

Según el Tutorial de LyX<sup>8</sup> es “un interfaz casi WYSIWIG (*What You See Is What You Get*) para  $\text{\LaTeX}$ ” y SGML. Como se puede deducir de la frase anterior para poder trabajar con LyX es imprescindible tener instalado  $\text{\LaTeX}$ . LyX permite componer documentos siguiendo la filosofía de  $\text{\LaTeX}$  pero sin tener que conocer comandos de  $\text{\LaTeX}$ <sup>9</sup>. Con LyX nos centramos en lo que queremos escribir y no en cómo hacerlo. El proceso de edición y composición final es responsabilidad de  $\text{\LaTeX}$ .

Donde más sobresale LyX es la composición de textos matemáticos<sup>10</sup>, tras trabajar con varios editores no conozco una forma más rápida y sencilla de introducir expresiones matemáticas con un resultado tan espectacular. Si a esto añadimos la facilidad con que podemos insertar un gráfico, el manejo de tablas, las notas en las páginas, etc, nos encontramos ante una herramienta que sin duda merece la pena instalar en nuestro equipo.

**Antonio Zugaldía Rodríguez** expone en LyX: UN PROCESADOR DE TEXTOS WYSIWYM

*“Si bien la idea de  $\text{\LaTeX}$  es crear documentos profesionales con el mínimo esfuerzo, la idea cobra realmente sentido cuando se introduce LyX en escena. Aquí tienes 20 razones para usar LyX.*

1. *Interfaz gráfica para acceder a todas las funciones (incluidos los típicos shortcuts).*
2. *Operaciones estándar de otros procesadores de textos: cortar/pegar, múltiples documentos abiertos, infinitos deshacer/rehacer, corrector ortográfico.*
3. *Diferentes plantillas de documentos para escribir cartas, artículos, libros, guiones, LinuxDoc... También se incluyen plantillas para sociedades científicas como la AMS, APS, IEEE, o para revistas específicas como Astronomy and Astrophysics. Se pueden personalizar las plantillas.*
4. *Numeración automática de secciones.*
5. *Creación automática de índice de contenidos (con funcionalidad hipertexto), alfabético, de tablas y de figuras.*
6. *Listas anidadas.*

<sup>8</sup>Que está en castellano y se instala junto con el programa

<sup>9</sup>En realidad, a la vez que se va conociendo LyX se tiene que ir conociendo  $\text{\LaTeX}$

<sup>10</sup>Aclarar que esto no significa que sea sólo para crear este tipo de textos. De nuevo el mejor ejemplo: estos apuntes.

7. *Editor matemático interactivo WYSIWYG.*
8. *Soporte de figuras Postscript, con rotación, escalado y texto asociado.*
9. *Tablas interactivas WYSIWYG.*
10. *Notas a pie de página y notas al margen.*
11. *Etiquetas/Referencias y Bibliografía (incluyendo soporte BibTeX).*
12. *Acceso a toda la funcionalidad LaTeX introduciendo directamente el código.*
13. *Capacidad de importar LaTeX y de exportar LaTeX, Postscript, DVI y ASCII.*
14. *Envío de faxes.*
15. *Soporte para las SGML Tools (LinuxDoc).*
16. *Menús, mensajes de error y shortcuts disponibles en multitud de lenguajes (incluido, por supuesto, el castellano).*
17. *Extensa documentación, incluyendo un tutorial para principiantes. Algunos documentos han sido ya traducidos a otros idiomas.*
18. *Rapidez y poco consumo de recursos.*
19. *Creación de transparencias para presentaciones.*
20. *Y por último la más sorprendente. LyX es totalmente gratuito, es libre. LyX está siendo desarrollado por programadores voluntarios que ponen el código fuente a disposición de todos los demás. ¿Qué más se le puede pedir?"*

Yo añadiría alguna más:

21. Posibilidad de exportar a HTML, PDF.
22. Me encanta

### 23.3.1. Instalación de LyX

Veamos cómo se instala, lo primero es bajarse la última versión (1.3.2) en formato rpm para Red Hat 8.0. No la hemos encontrado en Internet, así que la hemos compilado<sup>11</sup>, y la podéis bajar de la Web del curso<sup>12</sup>.

La página principal de LyX es:

<http://www.lyx.org>

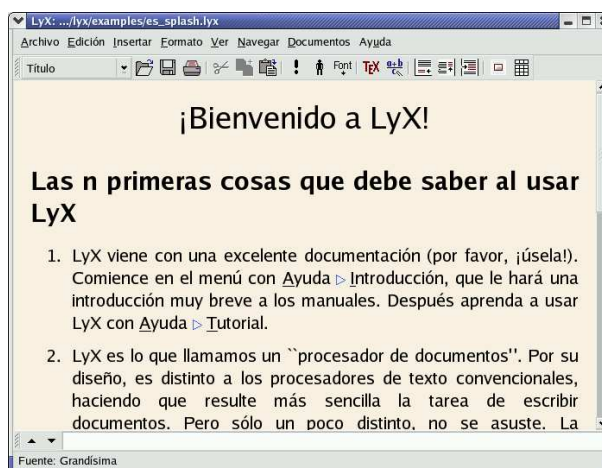
Vamos a explicar cómo instalar y trabajar con la última versión disponible en este momento. Supongamos que ya nos hemos bajado el paquete y pasemos a instalarlo

```
# rpm -ivh lyx-1.3.2-1_qt.i386.rpm
Preparing... ##### [100%]
1:lyx ##### [100%]
Configuring LyX for your system...
```

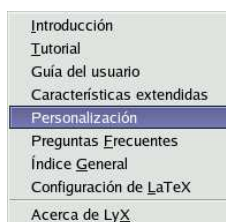
Si ahora en modo gráfico ejecutamos LyX y creamos nuestra configuración particular en el directorio de usuario, entramos en LyX:

<sup>11</sup>También hemos puesto el fichero de fuentes `lyx-1.3.2-1_qt.src.rpm`, que no es necesario bajar.

<sup>12</sup>Para Red Hat 9.0, se puede conseguir en la página principal de la aplicación o en <ftp://ftp.sylvan.com/pub/lyx/>



Esta es la pantalla de bienvenida al programa. Como se puede observar, si las variables de entorno relativas al lenguaje están bien, tendremos los menús en castellano. Trabajar con el programa no supone ninguna dificultad, máxime si tenemos en cuenta la magnífica documentación que lo acompaña. Para acceder a ella tan sólo tenemos que entrar en el menú **Ayuda**:



De estos manuales tenemos dos traducidos al castellano:

**Introducción** Breve introducción sobre qué es LyX, qué manuales acompañan al programa así como las teclas y sus combinaciones.

**Tutorial** *Tutorial de LyX* por AMIR KARGER y el EQUIPO DE LyX traducción al castellano: SERGIO GARCÍA REUS

El primero trae un listado de los manuales que acompañan al programa y un breve resumen de su contenido (ya se ha comentado antes). Sólo los primeros están en castellano, el resto están aún en inglés. En la **Introducción** nos aparece:

#### **“Introducción**

##### **Tutorial**

Si eres nuevo usando LyX, y nunca antes has usado ni oído hablar de L<sup>A</sup>T<sub>E</sub>X, entonces debes empezar aquí. Si ya has usado L<sup>A</sup>T<sub>E</sub>X, deberías leer aún así la sección “LyX para usuarios de L<sup>A</sup>T<sub>E</sub>X” (y hojear el resto del documento no te haría daño).

##### **Guía del Usuario**

La documentación principal. Intentaremos cubrir aquí la mayor parte de las opciones y características básicas de LyX. El manual principal asume que ya tienes algunos conocimientos de L<sup>A</sup>T<sub>E</sub>X, o que has leído el *Tutorial*.

##### **Características Extendidas**

Ampliación de la Guía del Usuario. Documenta los formatos adicionales y características de edición de propósito específico, incluyendo algunos trucos de expertos en L<sup>A</sup>T<sub>E</sub>X.

### Personalización

Una descripción de características avanzadas de LyX, entre las que se incluyen cómo personalizar el comportamiento global del programa: cosas tales como asociaciones de teclas, internacionalización y ficheros de configuración.

### Manual de Referencia

Contiene una descripción detallada y exhaustiva de las características de LyX, pero sin los útiles trucos y descripciones de uso de la *Guía del Usuario* u otros manuales.

### Errores conocidos

Este fichero contiene una lista de errores detectados en LyX, así como información sobre lo que deberías hacer si encuentras un error en el programa.

### Configuración de $\text{\LaTeX}$

LyX explora tu sistema durante la instalación. Este fichero contiene la información que LyX ha aprendido de tu instalación. Examínalo si echas en falta algo que deberías tener.”

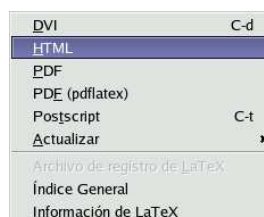
Además, se han añadido:

**FAQ** Una FAQ sobre LyX


**Tabla de contenidos** Tablas de contenidos de los manuales que acompañan al programa.

Para comenzar a trabajar con el programa deberíamos imprimir el TUTORIAL, para ello pulsaremos en la ventana de **Ayuda** y después en **Tutorial**. Una vez que tengamos el TUTORIAL en pantalla, podemos:

- Visualizar cómo quedaría antes de imprimirlo: para esto pulsaremos sobre el menú **Ver** y después en **PostScript**<sup>13</sup>, **PDF**, **DVI**, etc.



LyX generará automáticamente el fichero PostScript/DVI y abrirá el visor correspondiente GostView/Xdvi. Desde el visor GostView podemos después imprimir el Tutorial.

- Imprimir directamente el Tutorial, pulsando sobre el icono **Imprimir** , se abrirá el menú:



<sup>13</sup>Si no hemos instalado el visor gv no dispondremos de esta opción.

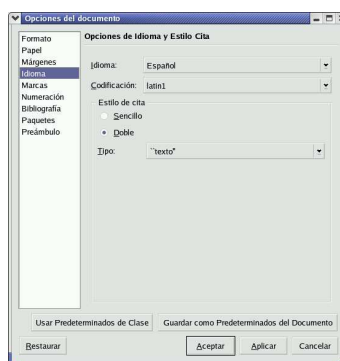
si aceptamos, el documento será enviado directamente a la impresora.

Con respecto a todas las posibilidades de LyX, os remitimos a la documentación del programa. Pero vamos a comentar varios aspectos no documentados que nos pueden facilitar trabajar con él.

### 23.3.2. Algunas cuestiones sobre LyX

#### Formato inicial del documento

Si queremos ver los caracteres correctamente (Latin1) y que LyX sepa que estamos escribiendo en castellano tendremos que entrar en **Documento** del menú **Formato**. En **Idioma** seleccionaremos **Español** y en **Codificación** **Latin1**.



En papel deberíamos optar por tamaño **A4** y si queremos que los márgenes se ajusten a un valor “razonable” optaremos por **Márgenes muy Pequeños**. Una vez realizados estos cambios deberíamos hacerlos efectivos para todos los documentos nuevos mediante la opción **Guardar como Predeterminados del Documento**, si activamos esta opción cada archivo nuevo que comencemos con LyX tendrá de codificación Latin1 y LyX sabrá que es en castellano.

#### Corrección ortográfica

Para que podamos usar la opción de corrección automática tenemos que tener instalado previamente el diccionario en castellano<sup>14</sup>, dicho paquete se instala por defecto y es:

```
aspell-es-0.2-11.i386.rpm
```

Escribamos un texto y verifiquemos la ortografía, para ello pulsemos sobre el menú **Edición**, en la opción **Corrección ortográfica**, se abrirá la ventana:


<sup>14</sup>Este paquete no sólo lo usa LyX, también es utilizado por otros editores instalados en nuestro ordenador.





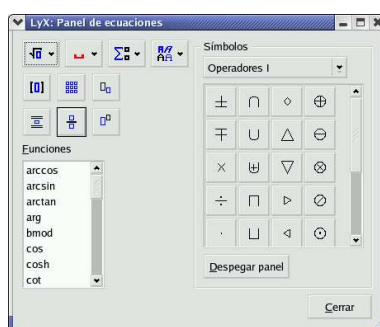
Ya sí, cuando aceptemos en esta ventana podremos comenzar la verificación ortográfica de nuestro documento.

### Panel de Fórmulas

Para introducir una fórmula matemática en LyX podemos usar el icono  o con el menú **Insertar**→**En Línea**, o bien pulsando **[Ctrl]+[m]**<sup>15</sup> se nos abrirá un rectángulo de borde rojo que nos indica que estamos en modo matemático<sup>16</sup>. En el recuadro que aparece podemos introducir expresiones de L<sup>A</sup>T<sub>E</sub>X directamente, por ejemplo si escribimos `\frac` y después pulsamos sobre la barra espaciadora se interpretará como una fracción.



Si no conocemos los comandos de L<sup>A</sup>T<sub>E</sub>X, podemos usar el Panel de Ecuaciones al que se accede desde **Insertar**→**Ecuación**→**Panel de Ecuaciones**



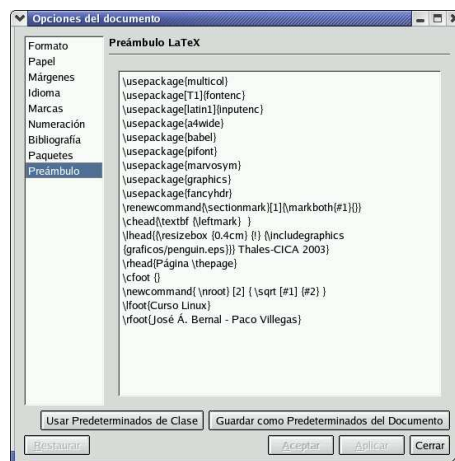
<sup>15</sup>Si pulsamos **[Ctrl]+[Shift]+[m]** o con la opción del menú **Presentación** la fórmula ocupa la línea completa. No es lo mismo una fórmula en la línea del texto que en línea completa, por ejemplo observar la diferencia entre  $\lim_{x \rightarrow \infty} f(x)$  frente a

$$\lim_{x \rightarrow \infty} f(x)$$

<sup>16</sup>Para los que sepáis L<sup>A</sup>T<sub>E</sub>X y sus comandos: para introducir un comando de L<sup>A</sup>T<sub>E</sub>X hay que anteponer la barra “\”, así por ejemplo para conseguir el símbolo de fracción podemos, o bien usar los menús de LyX, o bien el panel de fórmulas, o bien escribiríamos en modo matemático: `\frac`

## Preámbulo Latex

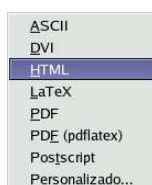
Si queremos definir nuevos comandos, cargar paquetes de  $\text{\LaTeX}$  o afinar sobre la estructura del documento tendremos que modificar el Preambulo de  $\text{\LaTeX}$ . Para conseguirlo pulsaremos en el menú **Formato**→**Documento** opción **Preámbulo Latex**, se nos abre una ventana desde la que podemos añadir opciones de  $\text{\LaTeX}$  a este documento



por ejemplo, cargar paquetes para poder trabajar con columnas (`\usepackage{multicol}`). La captura gráfica está realizada sobre el preámbulo usado para crear estos apuntes.

### 23.3.3. Exportar a HTML

Desde LyX podemos exportar directamente un fichero a HTML para publicarlo en internet, esto lo podemos hacer si tenemos instalado bien `tth` o bien `LaTeX2HTML`. Esta última utilidad se instala al instalar  $\text{\LaTeX}$ <sup>17</sup>. Si deseamos guardar un archivo de LyX en formato HTML tan sólo tenemos que pulsar en el menú **Archivo**, submenú **Exportar**, opción **Html**



#### Problemas y cómo resolverlos:

- Un inconveniente de exportar a HTML usando este método es que LyX se configura mediante el fichero `~/ .lyx/lyxrc.defaults`<sup>18</sup>. Para modificar los parámetros que se le pasan al programa tenemos que modificar una línea del tipo<sup>19</sup>:

```
\converter latex html "latex2html -no_subdir -split 0 -show_section_numbers
\i" "originaldir,needaux"
```

<sup>17</sup>Forma parte del paquete `tetex-latex`.

<sup>18</sup>Entre otros.

<sup>19</sup>Así aparece por defecto

con ella, además de parámetros de LyX (*needaux* obliga a que ejecutemos  $\text{\LaTeX}$  y generemos el fichero *.aux*), estamos usando  $\text{\LaTeX2HTML}$  de la forma:

```
latex2html -no_subdir -split 0 -show_section_numbers $$i
```

con este comando obligamos al programa a:

**-no\_subdir** al final del proceso nos copiará el fichero en HTML a nuestro directorio de trabajo (sin copiar los gráficos)

**-split 0** se generará un solo fichero con todo el contenido.

**-show\_section\_numbers** numera las secciones del documento

**\$\$i** fichero a pasar como argumento.

Si usamos el programa desde LyX, al ejecutar el comando anterior, no tendremos el fichero html y los gráficos fácilmente localizables y tendremos que buscar en */tmp* el subdirectorio temporal generado por LyX para almacenar el fichero que estamos usando y dentro de él, ya sí, tendremos el HTML. Por tanto, lo mejor es:

- Si queremos tener más control sobre el proceso de exportación podemos exportar primero a  $\text{\LaTeX}$  y después usar directamente el programa para exportar ficheros  $\text{\LaTeX}$  a html, la sintaxis a seguir es:

```
$ latex2html fichero_a_convertir.tex
```

Al usarlo con esta sintaxis se creará un directorio de nombre *fichero\_a\_convertir* en donde se almacenarán los gráficos y el html.

Si deseamos configurar el programa para un usuario debemos copiar el fichero

```
/usr/share/latex2html/dot.latex2html-init
```

a nuestro directorio de usuario con nombre *.latex2html-init*

```
$ cp /usr/share/latex2html/dot.latex2html-init ~/.latex2html-init
```

y modificarlo hasta que el programa quede a nuestro gusto.

Por ejemplo, para que los gráficos que permiten desplazarnos por las páginas se almacenen junto con el código html, debemos añadir la línea:

```
$LOCAL_ICONS= 1;
```

si lo que deseamos es que los gráficos no se exporten a png y sí a gif<sup>20</sup> añadiremos:

```
$IMAGES_TYPE= 'gif' ;
```

Para que las notas a pie de página aparezcan en el mismo fichero y no en uno aparte usaremos:

```
$NO_FOOTNODE= 1;
```

Para ampliar sobre el uso de este programa os remitimos a las *manpages* o bien a la completa documentación que lo acompaña y que se instala en:

```
/usr/share/latex2html/docs/
```

merece especial atención el fichero *manual.tex*.

## ➡ Para practicar

1. Obtener los ficheros *manual.dvi*<sup>21</sup>, *manual.ps*<sup>22</sup>, *manual.pdf*<sup>23</sup> a partir del fichero */usr/share/latex2html/docs/manual.tex*.
2. Optar por

<sup>20</sup>Es un formato propietario.

<sup>21</sup>Ejecutar dos veces

```
# latex /usr/share/latex2html/docs/manual.tex
```

```
22# dvi2ps /usr/share/latex2html/docs/manual.dvi -o /usr/share/latex2html/docs/manual.ps
```

```
23#dvipdf /usr/share/latex2html/docs/manual.dvi
```

- a) Imprimirlo con:  
`$ lpr manual.dvi`
- b) Visualizarlo usando el visor `xdvi`. Desde una `xterm` escribir:  
`$ xdvi manual.dvi`

## 23.4. Otros

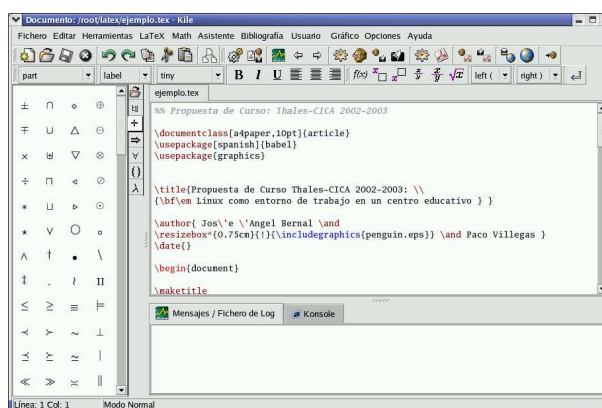
### 23.4.1. Kile

Si optamos por crear documentos con  $\text{\LaTeX}$ , kile es el mejor editor que conozco para Linux (salvando Emacs). Se trata de un programa que facilita la composición de documentos en  $\text{\LaTeX}$ . Tiene incorporados los comandos más usuales de  $\text{\LaTeX}$ , lo que supone mucha más rapidez a la hora de escribir un documento. Es necesario conocer algo de  $\text{\LaTeX}$  para poder usarlo.

Las últimas versiones disponibles son para el KDE 3.1 (<https://mall.kde.com/na/2/info/id/1011?sid=&sid=>), así que no podemos trabajar con ellas (salvo que actualicemos el sistema). La versión disponible para Red Hat 8 (con KDE 3.0) es la 1.32 y lo podéis bajar en formato rpm de<sup>24</sup>:

[http://ftp.kde.com/Office/Text\\_Word\\_Processing/Kile/](http://ftp.kde.com/Office/Text_Word_Processing/Kile/)

Una vez instalado veremos una pantalla como la que sigue:



Nos permite compilar los documentos y verlos en distintos formatos (DVI, Postscript, ps), insertar símbolos fácilmente, dispone de magnífica ayuda en línea, etc.

### 23.4.2. TexMacs

Somos conscientes de que se queda en el tintero un clásico, que emulando a un conocido club es “más que un tratamiento de textos”, se trata de emacs. Ya que lo dejamos “olvidado”, al menos comentar el editor  $\text{\TeX}$ macs. La página principal es:

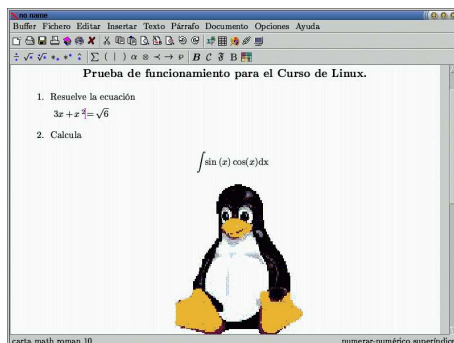
<http://www.texmacs.org/>

y la última versión es la `texmacs-1.0.1-13`. Si lo instalamos<sup>25</sup> y ejecutamos

`$ texmacs`

<sup>24</sup>Y de la Web del curso

<sup>25</sup>Si no hemos instalado  $\text{\LaTeX}$  no podremos instalarlo.



dispondremos de un editor de textos científico WYSIWYG inspirado en  $\text{T}_\text{E}\text{X}$  y GNUEmacs (necesita un equipo con cierta potencia para poder trabajar con él de forma desenvuelta), con el entorno en castellano.

### 23.4.3. Ampliación: Epix

Podemos bajarlo de <http://mathcs.holycross.edu/~ahwang/current/ePiX.html#Downloading>. Se trata de un pre-procesador de  $\text{L}_\text{A}\text{T}_\text{E}\text{X}$  que permite obtener gráficos de alta calidad usando una sintaxis “fácil de aprender”. Si bien su uso no es inmediato, los resultados son excelentes.

En este caso hemos optado por bajar las fuentes de la última versión estable, se trata de `epix-0.8.9_complete.tar.gz`. Para instalarlo, copiemos el fichero a `/opt` y

```
#cd /opt; tar -xzf epix-0.8.9_complete.tar.gz
# cd epix-0.8.9_complete
#make contrib test
#make install
```

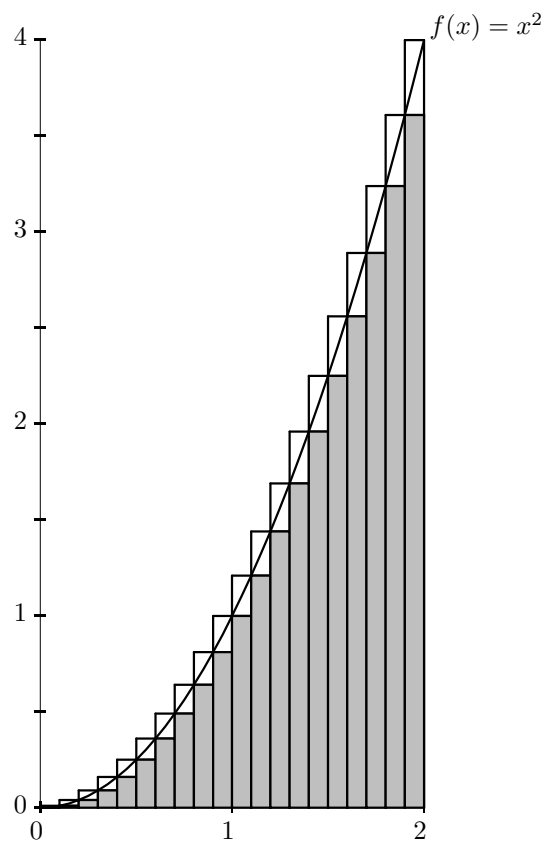
para ver algos ejemplos de qué podemos hacer, ejecutar:

```
#gv sample.ps
```

Podemos acceder al manual de uso del programa en `/opt/epix-0.8.9_complete/epix-0.8.9_tex`.

A ella os remitimos para conocerlo.

Como muestra de qué se puede hacer con él, el gráfico:



que se ha obtenido a partir del fichero fuente:

```
#include "epix.h"
using namespace std;
using namespace ePiX;
#define N 20.0 // Number of rectangles
double f(double t){
    return t*t;
}
main() {
    int i;
    double dx;
    picture(P(250, 500));
    bounding_box(P(0,0), P(2,4));
    unitlength("0.01in");
    begin();
    dx = x_size/N;
    bold;
    h_axis(P(x_min, y_min), P(x_max, y_min), x_size);
    v_axis(P(x_min, y_min), P(x_min, y_max), 2*y_size);
    h_axis_labels(P(x_min, 0), P(x_max, 0), x_size, P(-4, -12));
    v_axis_labels(P(0, y_min), P(0, y_max), y_size, P(-10, -2));
    label(P(x_max, f(x_max)), P(2,2), "$f(x)=x^{2}$");
    pen(2);
    plot(f, x_min, x_max, 40);
    plain;
    gray(0.25);
    for(i=0; i <N; ++i)
    {
        double ai=x_min + i*dx;
        double bi=x_min + (i+1)*dx;
```



```
bold;
line(P(bi, 0), P(bi, f(bi)));
line(P(ai, f(ai)), P(bi, f(ai)));
swatch(P(bi,0),P(ai,f(ai)));
blue;
line(P(ai, f(ai)), P(ai, f(bi)));
line(P(ai, f(bi)), P(bi, f(bi)));
}
end();
}
```

Una vez creado ese fichero tenemos que ejecutar:

```
$ epic fichero.xp
```

para obtener un fichero.eepic

```
$ elaps fichero.xp
```

para obtener fichero.eps. Como ya hemos comentado su uso no es inmediato pero pensamos que conocer su existencia puede ser de utilidad para algunos de vosotros.

➡ **Para practicar**

Escribir con LyX un documento como el que sigue y, tras exportarlo a L<sup>A</sup>T<sub>E</sub>X, “terminar” con él en formato html. Tux también hay que ponerlo.

EXAMEN

Nombre: .....

1. Calcula:  $\left[ \left( -\frac{2}{3} \right)^{-2} - 1 \right]^{-2} \cdot \left[ 1 - \left( -\frac{2}{3} \right)^2 \right]^2$

2. Resolver la ecuación:  $\sqrt{x+1} = \sqrt[3]{x^2-1}$

3. Resuelve gráficamente el siguiente sistema de inecuaciones:

$$\begin{cases} 3x + 2y - 7 < 0 \\ 2x - 4y + 3 > 0 \end{cases}$$

4. En la tabla que sigue se recoge el número de hermanos de una clase de 3º de ESO:

Nº Hermanos	0	1	2	3	4	5
Frecuencia	5	7	13	2	2	1

Halla  $\bar{x}$  y  $\sigma_x$ .





## Capítulo 24

# Matemáticas

Tan pronto como exista una Máquina Analítica, no cabe duda de que fijará los futuros derroteros de la ciencia. Y siempre que se busque un resultado por este medio, surgirá la pregunta: ¿cuál es el curso de computación mediante el cual puede la máquina obtener estos resultados en el menor tiempo posible? (CHARLES BABBAGE)

Linux y recursos de matemáticas: una conjunción casi perfecta. Linux es la herramienta idónea para construir las matemáticas. Vamos a hablar de unos cuantos programas de matemáticas exponiendo una breve reseña de lo que se puede hacer con cada uno. Todos ellos están muy bien documentados (casi siempre en inglés) pero no creemos que, para usar la mayoría de ellos, eso sea un problema.

De nuevo, nos saltamos algunos programas en modo texto como `bc`<sup>1</sup> o las calculadoras que incluyen tanto el KDE (`kcalc`) como Gnome (`gnome-calculator`). Merece la pena comentar aunque sea de pasada el generador de fractales del KDE, el `kfract`.

Para un listado más extenso que el que aquí se expone os emplazamos de nuevo a las direcciones ya comentadas en esta entrega:

<http://ceu.fi.udc.es/SAL/index.shtml>

<http://glub.ehu.es/soft/>

Además de los programas que vamos a comentar, existe uno en modo texto, es `Octave`<sup>2</sup>. `Octave` es un lenguaje de alto nivel para cálculo numérico. Como ya se ha dicho se trabaja con él en modo texto y permite representar gráficas usando `gnuplot`. Es compatible con `MatLab`. Es gratuito y la versión 2.1.31 está en el CD1 de la distribución.

Vamos a comentar:

**geg** representaciones gráficas

**drgenius** geometría plana dinámica

**scilab** programa de computación científica

**gnuplot** representaciones gráficas

**xgfe** fron-end para `gnuplot`

**grace** gráficos estadísticos y análisis de datos.

**R** para trabajar todas las posibilidades de la estadística

---

<sup>1</sup>`bc` es una calculadora en modo texto con precisión numérica arbitraria. Su sintaxis es similar a la del C

<sup>2</sup>

■ La página principal de `Octave` es:

<http://www.octave.org>

■ Otro de álgebra simbólica muy interesante es `Máxima`: <http://www.ma.utexas.edu/maxima.html>

**MuPAD** álgebra simbólica

**geomview** geometría a lo “grande”

## 24.1. Gráficos

### 24.1.1. geg

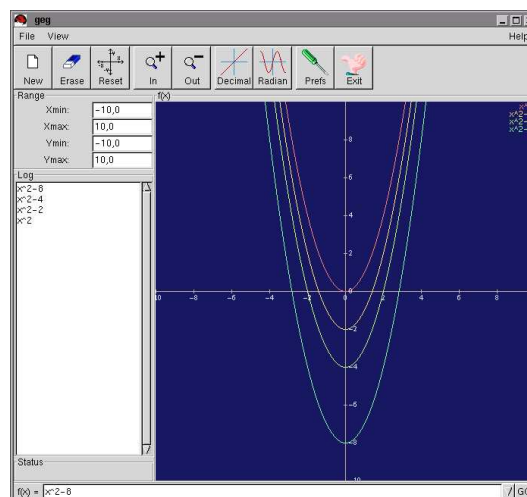
La página principal de este programa es:

<http://www.infolaunch.com/~daveb/>

en ella podéis encontrar la última versión en formato rpm, en el servidor de Thales la hemos puesto y ocupa muy poco, es el paquete `geg-1.0.2-1.i386.rpm`.

Este programa sirve para representar funciones matemáticas en dos dimensiones. Su uso no presenta ninguna dificultad y podría ser de utilidad para trabajar con él en el aula. Necesita de las librerías `gtk` para funcionar.

```
# rpm -ivh geg-1.0.2-1.i386.rpm
geg #####
Si lo ejecutamos:
$ geg
```



En la captura anterior se puede observar que se pueden representar varias funciones a la vez, cambiar la escala, etc. Los principales inconvenientes que presenta es que no permite imprimir ni guardar los trabajos. Para imprimir/guardar una representación gráfica hay que capturar la pantalla.

### 24.1.2. Gnuplot

GnuPlot<sup>3</sup> es un programa que trabaja en el entorno Xwindow y que permite representar gráficas de funciones y visualizar datos matemáticos. Acompaña a la distribución, o sea que no tenemos que bajarnos ningún paquete (¡menos mal!). Para instalarlo montaremos el CD2 de la RedHat y ejecutaremos:

```
# rpm -ivh gnuplot-3.7.2-1.i386.rpm
```

Para ejecutarlo sólo tenemos que escribir desde una xterm el comando:

```
$ gnuplot
```

Al ejecutarlo veremos el prompt de entrada de órdenes

<sup>3</sup>Hay versiones para bastantes plataformas

gnuplot >  
algunas ordenes básicas són:

**help** ayuda del programa

**helpplot** ayuda sobre la representación de funciones

**help comando** ayuda sobre comando

**pwd** muestra el directorio de trabajo

**cd "directorio"** cambiar el directorio de trabajo. **directorio** debe ir entre comillas, éstas pueden ser simples o dobles.

**clear** borra la pantalla

**plot [xmin:xmax][ymin:ymax] función** representa la función en 2D

**splot [xmin:xmax][ymin:ymax][zmin:zmax] funcion** representa la función en 3D

**quit, exit** salir de gnuplot

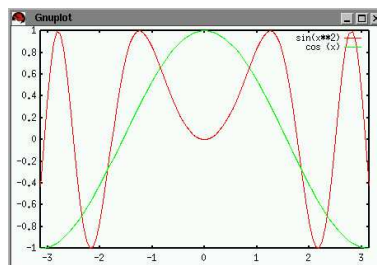
**load 'archivo'** cargar un fichero en el que se han incluido una serie de datos para representarlos

**save 'archivo'** guarda una gráfica en un fichero de nombre **archivo**

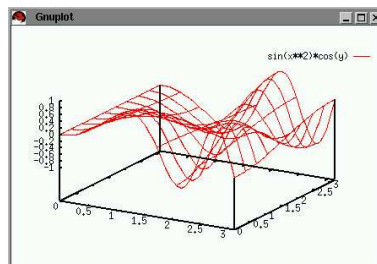
**show comando** muestra los valores de ese comando

Hemos puesto un par de mini-manuales (en inglés y en castellano) sobre su uso en la sección manuales. En el manual en castellano, se explica cómo salvar una gráfica en PostScript o cómo pasar una gráfica a  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .  
Veamos un par de ejemplos<sup>4</sup>

gnuplot>plot [-pi:pi] sin(x\*\*2), cos(x)



gnuplot>splot [0:pi][0:pi] sin(x\*\*2)\*cos(y)

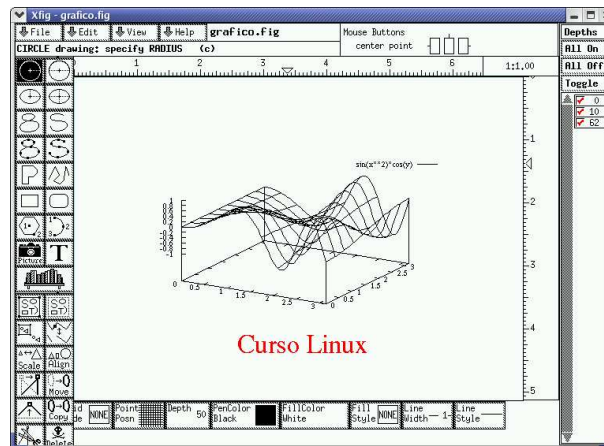


<sup>4</sup>Observar que estamos pintando dos funciones a la vez, esto se consigue separándolas con una coma

Si lo que queremos es salvar la gráfica anterior en otro formato (por ejemplo en formato fig para poder editarla con el xfig) escribiremos:

```
gnuplot>set terminal fig
gnuplot>set output "grafico.fig"
gnuplot>plot [0:pi][0:pi] sin(x**2)*cos(y)
para restaurar la salida al terminal gráfico:
gnuplot>set terminal x11
```

Con el programa xfig podremos editar el gráfico que hemos creado (en formato vectorial).

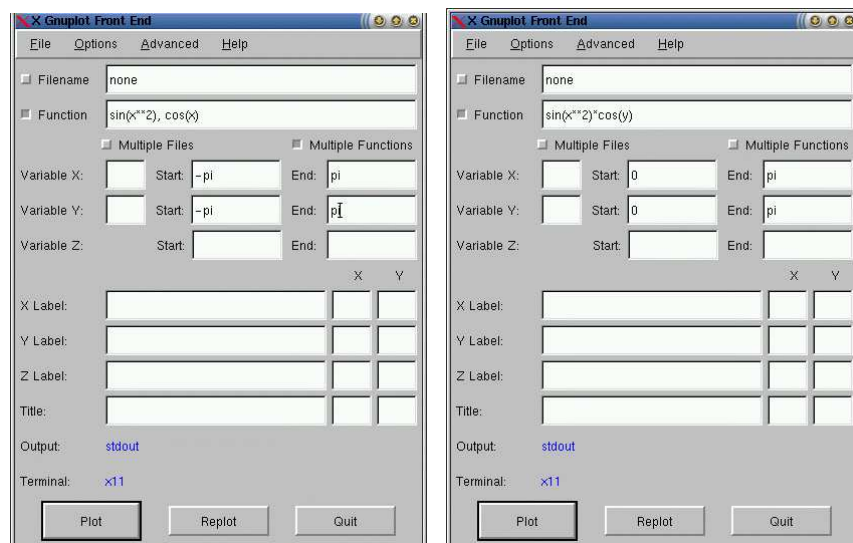


## Xgfe

Si la sintaxis del programa nos resulta “abrumadora” disponemos de un front-end para él en:

<http://www.bl.physik.uni-muenchen.de/rechner/grafik/xgfe/xgfe.html>

Se trata del programa Xgfe (xgfe-2.0-1.i386.rpm) que lo podemos bajar de la WEB del curso (xgfe-2.0-rh51-1.i386.rpm). Para poder instalarlo, antes hemos de instalar unas librerías (venían con la Red Hat 7.3) que podemos conseguir en <http://rpmfind.net/linux/RPM/redhat/7.3/i386/qt1x-1.45-16.i386.html>. Tras bajar ambos paquetes e instalarlos, dispondremos un entorno gráfico que nos va a permitir trabajar con más facilidad con el programa gnuplot. Así los dos gráficos anteriores se obtendrían (notar que tenemos que tener marcada la casilla función):





Para que podamos dibujar en 3D antes debemos marcar la opción **3D** en el menú: **Options** → **Datafile** → **Type**.

#### ➤ Para practicar:

Comprobar la salida de ejecutar (desde gnuplot o usando xgfe):

```
gnuplot>plot x
gnuplot>plot x**2
gnuplot>plot abs(x)
gnuplot>plot [-1:1] sqrt(1-x**2)
gnuplot>plot x-floor(x)
gnuplot>set parametric
gnuplot>splot cos(x)*cos(y),cos(x)*sin(y),sin(x)
```

## 24.2. Estadística

### 24.2.1. R

Se trata de un programa que permite un tratamiento completo de datos a nivel estadístico, tanto a nivel de obtención de parámetros, inferencia o gráficos de todo tipo de alta calidad. Podemos obtener la última versión del programa en:

<http://www.r-project.org/>

usando cualquiera de los *mirrors* a los que podemos acceder pulsando sobre *downloads*.

En esa página, pulsando sobre **Documentation** → **Contributed** accedemos a la posibilidad de bajar diferente documentación sobre R, de entre ella destacar tres manuales en castellano, se trata de:

- *R para Principiantes*, traducido por JORGE A. AHUMADA
- *Una introducción a R*, ANDRÉS GONZÁLEZ y SILVIA GONZÁLEZ
- *Gráficos Estadísticos con R*, JUAN CARLOS CORREA y NELFI GONZÁLEZ

Si bajamos la última versión en formato rpm para nuestra versión de Red Hat (por ejemplo de <http://cran.at.r-project.org/bin/linux/redhat/8.x/i386/>) y la instalamos con:

```
# rpm -ivh R-1.7.0-1.i386.rpm
```

```
warning: R-1.7.0-1.i386.rpm: V3 DSA signature: NOKEY, key ID 97d3544e
```

```
error: Failed dependencies:
```

```
libblas.so.3 is needed by R-1.7.0-1
```

```
Suggested resolutions:
```

```
blas-3.0-18.i386.rpm
```

Instalemos el programa pedido (del CD1) y después procedamos con R.

Para ejecutarlo (mejor en modo gráfico), desde un terminal escribimos:

```
$ R
```

```
R : Copyright 2003, The R Development Core Team
```

```
Version 1.7.0 (2003-04-16)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.
```

```
You are welcome to redistribute it under certain conditions.
```

```
Type 'license()' or 'licence()' for distribution details.
```

```
R is a collaborative project with many contributors.  
Type 'contributors()' for more information.
```

```
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for a HTML browser interface to help.  
Type 'q()' to quit R.
```

```
[Previously saved workspace restored]
```

```
>
```

aparece el prompt de entrada de órdenes del programa en el que se nos indica que para salir hay que escribir `q()`. Si deseamos acceder a la completa ayuda que trae podemos ejecutar:

```
>help()
```

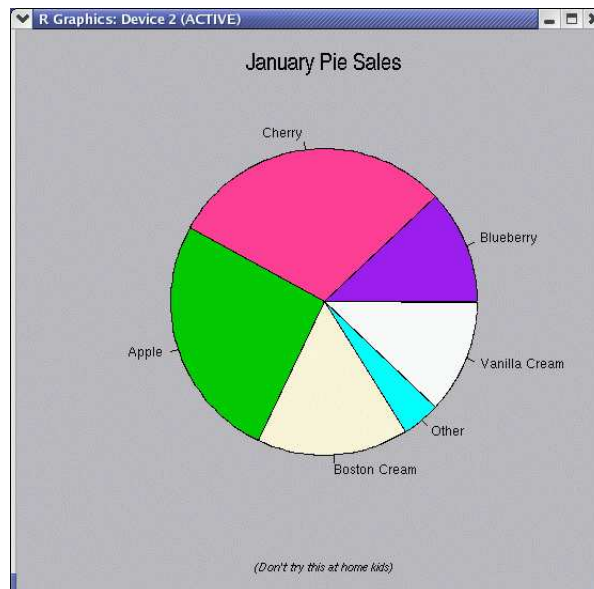
o en modo html

```
>help.start()
```

Para poder ver algunos ejemplos de los gráficos que podemos hacer con el programa, escribiremos:

```
>demo(graphics)
```

y pulsaremos sobre la tecla **Intro** varias veces hasta que finalice la presentación.



para salir:

```
>q()
```

### ➡ Para practicar:

En el fichero `Fechas.dat` hay una tabla de datos (se puede bajar del servidor) que contiene el sexo, fecha de nacimiento y defunción de una población de 375 individuos. Vamos a ver algunas cuestiones básicas que podemos hacer desde R.

1. Creamos un directorio en donde poner el trabajo que vamos a realizar. Por ejemplo:

```
$mkdir fechas
```

y ponemos en él el fichero `fechas.dat` bajado de internet.

2. Desde el directorio `fechas` comencemos una sesión del programa.

```
$R
>datos.fechas<-read.table("fechas.dat",header=TRUE)
>names(datos.fechas)
[1] "Sexo" "an" "fm"
```

Con la primera línea leemos el contenido del fichero y lo ponemos en la variable `datos.fechas`. Notar que hemos añadido la opción de que la primera línea del fichero es el nombre de los campos contenidos en él. Podemos ver cuáles son con el comando de la 2ª línea.

Para no tener que acceder a los nombres de los campos con la nomenclatura `variable.$nombrecampo`, ejecutamos:

```
>attach(datos.fechas)
```

a partir de ahora podemos referenciar un campo sólo por su nombre. Obtengamos algunos datos sobre la variable `an` (año de nacimiento)

```
>summary(an)
Min. 1st Qu. Median Mean 3rd Qu. Max.
1822 1905 1927 1929 1960 1997
```

Obtenemos el valor máximo, mínimo, la media y los cuartiles de esta variable.<sup>5</sup> Obtengamos la media, mediana, varianza, desviación típica y rangos de la variable `an`:

```
>mean(an)
[1] 1929.133
>median(an)
[1] 1927
>var(an)
[1] 1122.453
>sd(an)
[1] 33.50303
>range(an)
[1] 1822 1997
```

Para caracteres cualitativos:

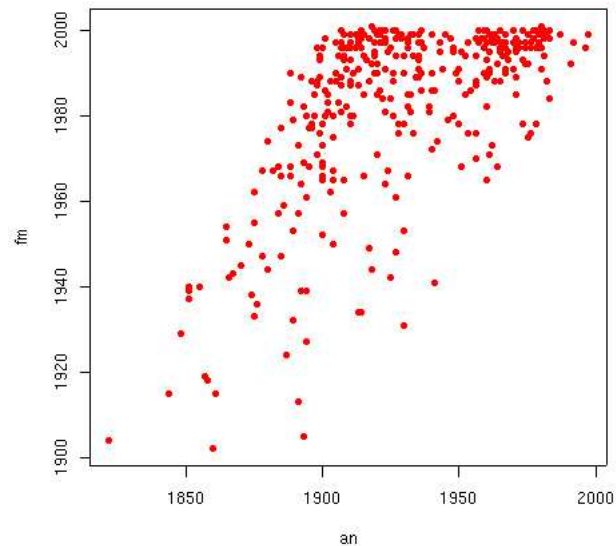
```
>summary(Sexo)
h m
228 147
```

3. Estadística bidimensional: obtengamos la covarianza, coeficiente de correlación y la nube de puntos (modificando el “punto” por defecto a relleno y color rojo) de las dos variables cuantitativas

---

<sup>5</sup>Si no hubiésemos ejecutado el comando `attach`, tendríamos que haber escrito  
`>summary(datos.fechas$an)`

```
>cov(an, fm)
[1] 419.8520
>cor(an, fm)
[1] 0.6177829
>plot(an, fm, pch=20, col="red")
```



Con:

```
>lm(an~fm)

Call:
lm(formula = an ~ fm)

Coefficients:
(Intercept)      fm
   -94.181      1.020
```

obtenemos la ordenada en el origen y la pendiente de la recta de regresión de *an* sobre *fm*, y con:

```
>lm(fm~an)

Call:
lm(formula = fm ~ an)

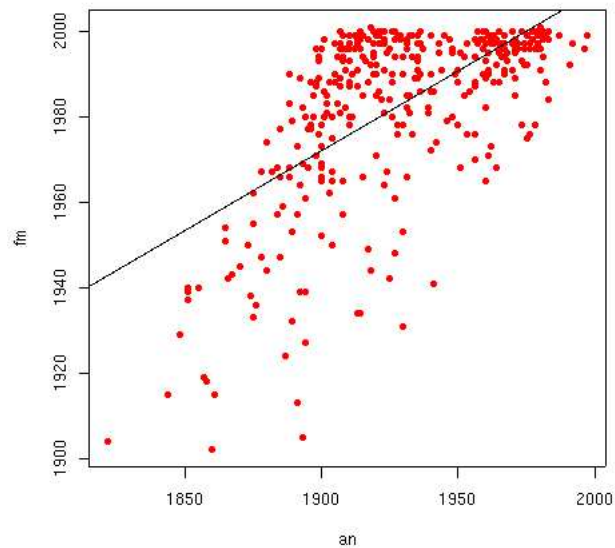
Coefficients:
(Intercept)      an
  1261.3968    0.3740
```

la ordenada en el origen y la pendiente de la recta de regresión de *fm* sobre *an*.

Representemos la recta de regresión de *fm* sobre *an* en la nube de puntos<sup>6</sup>.

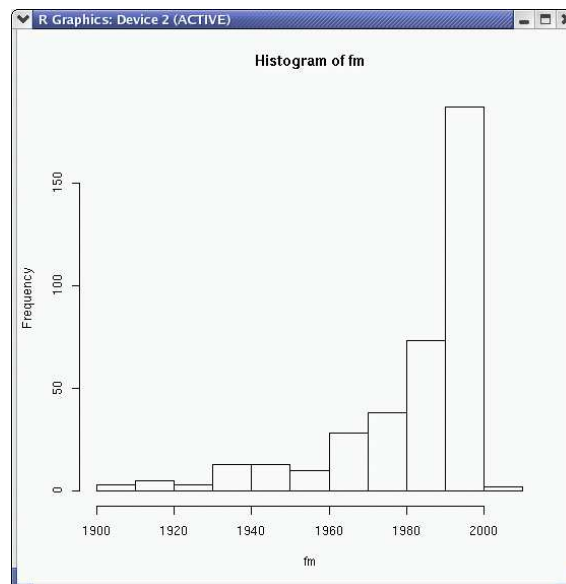


```
>abline(lm(fm~an))
```



4. También podemos hacer un histograma con:

```
>hist(fm)
```

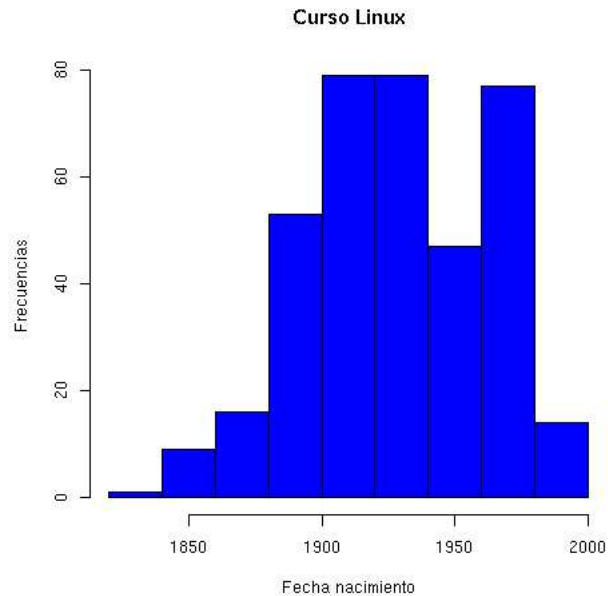


Pero nos interesa adecuarlo a nuestro gusto y para eso ejecutamos:

```
>hist(an, main="Curso Linux", xlab="Fecha nacimiento", ylab="Frecuencias", col="blue")
```

- Para poder representar la otra recta es necesario ajustar adecuadamente la pendiente y la ordenada en el origen de la recta, es decir, despejar y en función de x.
- Merece la pena ver el resultado de:  

```
>plot(lm(fm~an))
```



5. Si deseamos guardar el fichero en formato png escribiremos:

```
>png()
```

```
>hist(an, main="Curso Linux",xlab="Fecha nacimiento",ylab="Frecuencias",col="blue")
```

Con

```
>x11()
```

dejamos las cosas como estaban. Y en la zona de trabajo se almacenará el fichero.

6. salimos (y guardamos la sesión) con<sup>7</sup>:

```
>q()
```

Las posibilidades del programa son inmensas y la documentación disponible excelente.

### 24.2.2. Grace

Grace es un programa para representar gráficos en dos dimensiones. No sólo permite representar funciones, sino que además es muy bueno para gráficos estadísticos. Destaca porque permite realizar análisis estadísticos sobre los datos, integrales y derivadas, interpolaciones, etc.

La página principal del programa es <http://plasma-gate.weizmann.ac.il/Grace/>. No vamos a trabajar con la última versión ya que en la Web del programa no está en formato rpm. La versión que vamos a utilizar (no hay diferencias significativas) es la 5.1.10, que podemos bajar desde:

```
ftp://plasma-gate.weizmann.ac.il/pub/grace/contrib/RPMS/
```

Si comenzamos la instalación sin más

```
# rpm -ivh grace-5.1.10-1.i386.rpm
```

error: Failed dependencies:

```
libfftw.so.2 is needed by grace-5.1.10-1
```

```
libpdf.so.1 is needed by grace-5.1.10-1
```

<sup>7</sup>Antes de salir nos preguntará si queremos guardar la imagen del espacio de trabajo, lo mejor sería decir que sí. Podemos guardar el espacio de trabajo en cada directorio en donde ejecutamos R, así, según el trabajo que estemos realizando disponemos de la posibilidad de tener distintas sesiones de trabajo guardadas en función del directorio desde donde lo hayamos llamado.

nos aparecerán problemas de dependencias. Necesitamos los paquetes

fftw-2.1.3-1.i386.rpm

pdflib-4.03-1.i386.rpm

los dos están en la misma dirección que el programa y ¡qué casualidad!, también en MILETO (es que está todo previsto).

Si los bajamos e instalamos no deberíamos tener problemas, siempre que los instalemos en “orden”.

Previo a la instalación del programa debemos “decirle” al sistema que hemos instalado unas librerías nuevas, y el lugar en donde están instaladas. Para eso editaremos el fichero `/etc/ld.so.conf` y añadiremos la línea

```
/usr/local/lib8
```

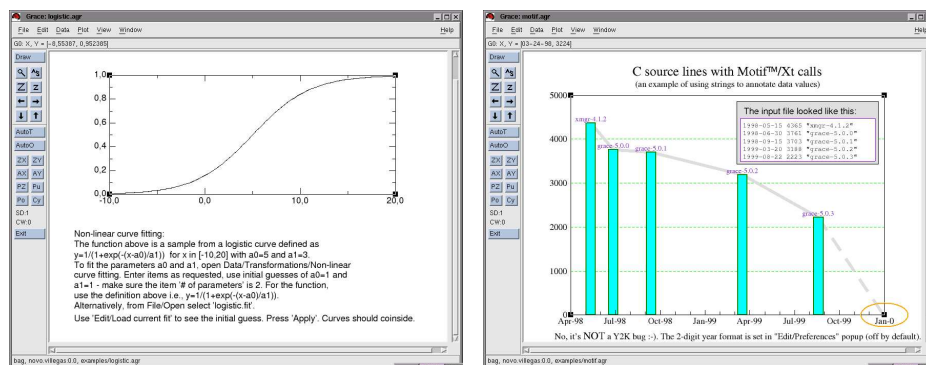
después, como root, hay que actualizar los *links* a las librerías que se encuentran en los directorios listados en el fichero editado. Para eso ejecutaremos:

```
# /sbin/ldconfig
```

Para ejecutar el programa escribiremos desde una xterm

```
$ xmgrace9
```

Si pulsamos sobre **Help** podremos comprobar que tenemos a nuestra disposición un tutorial, una guía de uso, una FAQ y además podemos acceder a ejemplos. Un par de capturas de ellos son:



Veamos un ejemplo sencillo de sus posibilidades:

➤ **Para practicar:** La tabla siguiente muestra las respectivas alturas  $X$  e  $Y$  de una muestra de 12 padres y sus hijos primogénitos (en pulgadas):

Altura $X$ del padre (en pulg)	65	63	67	64	68	62	70	66	68	67	69	71
Altura $Y$ del hijo (en pulg)	68	66	68	65	69	66	68	65	71	67	68	70

1. Construye el diagrama de dispersión.
2. Halla la recta de regresión de  $Y$  sobre  $X$ .

**Solución:**

Ejecutemos

```
# xmgrace
```

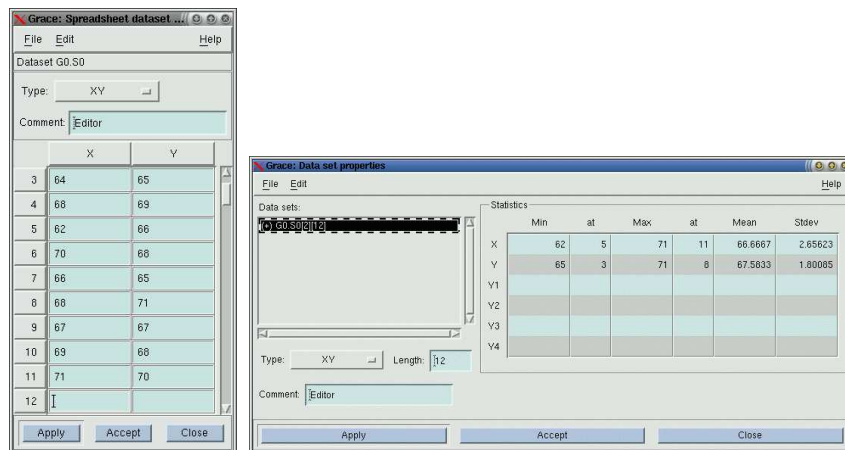
y en la ventana resultante, pulsemos sobre **Edit**→**Data Sets...**→**Edit**→**Create new**→**In spreadsheet**, después pasemos a introducir los datos:

<sup>8</sup>Aquí se instala el paquete `fftw`.


<sup>9</sup>Con

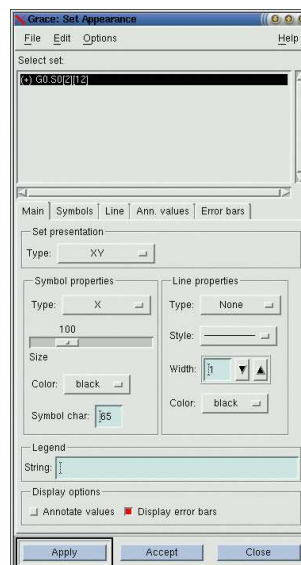
`$ grace`

se ejecuta en modo alfanumérico.

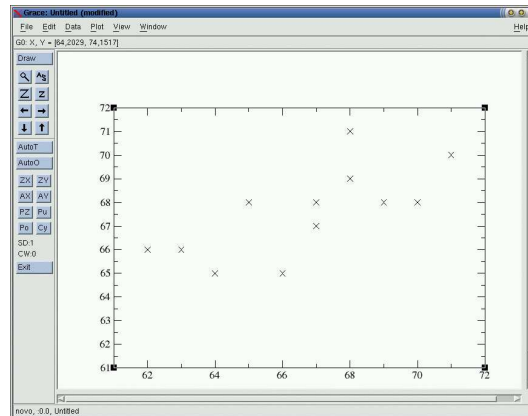


Tras aceptar, veremos en la ventana anterior que ya tenemos calculados algunos parámetros estadísticos

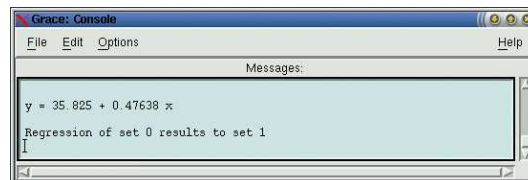
1. Si aceptamos en la última ventana y pulsamos sobre  de la pantalla principal de la aplicación, nos aparecerán una serie de líneas que hemos de “eliminar”:
  - a) Pulsemos sobre los ejes con el ratón y adecuemos la escala (desde 61 a 72).
  - b) Después pulsemos sobre las líneas con el ratón y, en la ventana que aparece, optemos por que no se unan los puntos con líneas y que los puntos se muestren como X.



Si aceptamos tendremos:



2. En el menú principal pulsemos sobre **Data**→**Transformations**→**Regression** y aceptemos en la ventana resultante, saldrá



Además, la recta de regresión se dibujará en la nube de puntos.

## 24.3. Álgebra simbólica: MuPAD

MuPAD<sup>10</sup> es un programa de computación algebraica de propósito general y con unas posibilidades amplísimas. La página principal del programa es:

[http://www.mupad.de/index\\_uni.shtml](http://www.mupad.de/index_uni.shtml)



A través de ella podéis registraros y vía correo electrónico recibiréis la clave para eliminar el límite de uso.

Desde esa misma página se puede acceder a algunos manuales en formato PostScript que permiten iniciarse en su uso. Por otro lado, la documentación que trae es considerable y se accede a ella desde el mismo programa. Para instalar la última versión hay que bajársela<sup>11</sup> de <ftp://ftp.mupad.de/MuPAD/distrib/unix/>:

```
MuPAD-2.5.2-2.i386.rpm
```

La instalación no presenta ningún problema. Si lo ejecutamos con

```
$ xmuPAD12
```

y, por ejemplo, escribimos

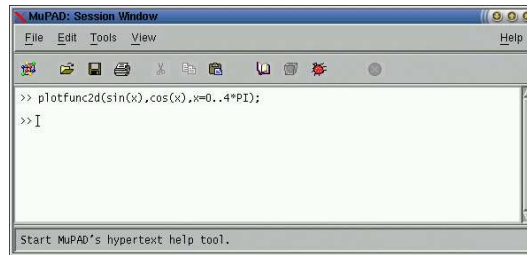
```
>>plotfunc2d(sin(x), cos(x), x=0..4*PI)
```

<sup>10</sup>Gratuito para uso académico/personal

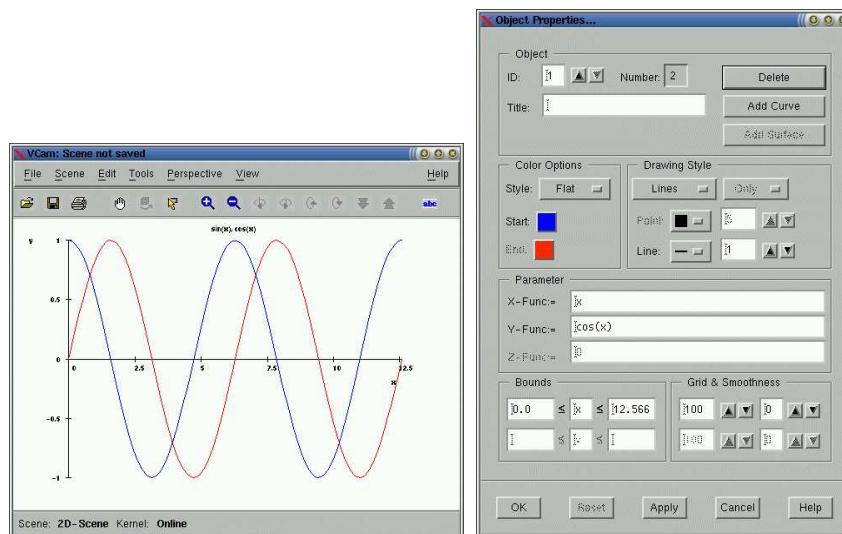
<sup>11</sup>Son 25 MB aproximadamente. Si deseamos la versión de MuPAD con Scilab hay que bajar e instalar: `MuPAD-scilab-2.5.2-2.i386.rpm`

<sup>12</sup>Si no queremos trabajar con él en modo gráfico hay que usar:

```
$ mupad
```



obtendremos



Con la segunda ventana podremos modificar el gráfico a nuestro gusto. Merece la pena.

#### ➡ Para practicar: Un paseo por el programa

Vemos algunas posibilidades más del programa (la explicación de cada línea es el propio resultado). Iniciemos el entorno (por si teníamos algo escrito):

```
>>reset()
```

- Para operaciones/aproximaciones con números:

```
>>(1+2/3)/(1-3/4);
```

$\frac{20}{3}$

```
>>23^32;
```

37608910510519071039902074217516707306379521

```
>>factor(%);
```

32

23

```
>>float(sin(3*PI));
```

0.0

```
>>DIGITS:=100: float(E);
```

```
2.718281828459045235360287471352662497757247093699959574966967627724076630\
353547594571382178525166428
```

```
>>isprime(3);
```

TRUE

```
>>isprime(4);
```

FALSE

```
>>simplify(E^(I*PI)+1);
```

0

#### ■ Para resolver ecuaciones y sistemas:

```
>>solve({x-y=3, x+y=6}, {x, y});
```

$\{[x = 9/2, y = 3/2]\}$

```
>>factor(x^2-x-2);
```

$(x + 1) (x - 2)$

#### ■ Como herramienta para simplificaciones

```
>>expand(sin(x+y));
```

$\cos(x) \sin(y) + \cos(y) \sin(x)$

#### ■ Para trabajar con funciones

```
>>f:=x/(x-1)^2;
```

$$\frac{x}{(x-1)^2}$$

```
>>f1:=diff(f,x);
```

$$\frac{1}{(x-1)^2} - \frac{2x}{(x-1)^3}$$

```
>>solve(f1,x);
```

$\{-1\}$

```
>>eval(subs(f,x=-1));
```

$-1/4$

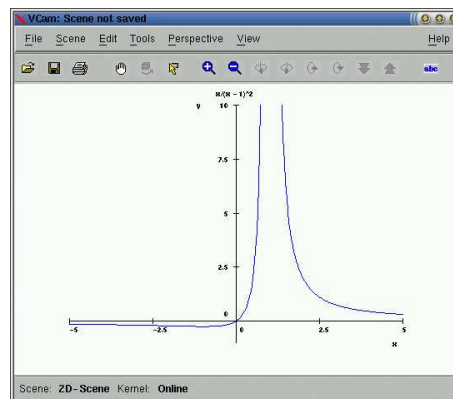
```
>>limit(f,x=infinity);
```

0

```
>>limit(f,x=1,Left);
```

infinity

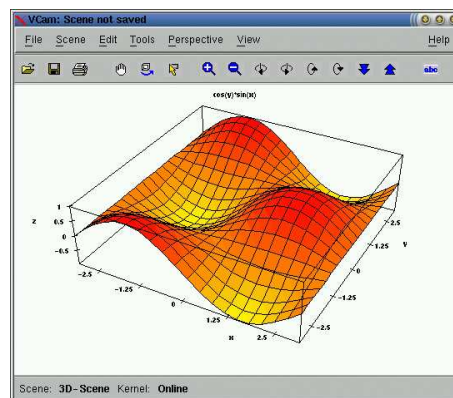
```
>>plotfunc2d(f,x=-5..5,y=-1..10);
```



```
>>series(f,x=infinity);
```

$$\frac{1}{x} + \frac{1}{2x^2} + \frac{1}{3x^3} + \frac{1}{4x^4} + \frac{1}{5x^5} + O\left(\frac{1}{6x^6}\right)$$

```
>>plotfunc3d(sin(x)*cos(y),x=-PI..PI,y=-PI..PI);
```



#### ■ Para álgebra lineal

```
>>M:=Dom::Matrix();
```

```
Dom::Matrix()
```

```
>>export(linalg);
```

```
>>A:=M([[1,2,3],[0,5,-1],[6,5,4]]);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 5 & -1 \\ 6 & 5 & 4 \end{bmatrix}$$

```
>>A*A;
```



```

+-          +-
| 19, 27, 13 |
|           |
| -6, 20, -9 |
|           |
| 30, 57, 29 |
+-          +-

>>info(linalg);
Library 'linalg': the linear algebra package

-- Exported:
addCol,      addRow,      adjoint,      angle,
basis,       charmat,     charpoly,     col,
companion,   concatMatrix, conjugate,     crossProduct,
curl,        delCol,      delRow,       det,
divergence,  eigenvalues,  eigenvectors, exp,
expr2Matrix, factorCholesky, factorLU,     factorQR,
frobeniusForm, gaussElim,  gaussJordan,  grad,
hermiteForm, hessenberg,  hessian,      hilbert,
intBasis,    inverseLU,    invhilbert,   isHermitean,
isPosDef,    isUnitary,     jacobian,     jordanForm,
matdim,      matlinsolve,    matlinsolveLU, minpoly,
multCol,     multRow,        ncols,        nonZeros,
norm,        normal,      normalize,    nrows,
nullspace,   ogCoordTab,    orthog,       permanent,
pseudoInverse, randomMatrix, rank,         row,
scalarProduct, setCol,      setRow,       smithForm,
stackMatrix, submatrix,     substitute,    sumBasis,
swapCol,     swapRow,        sylvestor,    tr,
transpose,   vandermondeSolve, vecdim,       vectorPotential,
wiedemann

>>transpose(A);

+-          +-
| 1, 0, 6 |
|         |
| 2, 5, 5 |
|         |
| 3, -1, 4 |
+-          +-

>>det(A);

-77

>>1/A;

+-          +-
| -25/77, -1/11, 17/77 |
|                       |
|                       |

```

```

| 6/77, 2/11, -1/77 |
|                    |
| 30/77, -1/11, -5/77 |
+-+                  +-+

```

Si pulsamos sobre **Help** accederemos a la completísima documentación sobre las posibilidades de este programa.

## 24.4. scilab

La página principal del programa es

<http://www-rocq.inria.fr/scilab>

desde ella accedemos a <http://www-rocq.inria.fr/scilab/get.html>, y desde aquí podremos bajarnos

`scilab-2.7-1.i586.rpm`

Scilab es un programa de computación científica y matemática. Su potencia de cálculo es magnífica.

Su sintaxis es similar a la de MatLab. Permite hacer animaciones, gráficos en dos y 3 dimensiones, etc.

Necesita de unas librerías

```
# rpm -ivh scilab-2.7-1.i586.rpm
```

error: Failed dependencies:

`libtk8.3.so` is needed by `scilab-2.7-1`

Suggested resolutions:

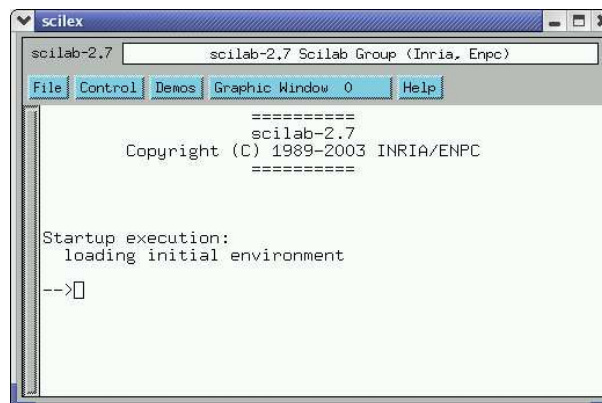
`tk-8.3.3-74.i386.rpm`

contenidas en el CD2.

Una vez instaladas las librerías que nos han generado el problema de dependencias (`tk-8.3.3-74.i386.rpm`) y el programa (lo instala por defecto en `/usr/lib/scilab-2.7`) podremos ejecutarlo con:

```
$ scilab
```

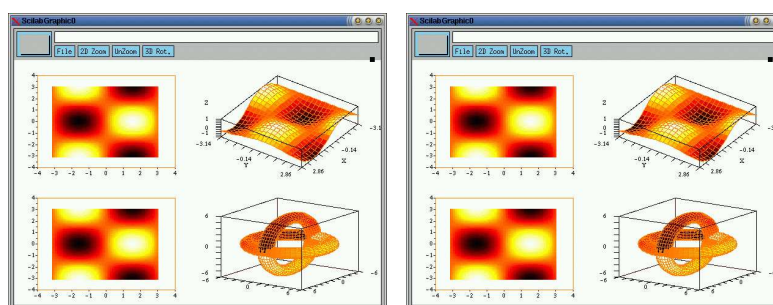
La pantalla principal del programa es:



Si pulsamos sobre el botón **Help** tendremos acceso a una ayuda bastante extensa. Es interesante pulsar sobre el botón **demos** para hacerse una idea de sus posibilidades



Las capturas que siguen se han realizado con las demos



Trae una ayuda muy buena. Para poder disponer de documentación sobre él os remitimos a la página principal del programa:

<http://www-rocq.inria.fr/scilab/doc.html>

## 24.5. Geometría

### 24.5.1. DrGenius

DrGenius es un programa de Geometría tipo Cabri, es decir, un programa de geometría dinámica plana que permite exportar a  $\text{\LaTeX}$ /PostScript. No está tan desarrollado como Cabri pero va evolucionando y mejorando bastante. La página principal del programa es<sup>13</sup>:

<http://ofset.sourceforge.net/drgenius/>

y el paquete a bajar `drgenius-0.8.4-0.dag.rh80.i386.rpm`. Para acceder a él, una vez instalado, ejecutamos:

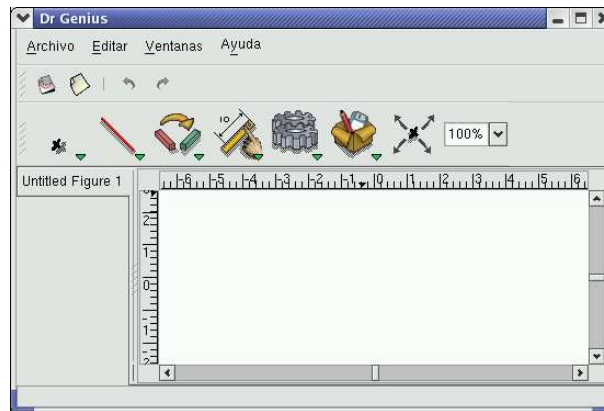
```
$ drgenius
```

<sup>13</sup>La dirección

<http://www.seul.org>

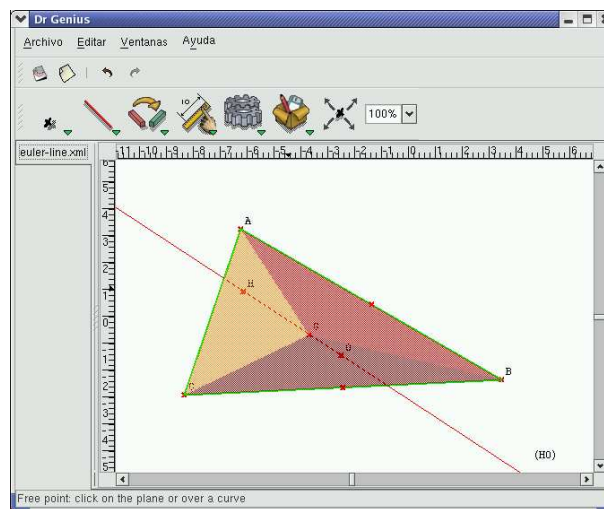
es una página especializada en el uso educativo de Linux. No dejéis de visitarla.

veremos la ventana principal del programa:



Su uso no presenta mayor problema gracias a la ayuda contextual (en castellano) que aparece al pasar sobre cualquiera de los elementos de los distintos barras de herramientas desplegables. Un manual de uso se puede bajar de <http://www.offset.org/drgenius/doc.html>.

Si abrimos algunos de los ejemplos (`/usr/share/drgenius/examples/figures/`) podremos acceder al ejemplo típico sobre la recta de Euler (`euler_line.xml`):

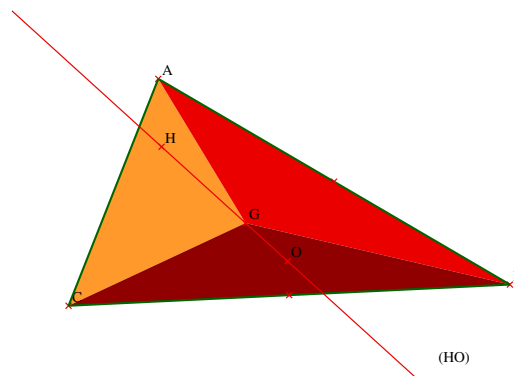


Si exportamos el gráfico anterior a  $\text{\LaTeX}$  obtendremos:

```
\documentclass[a4paper]{minimal}
\usepackage{pstricks}
\begin{document}
\begin{pspicture*}(0,0)(16.166667,9.900000)
\psset{dotstyle=x,dotscale=2.0,linewidth=0.02}
\psframe(0,0)(16.166667,9.900000)
\psdots[dotscale=2.5,linecolor=red](2.150000,8.083334)
\rput[b1]{0}(2.350000,8.283334){A}
\psdots[dotscale=2.5,linecolor=red](-0.250001,2.016667)
\psdots[dotscale=2.5,linecolor=red](11.550000,2.583333)
\rput[b1]{0}(11.750000,2.783333){B}
\psline[linewidth=0.05,linecolor=green](2.150000,8.083334)(11.550000,2.583333)
\psline[linewidth=0.05,linecolor=green](11.550000,2.583333)(-0.250001,2.016667)
```

```
\psline[linewidth=0.05,linestyle=green] (-0.250001,2.016667) (2.150000,8.083334)
\psdots[dotstyle=2.5,linestyle=red] (6.850000,5.333334)
\psdots[dotstyle=2.5,linestyle=red] (5.650000,2.300000)
\psdots[dotstyle=2.5,linestyle=red] (4.483333,4.227778)
\rput [bl] {} (4.683333,4.427778) {G}
\psdots[dotstyle=2.5,linestyle=red] (5.606400,3.207908)
\rput [bl] {} (5.806400,3.407908) {O}
\psdots[dotstyle=2.5,linestyle=red] (2.237200,6.267519)
\rput [bl] {} (2.437200,6.467519) {H}
\psline[linestyle=red] (0.000000,8.299148) (9.138901,0.000000)
\rput [bl] {} (9.738901,0.600000) {(HO)}
\pspolygon[fillstyle=crosshatch, hatchcolor=yellow]
(2.150000,8.083334)
(4.483333,4.227778)
(-0.250001,2.016667)
\pspolygon[fillstyle=crosshatch, hatchcolor=red]
(4.483333,4.227778)
(2.150000,8.083334)
(11.550000,2.583333)
\pspolygon[fillstyle=crosshatch, hatchcolor=red]
(4.483333,4.227778)
(-0.250001,2.016667)
(11.550000,2.583333)
\end{pspicture*}
\end{document}
```

Si lo exportamos a formato .eps el resultado es:



## 24.5.2. Geomview

Geomview es un programa que permite ver y manipular objetos geométricos. Sólo vamos a dar unas ideas sobre su uso y, tanto para él como para los módulos que se instalan después, os recomendamos la propia documentación que acompaña al programa en `/usr/share/doc/geomview-1.8.1/` así como a la web:

<http://www.geomview.org>

La última versión en formato rpm la podemos conseguir en <http://www.geomview.org/download/>, se trata de

`geomview-1.8.1-4.i386.rpm`

Además, podemos disponer de una serie de añadidos:

`geomview-plugins-1.8.1-4.i386.rpm`

`orrery-0.9.3-1.i386.rpm`<sup>14</sup>

<sup>14</sup>Si deseamos añadirle la posibilidad de visualizar un sistema solar instalaremos también este paquete.

Pongámonos “manos a la obra”. Previo a la instalación del programa necesitamos tener instaladas las librerías:

`xforms-0.89-3.i386.rpm`<sup>15</sup>

que podemos bajar desde la zona de descarga del programa.

Si instalamos los cuatro paquetes anteriores<sup>16</sup> e intentamos ejecutar el programa dará error. Para que podamos trabajar con él hemos de actualizar los paquetes:

`glibc-2.2.93-5`

`glibc-common-2.2.93-5`

`glibc-devel-2.2.93-5`

a las últimas versiones disponibles en <https://rhn.redhat.com/errata/RHSA-2003-089.html#RedHatLinux208.0>. Se trata de los paquetes:

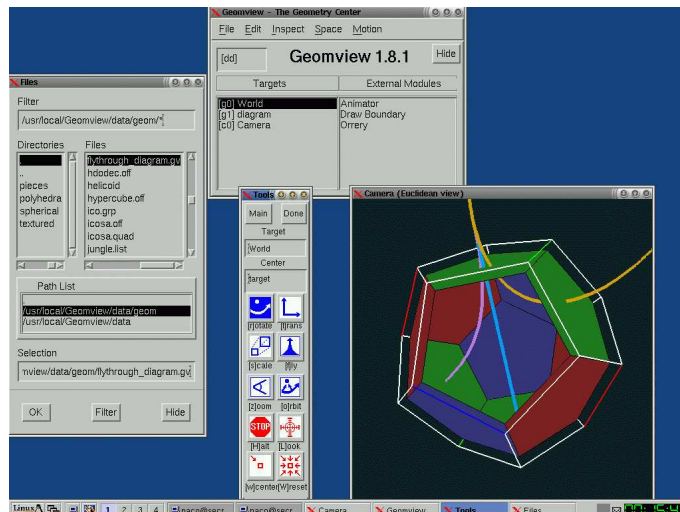
`glibc-2.3.2-4.80.6.i386.rpm`<sup>17</sup>

`glibc-common-2.3.2-4.80.6.i386.rpm`

`glibc-devel-2.3.2-4.80.6.i386.rpm`

Una vez actualizados los paquetes anteriores (con `#rpm -Fvh glibc*`) pasemos a la acción, desde un xterm escribimos:

`$ geomview`



Se trata de probar, puede que con este programa entendamos qué significa ESO de valorar la belleza de la forma.

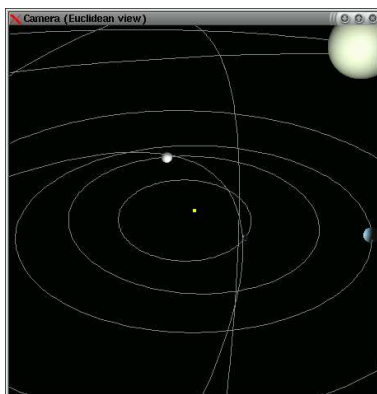
Si hemos instalado el módulo *Orrery* y marcamos sobre él dispondremos de un sistema solar.

<sup>15</sup>Si bien geomview no las usa, sí que son necesarias para algunos de los módulos externos del programa.

<sup>16</sup>El orden de instalación es:

1. `geomview` y `xforms`
2. `geomview-plugins` y `orrery`

<sup>17</sup>El adecuado para nuestro micro.



Como añadido y si disponéis de Mathematica/Maple es posible interactuar con gráficos 3D creados con estos programas, se pueden exportar/visualizar con Geomview.

## Capítulo 25

# Astronomía

El hombre empezó por el principio: "Toda magnitud es igual a sí misma" y acabó midiendo el sol y las estrellas (LITCHTENBERG)

Tan sólo comentar dos programas que me “encantan” para jugar con ellos xephem y xpnls. En algunos aspectos son parecidos y optar por uno u otro es cosa vuestra.

### 25.1. xephem

La página principal es:

<http://www.clearskyinstitute.com/xephem/xephem.html>

De aquí lo podemos bajar en formato rpm<sup>1</sup>

<http://loke.as.arizona.edu/~ckulesa/xephem/>

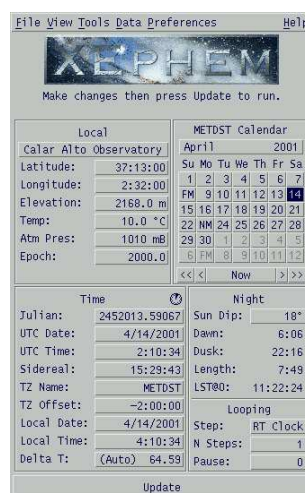
Si instalamos ahora el programa xephem:

```
# rpm -ivh xephem-3.5.2-1.i386.rpm
```

lo ejecutamos con:

```
# xephem &
```

La ventana principal es

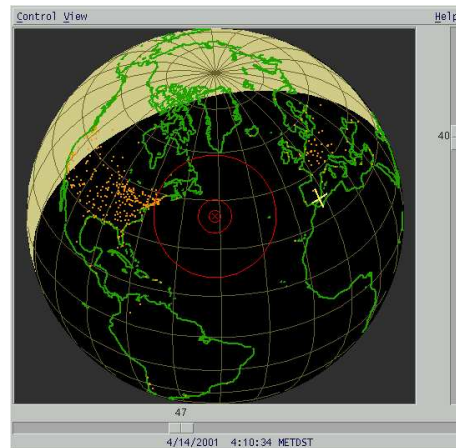


<sup>1</sup> También está puesto en el servidor



para controlar todos los datos actuales de posición, fecha, condiciones atmosféricas.

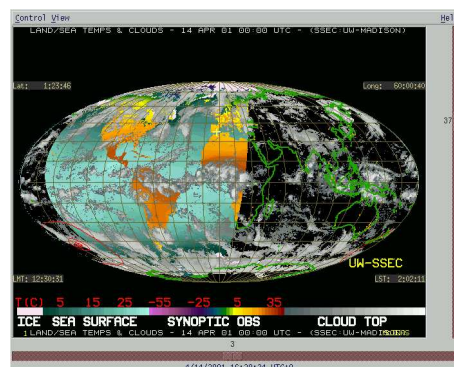
Desde el menú **View** permite visualizar la bóveda celeste, el sistema solar, etc. Por ejemplo si pulsamos sobre el menú **View** y después sobre **Earth** podremos visualizar e incluso animar las zonas iluminadas en la superficie terrestre.



Una opción interesante consiste en que si estamos conectados a Internet y pulsamos sobre **Weather map**



podremos obtener en “tiempo real” un mapa atmosférico de la superficie terrestre:



## 25.2. xplns

La página principal de este programa de astronomía es

<http://www.astroarts.com/products/xplns/index.html>

xplns es un programa de astronomía que permite ver el cielo (incluso animarlo), las órbitas de los planetas, calcular la posición de la mayoría de los cuerpos celestes, etc. Los paquetes necesarios para una instalación completa son:

xplns-3.3.0-1glibc22.i386.rpm → Ejecutable

xplns-cat-3.3.0-1.i386.rpm → Catálogo de estrellas

xplns-elm-3.3.0-1.i386.rpm → Necesario para las órbitas de cometas y planetas menores

xplns-img-3.3.0-1.i386.rpm → Permite conocer datos de los objetos que se “ven” con sólo pulsar sobre ellos

Si los instalamos todos con `rpm -ivh xplns-*`, al ejecutarlo con

```
$ xplns
```

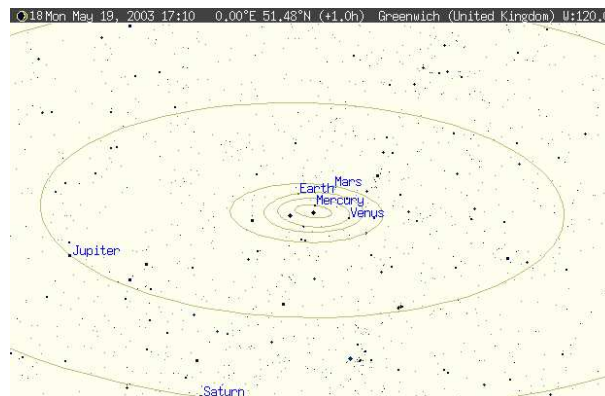
desde un xterm, no se iniciará el programa y es que no reconoce el lenguaje. Podemos “engañarlo” y especificar el inglés:

```
$ xplns -l=en
```

la primera vez que ejecutemos el programa, para que almacene los cambios y no haya que pasarle ese parámetro hay que pulsar sobre **Setting**→**Environment** y aceptar en la ventana que aparece<sup>2</sup>. Después ya no dará mas errores y podremos ejecutarlo con el comando:

```
$ xplns
```

veremos la ventana principal de la aplicación. Pulsando sobre los distintos iconos de la barra de herramientas podremos visualizar y comprobar las distintas posibilidades del programa



Además, si pulsamos sobre el icono  podremos conseguir animaciones.

<sup>2</sup>Acabamos de modificar el fichero `~/xplns/option`, en él hemos guardado los cambios.

# **Parte VI**

## **Redes en Linux**

## Capítulo 26

# Introducción a los protocolos TCP/IP.

El concepto de trabajo en redes es probablemente tan antiguo como lo es el de las telecomunicaciones.

Imagínese por un momento, gente viviendo en la Edad de Piedra, en donde los individuos usen tambores para transmitirse mensajes. Supóngase que un hombre de las cavernas A quiere invitar a otro hombre B a una partida de choques de piedra. Lamentablemente viven tan distantes, que a B le sería imposible escuchar el tambor de A cuando éste lo llame. ¿Qué puede hacer A para remediar esto? Él podría 1) ir caminando al sitio de B, 2) conseguir un tambor más grande, o 3) pedirle a C, quien vive a mitad de camino que reenvíe el mensaje. La tercera elección es denominada Trabajo en Redes. (*Guía de Administración de Redes Segunda Edición*, OLAF KIRCH)

### 26.1. Introducción

Linux, tal como lo conocemos, es posible gracias a “La Red”<sup>1</sup> y “La Red” se ha expandido también en gran parte, gracias a Linux. Todos sabemos la importancia de Internet y en este capítulo vamos a ver cómo configurar nuestra máquina Linux para que se integre en una red TCP/IP<sup>2</sup> y dotarla de algunos servicios que nos pueden ser de utilidad.


Un documento interesante que versa sobre redes es:

- *Guía de Administración de Redes*, de OLAF KIRCH. Traducción de la versión 1.0 y 2.0 del libro *Linux Network Administration Guide*. Es un libro sobre administración avanzada de servicios en red con Linux.

Otro documento interesante sobre este tema es:

- *Administración de una Red Local basada en Internet*.
- *Introducción a la administración de redes tcp-ip*, de CHARLES L. HEDRICK

Están a vuestra disposición en Lucas (<http://es.tldp.org>).

Para no tener problemas con las configuraciones de los servicios de red y si al configurar la tarjeta de red establecimos algún tipo de *firewall*, es preferible ejecutar el programa (también se puede acceder a él con `#setup→firewall configuration` o si nos gusta más el modo gráfico con →**Configuración del sistema→Security Level**):

```
# lokkit
```

---

<sup>1</sup>Nos referimos a Internet

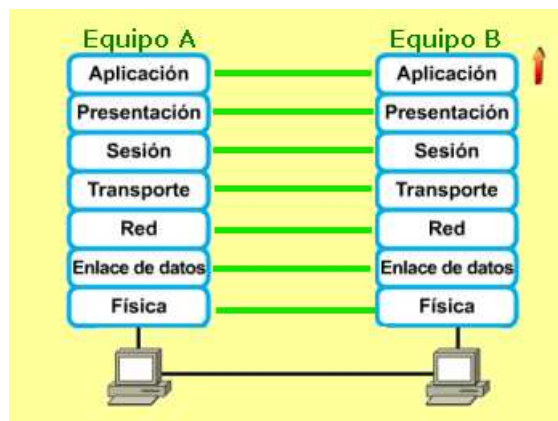
<sup>2</sup>Protocolos de comunicación que utiliza Internet



y optar por **No firewall** en espera de llegar a la página 400.

## 26.2. Protocolos de Red

**Protocolo** Es un conjunto de normas y procedimientos para la transmisión de datos que ha de ser observado por los dos extremos de un proceso de comunicaciones (emisor y receptor<sup>3</sup>). El modelo OSI (*Open Systems Interconnection*) definido por la ISO (*International Standards Organization*) propone un modelo en capas en donde cada una de las capas se ocupa de una determinada tarea y presenta un interfaz bien definida para sus capas adyacentes (superior e inferior). Las capas del modelo OSI son, desde la superior a la inferior: Aplicación, Presentación, Sesión, Transporte, Red, Enlace y Físico. Entre los protocolos del nivel físico se encuentran Ethernet o Token Ring.



Ejemplos de algunos protocolos son:

**IPX/SPX** (*Internetwork Packet Exchange*) protocolo para redes Novell Netware.

**NetBIOS** (*Network Basic Input/Output System*) desarrollado por IBM fué adoptado después por Microsoft.

**NetBEUI** (*NetBIOS Extended User Interface*) versión mejorada de NetBIOS, es el protocolo predominante en las redes Windows.

**AppleTalk** protocolo de comunicación para ordenadores Apple Macintosh.

**TCP/IP** conjunto de protocolos: TCP (*Transmission Control Protocol*, se encuentra en el nivel de transporte) e IP (*Internet Protocol*, se encuentra en el nivel de red). Ellos rigen las comunicaciones en Internet. Estos protocolos pertenecen a un conjunto mayor de protocolos que se llama suite o conjunto de protocolos TCP/IP<sup>4</sup>. Se describen a partir de la pila de capas que, aunque no tienen una

<sup>3</sup>Y todos los elementos intermedios si los hubiere

<sup>4</sup>

equivalencia total con el modelo OSI, se asemejan a él: capa de subred (tramas Ethernet y direcciones MAC; cable de cobre, de fibra óptica, etc), capa de red/internet (paquetes IP, dirección IP y mensajes ICMP), capa de transporte (puertos de servicio y protocolos TCP y UDP) y capa de aplicación (programas de cliente y de servidor: servidor ftp, cliente telnet, etc).

## 26.3. Introducción a las direcciones IP.

Las direcciones del Protocolo Internet (IP) están compuestas por cuatro *bytes* (para la versión IPv4). Estas direcciones desde hace tiempo se dice que están llegando a su agotamiento<sup>5</sup> y la próxima versión del protocolo (IPv6) dispondrá de un rango de direcciones mucho más elevado<sup>6</sup>.

Cada máquina de una red TCP/IP<sup>7</sup> debe tener una dirección IP única. Como Internet es una red TCP/IP, no deben existir dos direcciones IP iguales en ella.

La convención es escribir estas direcciones en la denominada *notación decimal separada por puntos* (*dotted decimal notation*). De esta forma cada byte (octeto) es convertido en un número decimal (en el rango 0-255). Por ejemplo, una dirección válida sería: 192.168.1.5

Cada interfaz de una máquina de la red o de un encaminador (router) tiene una dirección IP. Por *interfaz de red* entendemos el dispositivo que nos une a la red, por ejemplo una tarjeta de red.

Internet se forma mediante la unión de muchas redes IP y mecanismos para pasar de unas a otras. Todas las direcciones dentro de una misma red IP tienen un número de bits en común. A la parte de la dirección IP que es común a todas las direcciones que se encuentran en una red IP, se le llama la *porción de la red*. Los bits restantes son llamados *porción de la máquina*. Al número de bits que comparten todas las direcciones de una red se le llama máscara de red (*netmask*), y su papel es determinar qué direcciones pertenecen a la red y cuáles no. Las máquinas que pertenecen a una misma red IP, pueden comunicarse directamente unas con otras. Para comunicarse con máquinas de una red IP diferente, deben hacerlo mediante mecanismos de interconexión como *routers*, *proxys* o *gateways*. Consideremos el siguiente ejemplo:

	Decimal	Binario
Dirección del Host	192.168.1.5	11000000.10101000.00000001.00000101
Máscara de red	255.255.255.0	11111111.11111111.11111111.00000000
Porción de red	192.168.1.	11000000.10101000.00000001.
Porción de Host (máquina)	.5	.00000101
Dirección de Red	192.168.1.0	11000000.10101000.00000001.00000000
Dirección de Difusión (o broadcast)	192.168.1.255	11000000.10101000.00000001.11111111

Por host nos referimos a una máquina, ya sea un servidor o un puesto de usuario final. En este caso, nuestro querido Linux.

Cualquier dirección a la que se aplique una operación “and”<sup>8</sup> de bits con su máscara de red, revelará la dirección de la red a la que pertenece. La dirección de red es por tanto siempre el menor número de dirección dentro del rango de la red y siempre tiene la porción de máquina codificada toda con ceros. Como anteriormente, 192.168.1.0.

La dirección *de difusión* (*broadcast*) es una dirección especial a la que escucha cada máquina en la red IP además de a la suya propia. Esta dirección es a la que se envían los paquetes de datos si se supone

- TCP permite controlar y gestionar el proceso por el que la información se divide en unidades individuales de datos (llamadas paquetes) y cómo son ensamblados de nuevo en la máquina de destino.
- IP se encarga de llevar los paquetes de una máquina a otra haciendo uso de las direcciones IP de las máquinas locales y remotas.

<sup>5</sup>Mediante diversas técnicas, como el mejor aprovechamiento de las direcciones o la utilización de direcciones de red privadas dentro de las organizaciones, se ha conseguido frenar este proceso

<sup>6</sup>Como unas cuantas por cada metro cuadrado de la tierra, posibles gracias a 128 bits de direccionamiento

<sup>7</sup>De igual manera que cada persona en España tiene un número de DNI

<sup>8</sup>O “Y” lógico

que todas las máquinas de la red lo deben recibir. Ciertos tipos de datos, como la información de encaminamiento y los mensajes de aviso son transmitidos a la dirección de difusión para que cada estación en la red pueda recibirlo simultáneamente. Para ello se utiliza la dirección más alta posible en la red, conseguida con la porción de red a la que se añade el resto de bits (de la porción de host) todos con valor a uno (1). En el ejemplo anterior sería 192.168.1.255.

Por razones administrativas, durante el desarrollo inicial del protocolo IP se formaron, de forma un poco arbitraria, algunos grupos de direcciones como redes, y estas redes se agruparon en las llamadas *clases*. Estas clases proporcionan un cierto número de redes de tamaño estándar que pueden ser reservadas. Los rangos reservados son<sup>9</sup>:

Clase de red	Máscara de red	Direcciones de red desde-hasta
A	255.0.0.0	0.0.0.0-127.255.255.255
B	255.255.0.0	128.0.0.0-191.255.255.255
C	255.255.255.0	192.0.0.0-223.255.255.255
Multicast	240.0.0.0	224.0.0.0-239.255.255.255

**Redes Clase A (/8).** Cada IP en una red de clase A posee un prefijo de red de 8 bits (con el primer bit puesto a 0 y con lo que quedan todavía 7 bits), seguido por un número de host de 24 bits, esto hace que se dispongan de:

- $2^7 - 2 = 126$  redes de clase A con
- $2^{24} - 2 = 16,777,214$  hosts cada una<sup>10</sup>

**Redes Clase B (/16).** Tienen un prefijo de red de 16 bits (los dos primeros puestos a 1-0 y un número de red de 14 bits), seguidos por un número de host de 16 bits. Por tanto tendremos:

- $2^{14} = 16,384$  redes de este tipo con
- $2^{16} - 2 = 65,534$  máquinas diferentes como máximo cada una.

**Redes Clase C (/24).** Cada dirección de red clase C tiene un prefijo de red de 24 bits (siendo los tres primeros 1-1-0 con un número de red de 21 bits) y sólo 8 bits para indicar el número de máquina.

- $2^{21} - 1 = 2,097,151$  redes posibles con un máximo de
- $2^8 - 2 = 254$  host por red.

El organismo encargado a nivel internacional de la asignación de direcciones en Internet es la IANA (*Internet Assigned Number Authority*)<sup>11</sup>, aunque su forma de funcionamiento es descentralizada, delegando por zonas el trabajo. A alguno de sus organismos delegados hay que dirigirse para obtener el número oficial de red en función de nuestras necesidades. Sin embargo, los usuarios finales de internet, en general, obtienen sus direcciones IP de forma dinámica a través de sus proveedores de acceso. Estos proveedores a su vez las obtienen o bien de otro proveedor, o bien del registro regional correspondiente.

Las direcciones que usemos dependen de lo que se vaya a hacer exactamente.

<sup>9</sup>Esta división es cada vez menos utilizada y se expone por razones históricas

<sup>10</sup>Hay que quitar dos números de red (las redes la "no red": 0.0.0.0 y la red de *loopback* o bucle interno: 127.0.0.0) y dos números de host en cada caso: no pueden ser asignados ni a ninguna red ni a ninguna máquina ya que se usan para otros propósitos (dirección de red y dirección de difusión)

<sup>11</sup>A nivel Europeo RIPE NCC, y a nivel nacional, tras haber sido anteriormente ES-NIC, relacionado con la red académica Rediris, actualmente se ocupa de ello Red.es

### Instalar una máquina Linux en una red IP existente

Si deseamos instalar una máquina Linux en una red IP existente entonces deberíamos contactar con los administradores de la red y preguntarles por la siguiente información:

- Dirección IP del Host
- Dirección IP de la red
- Dirección IP de broadcast
- Máscara de red IP<sup>12</sup>
- Dirección del encaminador (router)
- Dirección del Servidor de Nombres de Dominio (DNS)

### Redes privadas no conectadas a Internet.

Si estamos construyendo una red privada y no tenemos intención de conectar nunca esa red a Internet entonces podríamos elegir las direcciones que queramos, pues no vamos a colisionar con nadie.

Por razones de seguridad y consistencia, se han reservado algunas direcciones IP de red específicamente para este propósito. Están descritas en el RFC1597 y son las que siguen:

DIRECCIONES RESERVADAS PARA REDES PRIVADAS		
Clase de red	Máscara de red	Direcciones de red desde-hasta
A	255.0.0.0	10.0.0.0 - 10.255.255.255
B	255.255.0.0	172.16.0.0 - 172.31.255.255
C	255.255.255.0	192.168.0.0 - 192.168.255.255

Estas direcciones no se corresponden con las de ninguna máquina en Internet y no se encaminarán a través de los “routers” de Internet. Podremos utilizarlas de forma interna, con la ventaja de que si conectamos mediante un proveedor a Internet, nuestras direcciones no coincidirán con las de ninguna máquina de Internet.

En este caso, puede que haya multitud de equipos en redes distintas con estas mismas direcciones IP<sup>13</sup> pero como estas direcciones no se “rutean” no hay por qué coordinar su uso con las agencias de registro. En el caso de que se conecten a Internet, se habilitan mecanismos como la traducción de direcciones (NAT<sup>14</sup>) o el uso de intermediarios (proxys) para que las direcciones que salgan a Internet sean direcciones válidas.

---

<sup>12</sup>Conociendo la dirección IP de la máquina (host) y la máscara de red, podemos conocer la dirección de la red y la de broadcast. Prueba a hacerlo con la dirección de máquina 150.214.5.11 y la máscara de red 255.255.255.0

<sup>13</sup>Dentro de cada red, cada equipo tendrá direcciones IP diferentes: dos personas que no viajan (direcciones IP que no se “rutean”) pueden tener iguales DNI en estados diferentes pero nunca dentro del mismo país.

<sup>14</sup>Network Address Translation o Traducción de Direcciones de Red



## Capítulo 27

# Construir una red con direccionamiento de red privado.

Sin esta conexión (Internet), nuestras máquinas serían islas de ciclos de procesador que ejecutarían programas únicamente de nuestro interés, y totalmente blindadas contra todo aquel que no tenga una llave de tu oficina. (*Hackers en Linux*, BRIAN HATCH Y OTROS)

### 27.1. Introducción

En nuestros ejemplos, realizaremos la configuración de una máquina en la red 192.168.1.0<sup>1</sup>, con dirección 192.168.1.5 y cuya máscara de red sea 255.255.255.0. El rango de direcciones en la red será desde la 192.168.1.0 a la 192.168.1.255. Tendremos en total hasta 254 direcciones posibles en nuestra red local de las 256 que nos permite el último byte ( $2^8$ ), porque dos direcciones, como hemos visto, tienen un propósito específico (la dirección de la propia red y la de *broadcast*).

La 192.168.1.0 es la dirección de la red y por tanto no utilizable como dirección por ninguna máquina.

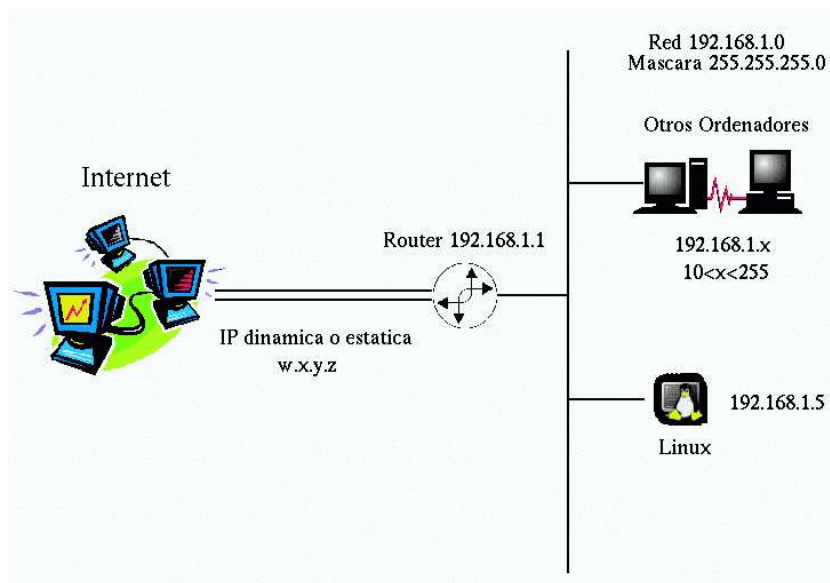
La 192.168.1.255 es la dirección de broadcast y sirve para referirse a todas las máquinas de la red.

Utilizaremos la dirección 192.168.1.1 como la dirección del router que nos sacará hacia Internet, que también se suele denominar *gateway*.

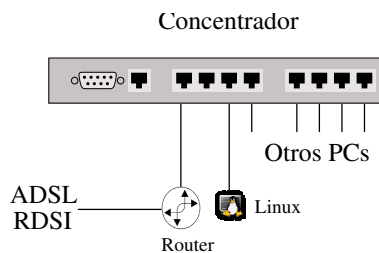
Para usos futuros de la red (por ejemplo, otros servidores que incorporemos) reservamos hasta la dirección 192.168.1.10, con lo que los otros ordenadores de la red pueden ir desde la 192.168.1.11 hasta la 192.168.1.254. Esta red se puede implementar en un centro educativo.

---

<sup>1</sup>Al tomar esta dirección para la red (véase la tabla anterior) nos garantizamos que si accedemos a Internet mediante un proveedor de acceso no tendremos problemas de encaminamiento.



Esa es la vista lógica de nuestra red. La vista física será un hub (o concentrador)<sup>2</sup>, que forma nuestro segmento de red y al que irán conectados los ordenadores de la red, el servidor linux y el router. Además, el router dispondrá de una salida ADSL/RDSI hacia el exterior.

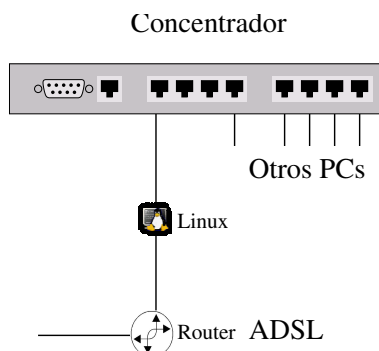


**Importante:** se está “construyendo” una red con la estructura estándar de Averroes o sin modificar el esquema de los routers ADSL. Se parte de la base de que los centros disponen del router que les ha mandado la JUNTA DE ANDALUCÍA o que en el caso de trabajar con una conexión ADSL se trabaja con NAT y multipuesto. En el caso de no disponer de router<sup>3</sup> o configurar el Router ADSL en monopuesto podría ser la máquina Linux la encargada de realizar el enrutamiento. En consecuencia, si optamos por un equipo con Linux (y un módem si deseamos salir al exterior) el router, aquí introducido como un elemento más de la red, podría ser innecesario.

En ese caso el esquema sería

<sup>2</sup> Mejor un switch (o conmutador).

<sup>3</sup> O tener demasiados problemas con él.



El modelo anterior es el más sencillo. Podemos querer dividir la red de los alumnos de la red administrativa y/o de la conexión al exterior por cuestiones de control y seguridad. Por ejemplo, dándoles distintas direcciones de red: una red la 192.168.1.0/24<sup>4</sup> y otra la 192.168.2.0/24. En dicho caso, un sistema Linux con tres tarjetas de red y unido a los dos concentradores y al router ADSL, a cada uno mediante una tarjeta de red y una dirección correspondiente a esa red, podría hacer de unión<sup>5</sup> entre las tres redes (dos redes locales y la conexión a internet).

La configuración la iremos viendo mediante los comandos necesarios y ficheros involucrados, para posteriormente ver una forma más sencilla de configurarlos; normalmente desde un interfaz gráfico. La razón es que el conocer qué es lo que hay por debajo nos ayuda a un mejor entendimiento y nos "protege" frente a fallos de los programas de configuración automática, que cuando ocurren, no sabemos hacia dónde dirigir nuestros pasos (bueno, siempre está el "experto local", aunque ese podemos ser nosotros mismos).<sup>6</sup>

## 27.2. Creación de las interfaces de red.

En la mayoría de los casos, los dispositivos de red son creados automáticamente por el controlador de dispositivos mientras se inicia y localiza el hardware. Por ejemplo, el controlador Ethernet (para nuestra tarjeta de red local) crea interfaces `eth0`, `eth1`... secuencialmente según va encontrando tarjetas Ethernet.

Sin embargo, en algunos casos, como interfaces de líneas de modem `slip` o `ppp`, los dispositivos de red son creados por la acción de algún programa de usuario. Se aplica la misma numeración secuencial de dispositivos, pero no se crean al arrancar. La razón es que al contrario que con los dispositivos Ethernet, el número de dispositivos `ppp` o `slip` puede variar durante la actividad de la máquina.

Este paso de detección normalmente lo realiza el kernel automáticamente al arrancar si dispone de soporte para nuestra tarjeta de red. En caso de que no se encuentre el soporte en el kernel para nuestra tarjeta de red, deberemos buscar un kernel que la soporte o compilarlo si fuera necesario<sup>7</sup>.

Cuando hablamos de configurar una interfaz de red nos referimos al proceso de asignar direcciones apropiadas a un dispositivo (tarjeta de red) y asignar valores adecuados a otros parámetros configurables. El programa usado para hacer esto es la orden `ifconfig` (*interface configuration*).

Si lo hacemos mediante línea de comandos, sería una orden como:

```
# ifconfig eth0 192.168.1.5 netmask 255.255.255.0 up
```

<sup>4</sup>Esta es otra notación para las redes. Significa que la dirección de la red es la 192.168.1.0 y la máscara correspondiente son 24 bits empezando por la izquierda (255.255.255.0 ó 8+8+8+0 ó /24)

<sup>5</sup>Funcionando como router.

<sup>6</sup>Suponemos que el hardware está ya instalado, si no, puede verse el artículo 134 de la revista LinuxFocus que habla sobre redes caseras y realiza un repaso al hardware necesario

<http://www.linuxfocus.org/Castellano/January2000/article134.html>

<sup>7</sup>Rara vez ocurre esto

En este caso estoy configurando la interfaz Ethernet `eth0`, con dirección IP 192.168.1.5 y máscara de red 255.255.255.0. El `up` del final de la orden le dice al interfaz que debería activarse, pero normalmente se puede omitir, ya que es el valor por defecto. Para desactivar una interfaz, simplemente hay que ejecutar<sup>8</sup>

```
#ifconfig eth0 down
```

Para un sistema RedHat en modo gráfico, podemos lanzar desde Gnome



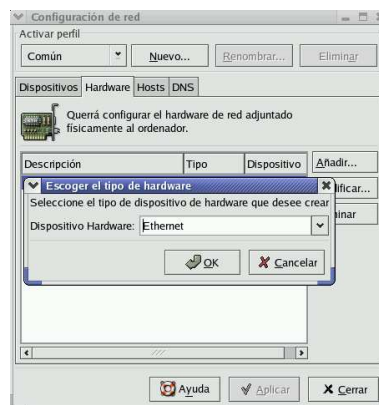
→ **Configuración del Sistema** → **Red**

o directamente desde una `xterm`<sup>9</sup>

```
# neat &
```



Lo usual es que la tarjeta haya sido detectada y configurada en el arranque/instalación y desde esta ventana podremos editarla. Si no es así, pulsando sobre **Añadir** podemos configurar nuestra nueva interfaz de red de área local.



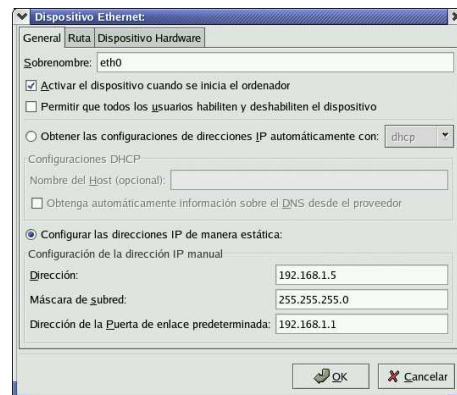
Una vez definidos correctamente los valores hardware de nuestra tarjeta, tendremos que configurar los dispositivos de red asociados con el hardware físico (pestaña **Dispositivos**). Pulsaremos sobre **Añadir**,

<sup>8</sup>O

`#/sbin/ifdown eth0`

<sup>9</sup>O con `redhat-config-network`. La captura gráfica no se corresponde con la que aparecería por defecto.

seleccionamos **Conexión Ethernet** y debemos elegir la tarjeta correspondiente<sup>10</sup>



Tendremos la posibilidad de permitir que la configuración se obtenga de un servidor de alguno de estos protocolos (DHCP, BOOTP) que se la proporcionará al arrancar, o bien, si marcamos la casilla *Configurar las direcciones IP de manera estática* podremos introducir la dirección IP (192.168.1.5), la máscara de red (255.255.255.0) y la *Dirección de la Puerta de enlace predeterminada* (192.168.1.1).

Si tenemos un router ADSL u otro Linux que hace de pasarela, es el momento de poner aquí su dirección IP para que podamos salir al exterior. Si no, debemos dejar esta casilla en blanco.

Mediante el comando:

```
$/sbin/ifconfig -a
```

podemos ver las interfaces que tenemos configuradas:

```
eth0 Link encap:Ethernet HWaddr 00:00:E8:3B:2D:A2
      inet addr:192.168.1.5 Bcast:192.168.1.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:1359 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1356 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:11 Base address:0xe400

lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:3924 Metric:1
      RX packets:53175 errors:0 dropped:0 overruns:0 frame:0
      TX packets:53175 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
```

que nos dice que la interfaz `eth0` tiene la dirección hardware (00:00:E8:3B:2D:A2). Ésta es una dirección de la propia tarjeta y cada fabricante se ocupa de que sea única en el mundo. Se denomina dirección MAC. Es una dirección física, que normalmente no se puede cambiar; al contrario que la dirección IP, que es una dirección lógica y puede modificarse. Cuando estamos en una red de área local (LAN) son éstas las direcciones mediante las que se comunican las máquinas. Existe un protocolo llamado ARP (Protocolo de resolución de direcciones) que se encarga de preguntar y traducir entre direcciones IP y direcciones MAC.

<sup>10</sup>Normalmente seleccionaremos la opción de activar el interfaz en el arranque (**Activar dispositivo cuando arranca el sistema**), pero no permitiremos que cualquier usuario pueda desactivarla en un sistema en el que puede haber usuarios que no son administradores del sistema (**Permitir a todos los usuarios ...**).



Comprobamos si se ha configurado con la dirección de red, broadcast y máscara de red deseadas. Si se encuentra en modo UP es que está habilitada.

La interfaz `lo` (loopback) es una interfaz virtual que se refiere a nuestra propia máquina, sin salir por ninguna interfaz de red, sólo para uso interno. Su dirección es siempre 127.0.0.1.

Si disponemos de un router de IP 192.168.1.1 y ejecutamos desde nuestra máquina Linux<sup>11</sup>

```
$ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1): 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.9 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.8 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.9 ms

--- ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.8/0.8/0.9 ms
```

Estamos mandando paquetes por la red al router. Si el resultado ha sido como el anterior, podemos llegar a éste de forma satisfactoria.

Si por el contrario, la salida nos indica que no puede llegar (100% *packet loss*) al router, debemos revisar nuestras conexiones de red.

```
$ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) from 0.0.0.0 : 56(84) bytes of data.

--- 192.168.1.1 ping statistics ---
11 packets transmitted, 0 packets received, 100% packet loss
```

El comando `ping` es una utilidad de red muy práctica. Trabaja de la siguiente forma:

Nuestra máquina envía paquetes de datos a la dirección de destino (en este caso 192.168.1.1) que, automáticamente, nos responde como si fuera un *boomerang*. Si todo es correcto, nos llegarán los paquetes de vuelta a nuestra máquina.

## 27.3. Configurando el encaminamiento (routing).

*El Encaminamiento IP es el proceso por el que una máquina decide por dónde dirigir un paquete IP que haya recibido.*

Ya tenemos nuestras direcciones IP asignadas. Continuemos ¿Cómo funciona el encaminamiento? Cada máquina tiene una lista de reglas, llamada tabla de encaminamiento (o tabla de *routing*). Esta tabla contiene columnas que suelen contener al menos tres campos:

- Una dirección de red de destino,
- El nombre de la interfaz a la que se va a encaminar el paquete, y
- Opcionalmente, la dirección IP de otra máquina que cogerá el paquete en su siguiente paso a través de la red si es que no podemos llegar directamente a ella.

En Linux se puede ver esta tabla usando las órdenes:

```
# /sbin/route -n
o
```

---

<sup>11</sup>Se detiene el comando con las teclas [CTRL]+[C].



```
# /bin/netstat -nr
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
127.0.0.0 0.0.0.0 255.0.0.0 U 0 0 0 lo
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
0.0.0.0 192.168.1.1 0.0.0.0 UG 0 0 0 eth0
```

Veamos su significado. Los paquetes con destino a la red 127.0.0.0/255.0.0.0 (ponemos la dirección de la red seguida de su máscara) deben ser enviados a la interfaz `lo`. Los paquetes con destino a la red 192.168.1.0/255.255.255.0 (por ejemplo la dirección 192.168.1.1) deben ser enviados al interfaz `eth0` (la primera tarjeta de red). Para el resto de redes que no están especificadas explícitamente (especificado como la dirección 0.0.0.0) debemos mandar los paquetes a través de un intermediario (el router 192.168.1.1) para que sea éste el que se ocupe de mandarlos a su destino. Es decir, le decimos "yo no sé cómo mandar paquetes a la dirección 150.214.4.34, ocúpate tú de ello". En cierto modo es lo más fácil y, además, como ese es el trabajo del router, lo hará bien.

El comando para la ruta al interfaz de loopback se crea automáticamente. Para crear la ruta a nuestra red local utilizamos el comando:

```
#/sbin/route add -net 192.168.1.0 netmask 255.255.255.0 eth0
```

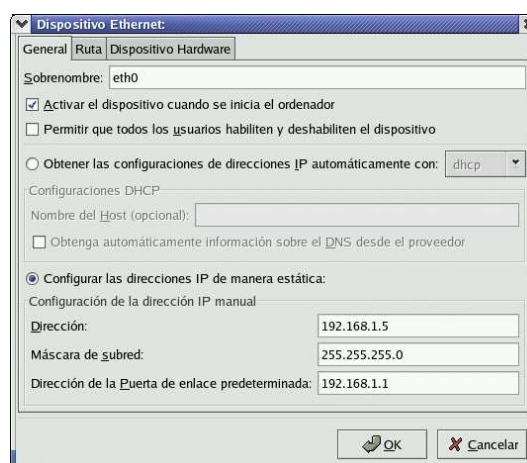
que quiere decir: para la red 192.168.1.0 con máscara 255.255.255.0 dirige los paquetes por el interfaz `eth0`. Esta ruta le permitirá establecer conexiones IP directamente con todas las estaciones de su segmento Ethernet; aquel accesible directamente mediante la tarjeta de red `eth0`.

Para crear la ruta por defecto, introducimos:

```
#/sbin/route add default gw 192.168.1.1
```

Añade la ruta por defecto<sup>12</sup> (cuando no encontramos una ruta específica para salir) y manda los paquetes a la dirección 192.168.1.1, que ya se encargará de dar salida a los paquetes<sup>13</sup>.

Ya hemos visto cómo usar la interfaz gráfica de RedHat 8 (*neat*), utilicémosla ahora para añadir las rutas. Si hemos configurado la tarjeta con la dirección de red de nuestro segmento local, esta ruta ya se habrá añadido. Para añadir la ruta por defecto, hemos puesto la dirección del router en el campo **Dirección de la Puerta de enlace predeterminada** (192.168.1.1)



<sup>12</sup>Denominada "default"

<sup>13</sup>Si no tenemos router ni ninguna forma de conectarnos a Internet (cada vez más inusual), no especificaremos esta ruta.

Para comprobar que funciona correctamente podemos hacer `ping` a una máquina de fuera de nuestra red y ver si realmente los paquetes pueden salir al exterior.

```
$ping 150.214.4.34
```

Si nos llegan los paquetes de vuelta, estupendo. En caso contrario, debemos asegurarnos de que hemos realizado correctamente la configuración de las rutas, y por si acaso, reiniciado nuestra máquina.

Si tenemos que añadir rutas adicionales<sup>14</sup>, seleccionamos el dispositivo (`eth0`) y en la opción **Modificar**, seleccionamos la pestaña **Ruta**. Para *Añadir* una nueva ruta, debemos especificar la dirección de la red, su máscara y la puerta de enlace para llegar a ella.

## 27.4. Configuración del sistema DNS

La función del *sistema de resolución de nombres* (DNS) es proporcionar un servicio para convertir las denominaciones “amigables para el hombre” de las máquinas como `thales.cica.es` a direcciones IP “amigables para la máquina” como `150.214.5.10`.

Los nombres de dominio de Internet son jerárquicos por naturaleza, tienen una estructura en árbol invertido. Un dominio puede ser fragmentado en subdominios. Un *dominio de nivel superior* (toplevel domain) es un dominio especificado en la parte más a la derecha del nombre de máquina o dominio. Los Dominios de Nivel Superior están especificados en el RFC-920. Algunos ejemplos de los más comunes son:

**COM** Organizaciones Comerciales

**EDU** Organizaciones Educativas

**GOV** Organizaciones Gubernamentales

**MIL** Organizaciones Militares

**ORG** Otras organizaciones, normalmente sin ánimo de lucro

**NET** Organizaciones relacionadas con InterNet

**Designador de País** éstos son códigos de dos letras que representan a un país en particular (**es**=España).

Cada uno de estos dominios de nivel superior tiene subdominios. Los dominios de nivel superior basados en el nombre de un país suelen estar divididos en subdominios basados en `com`, `edu`, `gov`, `mil` y `org`. Por ejemplo, encontrará cosas como `com.au` y `gov.au`, las organizaciones comerciales y gubernamentales en Australia<sup>15</sup>.

El siguiente nivel de división suele representar el nombre de la organización. Los siguientes subdominios varían, a menudo basándose en el siguiente nivel en la estructura departamental de la organización a la que pertenecen, pero pueden estarlo en cualquier criterio considerado razonable y con significado claro para los administradores de la red de la organización. Por ejemplo, `cec.juntadeandalucia.es` (Consejería de Educación y Ciencia, dentro de la Junta de Andalucía y dentro de España)

La parte más a la izquierda de un nombre de máquina es el nombre único asignado a la máquina, y es llamada *hostname*, la porción del nombre a la derecha del nombre de la máquina es el *domainname* (nombre de dominio), y el nombre completo es llamado *Fully Qualified Domain Name* (Nombre de Dominio Completamente Cualificado). Por ejemplo para el nombre completo `thales.cica.es`, el *hostname* sería `thales` y el *domainname* sería `cica.es`.

El software de resolución de nombres proporciona el servicio de traducción haciendo consultas a un Servidor de Nombres de Dominio (*Domain Name Server*), por lo que deberá saber la dirección IP del servidor de nombres (*nameserver*) que vaya a usar.

También podemos hacer resoluciones de nombres de forma local en nuestra máquina. El fichero `/etc/hosts` es donde se pone el nombre y dirección IP de las máquinas locales. En este fichero, ponemos la equivalencia

<sup>14</sup>No es lo normal. En el 99 % de los casos debería bastarnos con la ruta de nuestra red local y la Puerta de Enlace predeterminada.

<sup>15</sup>Esto ya se está empezando a aplicar en España por Red.es





entre el nombre de una máquina y su dirección IP. La desventaja de este método frente a los servidores de nombres es que habrá que poner el fichero al día si la dirección de alguna máquina cambia. En un sistema normal, las únicas entradas que suelen aparecer son la interfaz de loopback (prueba en bucle) y el nombre de la máquina local.

```
#more /etc/hosts
127.0.0.1 localhost loopback
192.168.1.5 linux linux.cec.juntadeandalucia.es mailhost
```

Se puede especificar más de un nombre de máquina por línea, normalmente el nombre de la máquina, su nombre completo con el dominio y posibles alias (o nombres alternativos) de la máquina.

En el fichero `/etc/nsswitch.conf`, en la entrada con el nombre `hosts` se especifica qué recursos se utilizan para resolver los nombres de máquinas.

```
hosts: files nisplus nis dns
```

En este caso, vemos que se utiliza primero `files`, yendo al fichero `/etc/hosts`, después `nisplus` y `nis` (son poco utilizados y no nos detendremos en ellos) y después al sistema `dns`, a los servidores de nombres.

El fichero `/etc/resolv.conf` es el de configuración principal del código de resolución de nombres. Su formato es bastante simple. Es un fichero de texto con una palabra clave por línea. Hay tres palabras clave de uso frecuente, que son:

**domain** esta palabra clave especifica el nombre de dominio local.

**search** ésta especifica una lista de dominios alternativos para completar el nombre de una máquina.

**nameserver** ésta, que puede utilizarse hasta tres veces, especifica una dirección IP de un servidor de nombres de dominio para consultarlo cuando se resuelvan nombres.

Por ejemplo, nuestro `/etc/resolv.conf` podría parecerse a éste:

```
domain cec.juntadeandalucia.es
search cec.juntadeandalucia.es juntadeandalucia.es
nameserver 150.214.3.9
nameserver 150.214.90.11
nameserver 150.214.4.34
```

Las palabras clave `domain` y `search` no pueden aparecer juntas. Cogería solamente la última.

Desde Redhat 8, esta configuración la podemos hacer desde

```
#neat
```

Pulsamos en la pestaña DNS e introducimos las IP de nuestros servidores de nombres. Se trata de rellenar los datos necesarios en estos campos, necesitamos conocer el nombre de nuestro servidor de Internet, que lo escribiremos en el campo Nombre del host (no es necesario) y los números DNS de nuestros servidores de nombres. En el caso de la red Averroes escribiríamos como DNS 150.214.3.9, 150.214.90.11 y 150.214.4.34, que serían los DNS primario, secundario y terciario. Quedaría:



Llegado a este punto, pulsamos sobre el botón **Aplicar** y cerramos.

Para comprobar que realmente resolvemos los nombres, podemos hacer

```
$ping thales.cica.es
```

Lo primero que hace la máquina será traducir el nombre `thales.cica.es` a su dirección IP que es con la que trabajan las tarjetas de red. Después mandará los paquetes a la dirección indicada, a través del router si no estamos en la misma red.

Para un sistema RedHat, la configuración que hemos hecho se guardaría en el directorio `/etc/sysconfig/`, que contiene los ficheros que leerá el sistema al arrancar y activar la red. El fichero `/etc/sysconfig/network` contendrá algo parecido a esto:

```
NETWORKING=yes
HOSTNAME="cursolinux"
FORWARD_IPV4="no"
GATEWAYDEV="eth0"
GATEWAY="192.168.1.1"
```

El fichero `/etc/sysconfig/network-scripts/ifcfg-eth0` contendrá la configuración para la tarjeta de red (`eth0`):

```
DEVICE=eth0
BOOTPROTO=static
IPADDR=192.168.1.5
NETMASK=255.255.255.0
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
NETWORK=192.168.1.0
BROADCAST=192.168.1.255
PEERDNS=no
GATEWAY=192.168.1.1
```

Otra utilidad de diagnóstico es el comando:

**traceroute** que nos permite ver por los sitios por los que van pasando los paquetes en el camino hasta su dirección de destino.

Por ejemplo,



```

$ /usr/sbin/traceroute metalab.unc.edu
traceroute to metalab.unc.edu (152.19.254.81), 30 hops max, 38 byte packets
 1 192.168.3.254 (192.168.3.254) 1.197 ms 1.085 ms 1.050 ms
 2 192.168.254.5 (192.168.254.5) 45.165 ms 45.314 ms 45.164 ms
 3 obsgate (192.168.2.254) 48.205 ms 48.170 ms 48.074 ms
 4 obsposix (160.124.182.254) 46.117 ms 46.064 ms 45.999 ms
 5 cismphjb.posix.co.za (160.124.255.193) 451.886 ms 71.549 ms 173.321 ms
 6 cisapl.posix.co.za (160.124.112.1) 274.834 ms 147.251 ms 400.654 ms
 7 saix.posix.co.za (160.124.255.6) 187.402 ms 325.030 ms 628.576 ms
 8 ndf-core1.gt.saix.net (196.25.253.1) 252.558 ms 186.256 ms 255.805 ms
 9 ny-core.saix.net (196.25.0.238) 497.273 ms 454.531 ms 639.795 ms
10 bordercore6-serial5-0-0-26.WestOrange.cw.net (166.48.144.105) 595.755 ms 595.174
ms *
11 corerouter1.WestOrange.cw.net (204.70.9.138) 490.845 ms 698.483 ms 1029.369
ms
12 core6.Washington.cw.net (204.70.4.113) 580.971 ms 893.481 ms 730.608 ms
13 204.70.10.182 (204.70.10.182) 644.070 ms 726.363 ms 639.942 ms
14 mae-brdr-01.inet.qwest.net (205.171.4.201) 767.783 ms * *
15 * * *
16 * wdc-core-03.inet.qwest.net (205.171.24.69) 779.546 ms 898.371 ms
17 atl-core-02.inet.qwest.net (205.171.5.243) 894.553 ms 689.472 ms *
18 atl-edge-05.inet.qwest.net (205.171.21.54) 735.810 ms 784.461 ms 789.592 ms
19 * * *
20 * * unc-gw.ncnren.net (128.109.190.2) 889.257 ms
21 unc-gw.ncnren.net (128.109.190.2) 646.569 ms 780.000 ms *
22 * helios.oit.unc.edu (152.2.22.3) 600.558 ms 839.135 ms

```

Nos indica las redes y routers por los que atraviesan los paquetes desde la máquina en la que se ha lanzado el comando hasta la máquina metalab.unc.edu.

Otra utilidad es el comando `tcpdump` que escucha en una interfaz de red todo el tráfico que pasa por ese segmento de red.

## 27.5. Configuración: servidores y servicios de red.

Los servidores de red y los servicios son aquellos programas que permiten a un usuario remoto hacer uso de nuestra máquina Linux. Los programas servidores escuchan en los puertos de red. A cada aplicación (FTP o Telnet, por ejemplo) se le asigna un único número, a modo de casillero, llamado PUERTO. Cuando se produce una solicitud de conexión a dicho puerto, se ejecutará la aplicación correspondiente. Los puertos de red son el medio de llegar a un servicio determinado en una máquina en concreto, y es así como un servidor conoce la diferencia entre una conexión telnet y otra de FTP que le lleguen.

Algunas asignaciones estándar de puertos son<sup>16</sup>

Servicio o Aplicación	Puerto
<i>File Transfer Protocol (FTP)</i>	21
<i>Secure SHell (ssh)</i>	22
Telnet	23
<i>Simple Mail Transfer Protocol (SMTP)</i>	25
Gopher	70
Finger	79
<i>Hypertext Transfer Protocol (HTTP)</i>	80
<i>Network News Transfer Protocol (NNTP)</i>	119
<i>Post Office Protocol (POP3)</i>	110

<sup>16</sup>Véase la sección siguiente: Fichero `/etc/services` en la página 368



Cuando establecemos una conexión con una máquina a través de la red, abrimos un *socket* (enchufe). El socket se compone de cuatro parámetros que lo identifican de forma única:

- la dirección de red de la máquina origen,
- puerto en la máquina origen,
- dirección de red de la máquina destino y
- puerto de la máquina destino.

Como prueba, utilizando el interfaz de loopback vamos a establecer una sesión con nuestra propia máquina y veamos los sockets que se han generado. Por ejemplo, hagamos<sup>17</sup>

```
$ssh localhost
```

y abramos una sesión con nuestro usuario. Si ejecutamos la orden

```
$ netstat -a|grep ssh
tcp 0 0 *:ssh *: LISTEN
tcp 0 0 localhost:ssh localhost:32776 ESTABLISHED
tcp 0 0 localhost:32776 localhost:ssh ESTABLISHED
```

La primera línea mostrada nos indica que en el puerto ssh<sup>18</sup> se encuentra un proceso a la espera de posibles conexiones (LISTEN). Las dos últimas líneas corresponden a la conexión que hemos realizado. Sale doble porque nuestro sistema la está viendo por dos partes, la del puerto de entrada y la del puerto de salida. Quieren decir que mediante el protocolo `tcp` hay una conexión establecida (ESTABLISHED) entre la máquina local `localhost` por su puerto `ssh` (corresponde al puerto número 22) y la máquina `localhost` por su puerto 32776.

Si estableciéramos otra conexión, la unicidad del socket no se vería afectada porque se cogería el siguiente puerto<sup>19</sup>, el 32777. Los dos sockets que son diferentes, podríamos verlos como<sup>20</sup>:

```
localhost:ssh ----- localhost:32776
localhost:ssh ----- localhost:32777
```

El usuario remoto establece una conexión de red (siguiendo el protocolo `ssh`) con la máquina, y el programa servidor, o demonio de red que esté escuchando en ese puerto (el 22), aceptará la conexión y se ejecutará. Hay dos modos de operación para los demonios de red. Éstos son en modo:

**autónomo (standalone)** el programa demonio de red escucha en el puerto de red asignado y, cuando llega una conexión, se ocupa él mismo de dar el servicio de red. En este modo suele trabajar por ejemplo un servidor web (en nuestro caso Apache).

**esclavo del servidor xinetd (o inetd)** se trata de un súper-demonio (siempre están en ejecución) cuya finalidad es estar a la espera de que se produzca alguna solicitud de conexión del exterior. Si esto pasa

<sup>17</sup>Véase la página 399 para saber qué necesitamos para disponer de un servidor `ssh`.

<sup>18</sup>Si el puerto recibe un nombre en el fichero `/etc/services`, el sistema pone automáticamente el nombre. En este caso el número 22, si vemos en el fichero `/etc/services`, corresponde al servicio `ssh`.

<sup>19</sup>Los puertos para conexiones externas se cogen a partir del 1024 y consecutivamente hasta el 65535, en donde se vuelve a empezar por el 1024 y saltando si existiera alguno ocupado.

<sup>20</sup>Recordemos que un socket era una dirección de origen, un puerto de origen, una dirección de destino y un puerto de destino.



`xinetd` analiza esa solicitud determinando qué servicio le están solicitando y le pasa el control a dicho servicio<sup>21</sup>.

A partir de la versión 7.0, Red Hat utiliza el sistema `xinetd`, que es una mejora del servidor `inetd`. Para cada servicio tiene un fichero de configuración que le dice qué programa debe ser ejecutado cuando se reciba una conexión. Cualquier puerto de servicio puede ser configurado tanto para el protocolo `tcp` como para `udp`<sup>22</sup>. El servicio de `telnet` o el de `pop3` suelen trabajar en este modo.

`xinetd` se escribió con la idea de superar las desventajas que presentaba el demonio `inetd`:

- No necesita del fichero `/etc/services`
- Permite matar los servicios modificados/borrados del directorio de configuración
- Es más configurable en cuanto a las opciones de registro

### Fichero `/etc/services`

El fichero `/etc/services` es una base de datos sencilla, que asocia un nombre que nosotros podamos entender, con un puerto de servicio de la máquina. Su formato es bastante simple. Es un fichero de texto en el que cada línea representa una entrada a la base de datos. Cada entrada comprende tres campos separados por cualquier número de espacios en blanco (espacio o tabulador).

Los campos son:

`nombre puerto/protocolo sobrenombres # comentario`

**nombre** una sola palabra que representa el servicio descrito.

**puerto/protocolo** este campo está dividido en dos subcampos.

**puerto** un número que especifica el número de puerto del servicio que estará disponible. La mayoría de los servicios comunes tienen asignados números de servicio, y están descritos en el RFC-1340.

**protocolo** este subcampo debe tener como valor `tcp` o `udp`. El protocolo `tcp` está orientado a conexión (podemos asimilarlo a una llamada telefónica, en la que las dos partes deben estar activas en el mismo instante). El protocolo `udp` es no orientado a la conexión (sería como un servicio de correo postal, mandamos una carta y nos despreocupamos, que ya llegará a su destino).

Para un puerto puede existir un servicio `tcp`, uno `udp`, ambos o ninguno.

**sobrenombres (o alias)** otros nombres que pueden usarse para referirse a esta entrada de servicio.

**#** Cualquier texto que aparezca en una línea después de un carácter `#` es ignorado y se trata como un comentario.

Un ejemplo de líneas en el fichero `/etc/services`.

```
telnet 23/tcp
ssh    22/tcp
ssh    22/udp
smtp   25/tcp    mail #servicio de correo
```

Asociamos el servicio `telnet` al puerto 23 mediante el protocolo `tcp`. El servicio `ssh` (sustituto del `telnet`, `ftp` y otros porque aporta cifrado de las comunicaciones) lo asigna al puerto 22, tanto por `tcp` como por `udp`. Para el servicio `smtp` asociamos el puerto 25, por protocolo `tcp` y le asignamos un alias `mail`.

<sup>21</sup>Su funcionamiento sería el siguiente. Escucha en una serie de puertos para los que se ha configurado. En caso de recibir una petición por uno de ellos, sabe a qué programa debe llamar. Si es el 23, llama al servidor de `telnet`; si es al 22, llama al servidor de `ssh`. Por ejemplo, si la máquina remota solicita una transferencia de fichero por el puerto 21, le pasará la solicitud a `wu-ftp` (proceso del servidor de `ftp`)

<sup>22</sup>Un poco más adelante se ejemplifica qué diferencias hay entre ambos protocolos.



### 27.5.1. El servicio xinetd

El servicio `xinetd` es una mejora (o evolución) del `inetd`. Se inicia al arrancar el sistema y permanece a la escucha pendiente de si se solicita alguno de los servicios que gestiona desde una máquina remota. Tiene un fichero de configuración general (`/etc/xinetd.conf`) y ficheros de configuración para cada uno de los servicios que lanza, que se encuentran en el directorio `/etc/xinetd.d`.

Para ampliar sobre su uso: Capítulo 8 del *Manual Oficial de Referencia de Red Hat Linux*<sup>23</sup>.

El fichero `/etc/xinetd.conf` tiene lo siguiente por defecto:

```
defaults
{
    instances                = 60
    log_type                  = SYSLOG authpriv
    log_on_success            = HOST PID
    log_on_failure            = HOST RECORD
    cps                       = 25 30
}
```

`includedir /etc/xinetd.d`

Nos dice que los valores por defecto son:

- arrancar como máximo 60 instancias de un servicio determinado
- el tipo de log (archivo determinado en `/etc/syslog.conf` con `authpriv: /var/log/secure`) que genera y da valores en caso de que la autenticación del usuario haya sido satisfactoria (IP de la máquina remota e ID de proceso), y en caso de fallo en la autorización (IP de la máquina remota e información sobre el sistema remoto).
- La línea `cps` nos dice que se rearrancarán un máximo de 25 procesos por segundo. Si se supera ese valor, se deshabilitará ese servicio durante 30 segundos.
- Además nos dice que el resto de ficheros los encontraremos en el directorio `/etc/xinetd.d`

El formato para cada uno de los ficheros que permiten configurar los distintos servicios, es de la forma:

```
service "nombre_servicio"
{
    ...
    atributo = valor
    serie_valores -= elimina_valor
    serie_valores += añade_valor
    ...
}
```

Lo normal es que sólo se use "=" para asignar una valor a un atributo. Si el atributo es una serie de valores podemos eliminar un elemento de la serie con "-=" o añadirlo con "+=". Normalmente son:

**disable** toma los valores "yes" o "no".

**type** toma los valores `RPC`, `INTERNAL` (servicio que provee `xinetd`, puede ser: `echo`, `time`, `daytime`, `chargen` y `discard`) o `UNLISTED` (no aparece en `/etc/services`)

**id** nombre unívoco para identificar este servicio

**socket\_type** describe el tipo de socket que esta entrada considerará relevante. Los valores permitidos son: `stream`, `dgram`, `raw` o `seqpacket`. Por regla general casi todos los servicios basados en `tcp` usan `stream`, y casi todos los basados en `udp` usan `dgram`. Sólo algunos demonios servidores muy particulares usarán otros valores

<sup>23</sup>Disponible en <http://europe.redhat.com>.



**protocol** el protocolo considerado válido para este servicio obtenido a partir del fichero `/etc/protocols` (suele ser `tcp` o `udp`). Si no se especifica, se usa el protocolo por defecto para ese servicio.

**wait** puede ser `yes` o `no`. Con este atributo indicamos a `xinetd` si el programa servidor de red libera el socket después de comenzar la ejecución (`wait=no`), y si por tanto `xinetd` podrá ejecutar otro servidor para la siguiente petición de conexión o deberá esperar (`wait=yes`) y asumir que el demonio servidor que esté ejecutándose controlará las nuevas peticiones de conexión. Por norma general todos los servidores `tcp` deberían tener esta entrada con el valor `no` y la mayoría de servidores `udp` deberían tener `yes`.

**user** indica qué cuenta de usuario de `/etc/passwd` será asignada como propietario del demonio de red cuando se ejecute. Esto es a menudo útil para protegerse ante riesgos de seguridad. Se puede asignar el usuario `nobody` a una entrada, por lo que si la seguridad del servidor de red es traspasada, el posible daño queda minimizado. Habitualmente, sin embargo, este campo está asignado a `root`, porque muchos servidores requieren privilegios de administrador para funcionar correctamente.

**group** por si queremos especificar el grupo con que se ejecuta el servicio. Tiene que tener una entrada en `/etc/group`.

**instances** número máximo de peticiones que este servicio puede administrar.

**server** camino completo hasta el programa servidor a ejecutar para esta entrada.

**server\_args** argumentos pasados al servidor.

**no\_acces** IP de máquinas que no podrán acceder a este servicio.

Para conocer todas las posibilidades

```
$ man xinetd.conf
```

Veamos por ejemplo el servicio `telnet` y el servicio `swat` (interfaz vía web para configuración de SAMBA).

Este es el contenido del fichero `/etc/xinetd.d/telnet`

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#              unencrypted username/password pairs for authentication.
service telnet
{
    disable           = no
    flags             = REUSE
    socket_type       = stream
    wait              = no
    user              = root
    server             = /usr/sbin/in.telnetd
    log_on_failure    += USERID
}
```

La primera línea (`service telnet`) nos dice el servicio que estamos configurando y en caso de no encontrar un puerto más adelante, cogerá el que tenga este nombre del fichero `/etc/services`.

Con `disable=no`, decimos que está habilitado (listo para funcionar).

Vemos que es del tipo `stream` y que no tiene por qué haber finalizado la ejecución anterior para lanzar otro servicio nuevo (`wait=no`). El usuario que lo ejecuta es `root` (`user=root`) y el programa que se lanza es `/usr/sbin/in.telnetd`. Por último, si no hemos podido identificarnos correctamente, guarda el identificador de usuario.

El del servicio `swat`, será el siguiente.

```
# default: off
```



```
# description: SWAT is the Samba Web Admin Tool. Use swat \
#               to configure your Samba server. To use SWAT, \
#               connect to port 901 with your favorite web browser.
service swat
{
    disable = yes
    port    = 901
    socket_type = stream
    wait    = no
    only_from = localhost
    user    = root
    server  = /usr/sbin/swat
    log_on_failure += USERID
}
```

En este servicio vemos que por defecto está deshabilitado. En caso de querer activarlo, debemos poner `disable=no` y decirle a `xinetd` que vuelva a releer su configuración. En este caso ejecutando:

```
# /etc/rc.d/init.d/xinetd reload
```

El control sobre los accesos que veremos después con el complemento `tcpwrappers`, puede ser realizado directamente ahora mediante las opciones `only_from` (para indicar direcciones o nombres de host o dominios desde los que se puede acceder al servicio) y `no_access` (para excluir direcciones o nombres y desde los cuales no se podrá acceder). Por ejemplo, una entrada para el servicio `telnet` anterior modificada así:

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#               unencrypted username/password pairs for authentication.
service telnet
{
    disable          = yes
    flags            = REUSE
    socket_type      = stream
    wait            = no
    user             = root
    server           = /usr/sbin/in.telnetd
    log_on_failure   += USERID
    only_from        = 192.168.0.0/16
    no_access         = 192.168.1.0/24
    acces_time       = 08:30-14:45
}
```

permite el acceso sólo a las máquinas de la red local de clase B del atributo `only_from`, impide que se conecten las máquinas de la subred de clase C 192.168.1.0 y además limita la posibilidad de conexión vía `telnet` al horario establecido en el atributo `acces_time`. Pero que ... ¡si este servicio está desactivado!

### 27.5.2. El mecanismo de control de acceso `tcpd` (`tcp_wrappers`)

Además de las posibilidades de control y registro de `xinetd`, podemos usar el programa `tcpd` (del paquete `tcp_wrappers`). Este demonio proporciona mecanismos de control de registro y acceso a los servicios que haya que proteger. Con este sistema, es él el encargado de controlar el acceso y el registro. Si todo ha ido como debe, ejecuta el servicio en cuestión y después se “quita de en medio”. Cuando es invocado lee dos ficheros que contienen reglas de acceso y que permiten o deniegan el acceso al servidor que está protegiendo.





Mirá en los ficheros de reglas hasta que encuentre la primera correspondencia. Si no se encuentran correspondencias asume que el acceso debería estar permitido para todo el mundo. La secuencia de archivos que revisa es:

```
/etc/hosts.allow,  
/etc/hosts.deny.
```

Si una línea del fichero `/etc/hosts.allow` coincide con el intento de conexión, éste es autorizado. Si no hay ninguna coincidencia en este fichero se mira en `/etc/hosts.deny`. Si hay alguna entrada que impida la conexión, la deniega, pero si no lo impide ninguna entrada, el servicio es autorizado.

Para una descripción completa de este servicio deberían mirarse a las páginas del manual apropiadas (`hosts_access(5)` es un buen punto de partida).

**Fichero `/etc/hosts.allow`** El `/etc/hosts.allow` es un fichero de configuración del programa `/usr/sbin/tcpd`. El fichero `hosts.allow` contiene reglas que describen qué máquinas tienen permiso para acceder a un servicio de nuestra máquina.

El formato del fichero es bastante sencillo:

```
<lista de servicios>: <lista de hosts>[: orden]
```

**lista de servicios** es una lista delimitada por espacios en blanco y/o comas, de nombres de servidores a los que se aplica esta regla. Ejemplos de nombre de servicio son: `ftpd`, `telnetd` y `fingerd`.

**lista de hosts** es una lista de nombres de máquinas, delimitada por espacios en blanco y/o comas. Aquí también podemos usar direcciones IP. De forma adicional, se pueden especificar nombres de máquinas o direcciones usando caracteres comodín para corresponder con grupos de máquinas. Por ejemplo: `ciencias.iesmurgi.org` para una máquina específica, `.uts.edu.au` para cualquier nombre de máquina que acabe en esa cadena, `80.32.` para cualquier dirección IP que comience con esos dígitos. Hay algunas palabras especiales para simplificar la configuración, algunas de las cuales son:

**ALL** que se corresponde con cualquier host.

**LOCAL** se corresponde con cualquier nombre de host que no contenga un `.` o sea que esté en el mismo dominio que nuestra máquina;

**KNOWN** se corresponde con cualquier máquina/usuario de IP/nombre conocido

**UNKNOWN** se corresponde con cualquier máquina/usuario de IP/nombre desconocido

**PARANOID** se corresponde con cualquier nombre que no se corresponda con su dirección IP (por si se intenta camuflar la identidad real de la máquina que solicita la conexión). Hay una última palabra que también es útil. La palabra

**EXCEPT** permite proporcionar una lista con excepciones.

**orden** es un parámetro opcional. Este parámetro es el camino completo hasta una orden que debería ser ejecutada cada vez que se cumpla esta regla. Podría por ejemplo ejecutar una instrucción que intentase identificar quién está autenticado en el host que conecta, o generar un mensaje de correo u otro tipo de alerta a un administrador de sistema avisando de que alguien intenta conectar. Hay cierto número de expansiones que podríamos incluir, algunos ejemplos comunes son:

**%h** se expande al nombre de la máquina que se conecta o a su dirección si no tiene un nombre,

**%d** es el demonio que está siendo llamado.

Un ejemplo:

```
# /etc/hosts.allow  
#
```



```
# Permitir correo de salida a todo el mundo
in.smtpd: ALL
# telnet y FTP sólo a hosts dentro de mi dominio y al host de
# thales
telnetd, ftpd: LOCAL, thales.cica.es
# Permitir finger a cualquiera pero mantener un registro de quién es.

fingerd: ALL: (finger %@h | mail -s "finger desde%h" root)
```

**Fichero /etc/hosts.deny** El fichero `hosts.deny` contiene reglas que describen qué máquinas tienen prohibido el acceso a un servicio en su máquina.

Un ejemplo simple podría parecerse a esto:


```
# /etc/hosts.deny
#
# Desautorizar a todos los host con nombre sospechoso
ALL: PARANOID
#
# Desautorizar a todos los host.
ALL: ALL
```

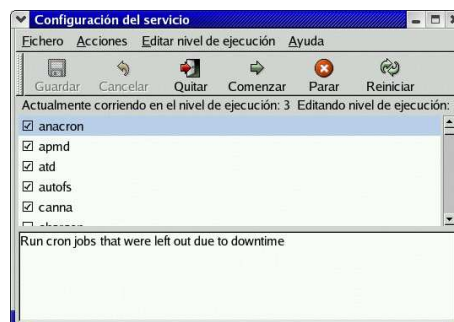
La entrada `PARANOID` es redundante porque la otra entrada abarca todo en cualquier caso. Ambas entradas serían razonables por defecto dependiendo de sus requisitos particulares.

La configuración más segura es tener `ALL: ALL` por defecto en `/etc/hosts.deny` para después dar permiso específicamente a aquellos servicios y hosts que se desee en `/etc/hosts.allow`.

### 27.5.3. Control de acceso a servicios

Antes de entrar de lleno en cómo instalar y configurar determinados servicios de red, vamos a recordar las herramientas de que disponemos en RedHat para poder activar o desactivar determinados servicios según los distintos niveles de ejecución, se trata de:

# **redhat-config-services** (desde Gnome:  → **Configuración de servidores** → **Servicio**), es una utilidad gráfica que permite seleccionar qué servicios están en activo para los niveles 3, 4 y 5. Permite reiniciarlos, pararlos y activarlos desde el propio programa.

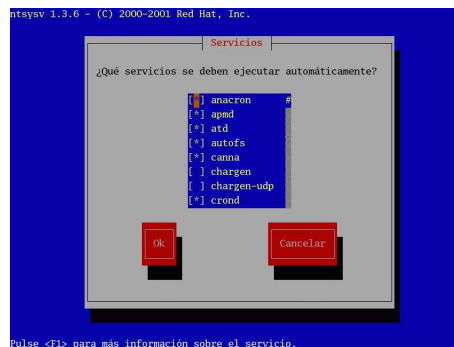


También se puede acceder al programa escribiendo el comando `serviceconf`.

# **/usr/sbin/ntsysv** (se puede acceder a él desde el programa `setup`, opción **System services**). Los cambios no se activan en el momento, además afecta sólo al nivel desde el que se ejecuta.<sup>24</sup>

<sup>24</sup>Para conseguir que "afecte" a los niveles 3 y 5 por ejemplo usar:

```
# ntsysv --levels 35
```



# **/sbin/chkconfig** utilidad en línea de comandos para activar o desactivar servicios. En general es más fácil trabajar con las dos anteriores.

### Algunos servicios de red usuales

<b>autofs</b> activa el proceso de administración de montaje automático de sistemas de ficheros o unidades (NFS, CD...)	<b>network</b> para activar interfaces de red de nuestra máquina.
<b>bootparamd</b> para poder usar un servidor con estaciones trabajo sin disco	<b>nfs</b> activa servicios NFS
<b>dhcpcd</b> inicia un servidor DHCP local que permite asignar direcciones IP de forma dinámica.	<b>portmap</b> Este demonio administra conexiones a servicios basados en RPC.
<b>httpd</b> activa el servidor web Apache	<b>postgresql</b> activa el servidor de bases de datos postgresql.
<b>innd</b> habilita un servidor de noticias local.	<b>sendmail</b> activa el Agente de transporte de correo (MTA) sendmail
<b>ipchains</b> permite usar nuestra máquina como pasarela, dotándola de cortafuegos (núcleos 2.2)	<b>smb</b> activa el servicio Samba (para compartir archivos e impresoras con redes Windows)
<b>iptables</b> igual que ipchains pero para los núcleos 2.4	<b>squid</b> permite disponer del proxy HTTP squid
<b>linuxconf</b> utilidad que permite configurar múltiples aspectos de nuestra máquina	<b>sshd</b> habilita servicios de red seguros (Secure SHell)
<b>lpd</b> servidor de impresión.	<b>syslog</b> demonio para registrar los log (o archivos de auditoría y trazas) del sistema
<b>mysqld</b> para disponer del servidor de bases de datos MySQL	<b>wu-ftp</b> activa los servicios ftp
<b>netfs</b> activo permite montar sistemas de archivos de red: NFS, Samba y NetWare.	<b>xfs</b> servidor de fuentes para las X
	<b>xinetd</b> permite activar múltiples servicios de red

Además de los programas anteriores, si tenemos un servicio en nuestra máquina, con:

```
$ /etc/rc.d/init.d/service
```

podemos comprobar qué parámetros admite. Por ejemplo, con el servidor de impresión obtendríamos:

```
$ /etc/rc.d/init.d/lpd
```

```
Uso: /etc/rc.d/init.d/lpd {start|stop|restart|condrestart|reload|status}
```

O sea que si queremos pararlo sólo hay que ejecutar:

```
# /etc/rc.d/init.d/lpd stop
```

```
Parando lpd: [ OK ]
```

Y si queremos arrancarlo, hay que ejecutar:

```
# /etc/rc.d/init.d/lpd start
```

```
Arrancando lpd: [ OK ]
```

## 27.6. Servidor Web (apache).

Apache es el servidor Web (protocolo HTTP) más utilizado en el mundo actualmente. Se encuentra muy por encima de sus competidores, ya sean gratuitos o comerciales. Por supuesto, es el más utilizado en sistemas Linux. En su forma más simple, un servidor web transmite páginas en formato HTML a los navegadores cliente (Netscape, Explorer, Opera...). Pero el servidor web, hoy día puede hacer mucho más, ya sea por sus propios medios o mediante su integración con otros programas.

Existen varias formas en las que Apache puede proveernos contenido:

- **Páginas estáticas** Es el modo básico y más primitivo, pero que en un gran número de casos es lo único que se necesita: transferir ficheros HTML, imágenes... Puede que con un servidor Linux de bajas prestaciones (puede ser un 486) consigamos estupendos resultados, si es sólo esto lo que queremos.
- **Contenido dinámico** La información cambia constantemente y un medio para mantener nuestras páginas actualizadas es generarlas dinámicamente desde una base de datos, ficheros u otras fuentes de datos.

Apache posee muchas facilidades para generar este tipo de contenido.

1. **Scripts CGI.** CGI viene de common gateway interface. Los scripts CGI son programas externos que se llaman desde el propio servidor cuando una página lo requiere. El CGI recibe información del servidor web y genera como salida una página web dinámica para el cliente. El script puede realizarse en cualquier lenguaje de programación siempre que siga las reglas del interfaz CGI. El problema es que es un proceso lento, al tenerse que lanzar un proceso externo al servidor web por cada petición. Perl es uno de los lenguajes más utilizados para ello.
2. **Intérpretes incluidos en Apache.** Tienen la ventaja sobre los cgi de que están incluidos en el propio Apache y no hay que lanzar un nuevo proceso por cada petición. Los módulos más utilizados son PHP y mod\_perl.
3. **Servlets en Java.** Es una opción que se utiliza en los servidores de aplicaciones, por ejemplo el de Oracle y su gran ventaja sería la portabilidad y escalabilidad.

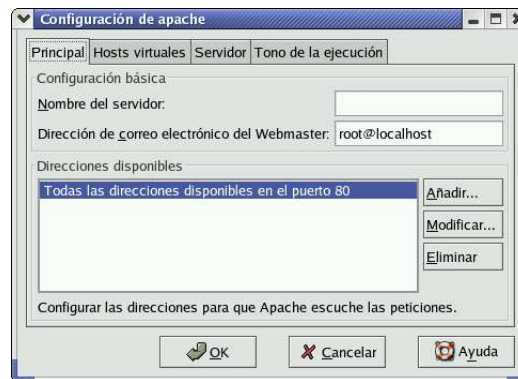
Montar un servidor Web en nuestro Linux es muy sencillo. El servidor web que viene en la distribución es el Apache y funciona como servidor independiente (no lo activa `xinetd`) y escucha por defecto en el puerto 80. Nos aseguramos de que tengamos el paquete apache instalado en nuestra máquina.

```
$rpm -q -a |grep httpd
httpd-2.0.40-8
httpd-manual-2.0.40-8
httpd-devel-2.0.40-8
```

Estos serían el paquete básico de apache, el manual y el paquete de desarrollo (sólo el primero es imprescindible). En caso de que no esté instalado lo instalamos desde el CD2 los dos primeros y del CD3 el último.



Hay un cuarto paquete que nos puede facilitar enormemente la configuración de Apache. Se trata de `redhat-config-httpd-1.0.1-13.noarch.rpm` (CD2), es una herramienta gráfica para la configuración de Apache.



Si se desea usar se puede consultar el capítulo 18 del *Manual oficial de personalización de Red Hat Linux*. Una nota a tener en cuenta y que aparece en en él:

#### “Aviso

No modifique el fichero de configuración de Apache `/etc/httpd/conf/httpd.conf` si desea utilizar esta herramienta. Dicha herramienta crea este fichero después de que haya grabado los cambios y haya salido del programa. Si desea añadir módulos u opciones que no se encuentren en la herramienta no podrá usarla.”

En el momento del arranque, si existe el enlace `/etc/rc.d/rc3.d/S85httpd` se arrancará de forma automática. Si no existe, podemos crearlo con cualquiera de la herramientas comentadas: `serviceconf`, `ntsysv` o `chkconfig`.

Si queremos activarlo de forma manual una vez, podemos ejecutar

```
#/etc/rc.d/init.d/httpd start
o
#service httpd start
```

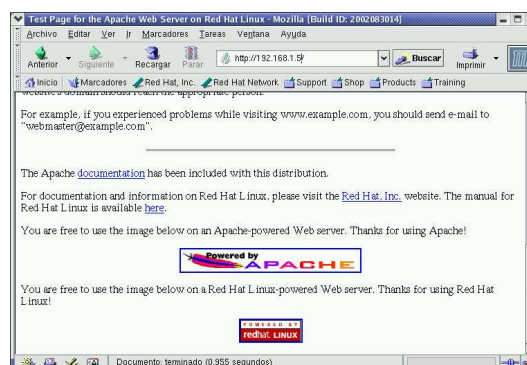
y si lo queremos parar

```
#/etc/rc.d/init.d/httpd stop
```

Para comprobar que funciona, podemos apuntar nuestro navegador preferido a la dirección

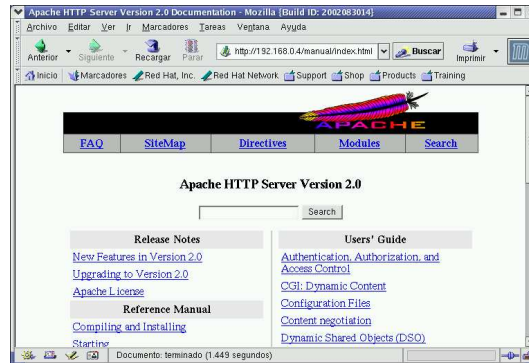
`http://192.168.1.525`

Si conseguimos una pantalla de bienvenida del servidor apache (It worked!), ya hemos contactado con nuestro servidor.



<sup>25</sup> Si es ésta nuestra dirección, o con `http://localhost` y nos serviría incluso si no tenemos tarjeta de red ni configuración de red.

Si hemos instalado el paquete del manual podremos acceder a la extensa documentación que acompaña al programa desde esta misma página:



Además podemos saber más sobre su uso en:

- Capítulos 18 y 19 del *Manual oficial de personalización de Red Hat Linux*.
- Capítulo 14 del *Manual oficial de referencia de Red Hat Linux*.

El siguiente paso es poner nuestras propias páginas en el servidor, en vez de las de bienvenida de apache. Sin más que ponerlas en el directorio

`/var/www/html`<sup>26</sup>

y empezando con la página `index.html`, podremos ver nuestras propias páginas.

Los ficheros de configuración se encuentran en el directorio `/etc/httpd/conf`, por si fuera necesario modificar algo. Si no es porque realmente lo necesitemos, con la configuración que viene por defecto podemos trabajar de forma satisfactoria. El fichero de configuración se llama `httpd.conf`<sup>27</sup>

Explicaremos algunas opciones de este fichero. El archivo `httpd.conf` está bien comentado y es bastante autoexplicativo. La configuración predeterminada funciona para los ordenadores de la mayoría de los usuarios, así que probablemente no necesitará cambiar ninguna de las directivas en el fichero `httpd.conf`. Sin embargo, es bueno conocer las opciones de configuración más importantes.

Antes de modificar el fichero `httpd.conf` es bueno copiar el fichero original dándole por ejemplo, el nombre `httpd.conf.ori`, `httpd.conf.20030519` u otro que nos sea significativo<sup>28</sup>. Si cometemos un error mientras estamos modificando el fichero de configuración, no debemos preocuparnos porque siempre dispondremos de una copia de seguridad.

Si cometemos un error y nuestro servidor de web no funciona correctamente, el primer sitio donde acudir es a lo que acabamos de modificar en `httpd.conf`. Después podemos consultar el fichero de conexión de error (`/var/log/httpd/error_log`), las últimas entradas deberían servirnos de ayuda para saber lo que ha pasado.

A continuación se dan breves descripciones de las directivas incluídas en el fichero `httpd.conf`, ordenadas según se encuentran en él.

**ServerType:** El comando `ServerType` puede ser tanto `inetd` como `standalone`. El servidor de web tiene como comando predeterminado el `ServerType standalone`, que es el recomendado, que funciona como servidor independiente.

**ServerRoot:** El comando `ServerRoot` es el directorio principal donde se encuentran todos los ficheros de configuración y trabajo del servidor. Su valor es `/etc/httpd`.

<sup>26</sup>En las versiones anteriores estaban en `/home/httpd/html`

<sup>27</sup>Antes, Apache utilizaba además `srn.conf` y `access.conf`, pero en las versiones actuales, el contenido de ambos se ha incluido en `httpd.conf`

<sup>28</sup>El objetivo es poder volver al estado original en caso de que algo vaya mal.

**User:** La directiva `User` establece el `userid` usado por el servidor para responder a peticiones. El valor de `User` determina el acceso al servidor. Cualquier fichero al que no pueda acceder este usuario será también inaccesible al visitante de la web. El comando predeterminado para `User` es `apache`.

`User` debería tener privilegios de tal manera que sólo pudiera acceder a ficheros que se supone que todo el mundo puede ver. El comando `User` también es dueño de cualquier proceso CGI que arranque el servidor. Al comando `User` no se le debería permitir ejecutar ningún código que no esté pensado para responder peticiones HTTP.

El proceso `httpd` padre se ejecuta como `root` durante operaciones normales, pero pasa al usuario `apache` inmediatamente. El servidor debe arrancar como `root` porque necesita un puerto por debajo de 1024 (el puerto por defecto es el 80). Los puertos por debajo de 1024 están reservados para el sistema, así que sólo se pueden usar si se es `root`. Una vez que el servidor se ha conectado al puerto, pasa el proceso a `User` antes de aceptar peticiones.

**Group:** El comando `Group` es similar a `User`. `Group` establece el grupo en el que el servidor responde a las peticiones. El valor predeterminado del comando `Group` también es `apache`.

**DocumentRoot:** `DocumentRoot` es el directorio que contiene la mayoría de los archivos HTML que se entregarán en respuesta a peticiones. El directorio predeterminado `DocumentRoot` es `/var/www/html`. Por ejemplo, el servidor puede recibir una petición para el siguiente documento:

```
http://localhost/prueba.html
```

El servidor buscará el fichero en el siguiente directorio por defecto:

```
/var/www/html/prueba.html
```

**Directory:** Las etiquetas `<Directory /path/a/directorio>` y `</Directory>` se usan para agrupar directivas de configuración que sólo se aplican a ese directorio y sus subdirectorios. Cualquier directiva aplicable a un directorio puede usarse en las etiquetas `<Directory>`. Las etiquetas `<File>` pueden aplicarse de la misma forma a un fichero específico.

Por defecto, se aplican parámetros muy restrictivos al directorio raíz (`/`). Con esta configuración, cualquier directorio del sistema que necesite valores más permisivos ha de ser configurado explícitamente.

El directorio `cgi-bin` está configurado para permitir la ejecución de scripts CGI, con la opción `ExecCGI`. Si se necesita ejecutar un script CGI en cualquier otro directorio, habrá que configurar `ExecCGI` para ese directorio. Por ejemplo, si `cgi-bin` es `/var/www/cgi-bin`, pero se quiere ejecutar scripts CGI desde `/home/usuario/cgi-bin`, añadirá una directiva `ExecCGI` a un par de directivas `Directory` como las siguientes al fichero `httpd.conf`:

```
<Directory /home/usuario/cgi-bin>
Options +ExecCGI
</Directory>
```

Para permitir la ejecución de scripts CGI en `/home/usuario/cgi-bin`, habrá que llevar a cabo pasos extra aparte de configurar `ExecCGI`. El valor de los permisos para scripts CGI y el recorrido entero a los scripts, debe ser de 0755. Además, el dueño del script y del directorio deben ser el mismo.

**UserDir:** `UserDir` es el nombre del subdirectorio dentro del directorio de cada usuario donde estarán los archivos HTML que serán servidos. Por defecto, el subdirectorio es `public_html`. Por ejemplo, el servidor podría recibir la siguiente petición:

```
http://localhost/~usuario/prueba.html
```

El servidor buscaría el fichero:

```
/home/usuario/public_html/prueba.html
```

En el ejemplo, `/home/usuario` es el directorio del usuario.

Hay que asegurarse que



- los permisos de los directorios de usuario sean correctos. El valor de los permisos deben ser de 0755. Los bits de lectura (r) y ejecución (x) deben estar activados en el directorio del usuario `public_html` (0755 valdrá). El valor de los permisos con que se servirán los ficheros desde `public_html` debe ser 0644 por lo menos.
- permitir el acceso usando el módulo `mod_userdir`. Con él conseguimos que Apache permita o deniege esta forma de acceso, así si deseamos activar esta posibilidad, hemos de cambiar la sección como sigue
 

```
<IfModule mod_userdir.c>
# UserDir disable
UserDir public_html
</IfModule>
```

 y releer después la configuración del servicio<sup>29</sup>.

**ErrorLog:** `ErrorLog` nombra el fichero donde se guardan los errores del servidor. Por defecto, el fichero de error del servidor es `/var/log/httpd/error_log`. El log de errores es un buen sitio para ver si el servidor genera errores y no se sabe muy bien qué pasó.

### ➡ Para practicar

1. Montar el servidor web Apache y comprobar que los usuarios del sistema pueden acceder a sus páginas web personales. Supongamos que en nuestra máquina hay un usuario de nombre `INVITADO`
  - a) Para el usuario `INVITADO` crear el directorio `$HOME/public_html`

```
$mkdir public_html
```
  - b) Poner en él un fichero html simple de nombre `index.html`, por ejemplo:
 

```
<html>
<body>
<h1>Esta es la web de Invitado</h1>
</body>
</html>
```
  - c) Modificar los permisos del `$HOME` de `INVITADO` así como del directorio `public_html` para que Apache pueda acceder a él:

```
$chmod 755 $HOME
$chmod 755 $HOME/public_html
```
  - d) Permitir que Apache acceda a directorios de usuario mediante `http://servidor_web/~usuario`, para eso cambiemos la sección del fichero de configuración del servidor como sigue
 

```
<IfModule mod_userdir.c>
# UserDir disable
UserDir public_html
</IfModule>
```

 y releer después la configuración del servicio

```
# service httpd reload
```
  - e) Comprobar que funciona apuntando con nuestro navegador a la página web `http://127.0.0.1/~invitado`
2. Crear un directorio con acceso restringido al usuario `INVITADO`
  - a) Creemos el directorio:
 

```
# mkdir /var/www/html/public
```

 y pongamos en él una página web simple (la de antes nos puede servir) de nombre `index.html`.

---

<sup>29</sup> `# service httpd reload`



- b) Creemos el directorio en donde almacenar las claves de acceso, por ejemplo:

```
# mkdir /var/www/passwd
```

Hay dos formas de trabajar con archivos de contraseñas. Si son pocos usuarios:

```
# htpasswd -c /var/www/passwd/.htpasswd invitado
```

Así creamos (-c) el archivo con el primer usuario y se nos pedirá la contraseña (hacer notar que no tiene por qué ser un usuario del sistema). Después, para añadir otros usuarios el parámetro -c no hay que ponerlo.

El fichero creado tiene de dueño al root y de permisos 600, como el servicio httpd se ejecuta como usuario apache, con esos permisos no puede leer su contenido. Para que Apache pueda leerlo hay que cambiarlo de dueño (o relajarle los permisos)

```
# chown apache /var/www/passwd/.htpasswd
```

de esta forma el servidor Web podrá leer la contraseña introducida.

- c) Creemos en /var/www/html/public el fichero .htaccess

```
# cat /var/www/html/public/.htaccess
```

```
AuthType Basic
```

```
AuthName "Pagina de Invitado"
```

```
AuthUserFile /var/www/passwd/.htpasswd
```

```
AuthGroupFile /dev/null
```

```
require user invitado
```

Comentemos un poco el fichero: con la directiva AuthType en Basic indicamos que la contraseña se negociará en texto plano. En el cuadro de verificación de contraseña veremos el texto "Página de Invitado". Por último indicamos a Apache el archivo en dónde buscar la contraseña, que el grupo no importa y que el nombre de usuario requerido es INVITADO.

- d) Modifiquemos el fichero /etc/httpd/conf/httpd.conf

Si desde nuestro navegador web intentamos cargar la página:

```
http://127.0.0.1/public/index.html
```

podremos cargarla sin problema. Esto se debe a que en el fichero

```
/etc/httpd/conf/http.conf
```

hay una sección como la que sigue (pero con menos comentarios y en inglés):

```
<Directory "/var/www/html">
```

```
# Puede ser "None", "All", o cualquier combinación de "Indexes",
```

```
# "Includes", "FollowSymLinks", "ExecCGI", o "MultiViews".
```

```
#
```

```
# Notar que "MultiViews" debe ser *explícitamente* llamado ---
```

```
"Options All"
```

```
# no se lo proporciona.
```

```
# Si se descomenta, podríamos ver el contenido de los subdirectorios para
```

```
# los que no haya fichero html de inicio y además podríamos seguir
```

```
# enlaces simbólicos (con el problema de seguridad que representa)
```

```
    # Options Indexes FollowSymLinks
```

```
#
```

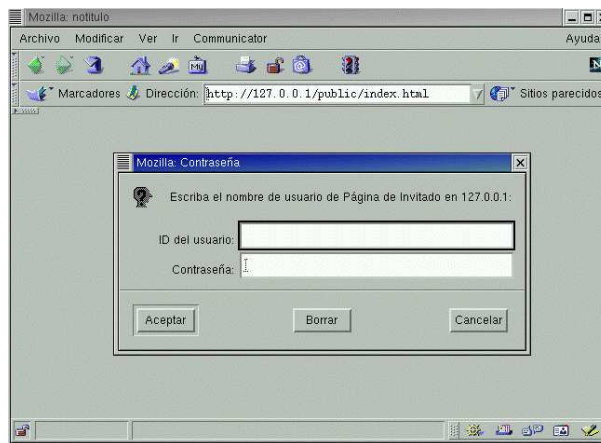
```
# Controla qué opciones pueden omitir los archivos .htaccess de los directorios
```

```
# Puede ser "All", o cualquier combinación de "Options", "FileInfo",
```

```
# "AuthConfig" y "Limit"
# En vuestro fichero estará la línea que sigue y eso implica

# que los ficheros .htaccess no pueden modificar nada.
# AllowOverride None
# Al poner esta otra permitimos que controlen la autenticación
AllowOverride AuthConfig
#
# Controla quién puede obtener la respuesta de este servidor.
# Tal cual está primero se procesan primero las directivas allow
y
# después las deny, además, la segunda línea permite el acceso
a todo el mundo
    Order allow,deny
    Allow from all
</Directory>
```

O sea que tras estudiarse esto, cambiemos la línea necesaria y debería de ir todo bien.



### 27.6.1. Servidores Seguros

Si queremos que nuestro servidor funcione con el protocolo SSL (https) para el envío de información cifrada, tenemos que instalar los paquetes `mod_ssl` y `openssl`<sup>30</sup>.

El fichero de configuración para el servidor en modo SSL se encuentra en `/etc/httpd/conf.d/ssl.conf`. Para un modo de trabajo normal, no es necesario modificarlo, solamente personalizamos por ejemplo, dónde se encuentran las páginas a mostrar (**DocumentRoot** y **Directory**).

Muy someramente, https se basa en criptografía de clave pública, en donde existen dos claves una privada y otra pública. La clave privada debe permanecer bajo el exclusivo control del propietario y la pública puede (y debe) ser conocida por todos. Un certificado digital es la clave pública, a la que se le añaden una serie de datos identificativos<sup>31</sup> y todo ello firmado por alguien en quien el resto de usuarios confían, denominado Autoridad Certificadora (CA).

Podemos solicitar un certificado a una Autoridad Certificadora, que nos llevará un dinero por ello, pero mediante `openssl` podemos erigirnos en nuestra propia Autoridad Certificadora<sup>32</sup>, transmitir la información cifrada y sin gastarnos un solo euro.

<sup>30</sup>Podemos utilizar las versiones del CD de RedHat 8, pero si hemos actualizado apache, tendremos que utilizar las versiones también actualizadas de estos paquetes

<sup>31</sup>Para un servidor pueden ser su nombre DNS y dirección IP. Para una persona física pueden ser su nombre, apellidos y DNI

<sup>32</sup>Claro está que solamente nuestros amigos y familiares confiarán en nosotros

En el directorio `/etc/httpd/conf/` aparece el directorio `ssl.key` con el fichero `server.key`. Este fichero contiene las claves pública y privada para nuestro servidor. En el directorio `ssl.crt` aparece el fichero `server.crt` que contiene el certificado (clave pública, más datos identificativos del servidor firmados por una CA). Si queremos construir nuestro propio certificado seguiremos los siguientes pasos. Nos situamos en el directorio `/etc/httpd/conf`. Modificamos el fichero `server.key`, por ejemplo con

```
#touch ssl.key/server.key
y a continuación
# make testcert
umask 77 ; \
/usr/bin/openssl genrsa -des3 1024 >/etc/httpd/conf/ssl.key/server.key
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter pass phrase:
Verifying - Enter pass phrase:
umask 77 ; \
/usr/bin/openssl req -new -key /etc/httpd/conf/ssl.key/server.key -x509
-days 365 -out /etc/httpd/conf/ssl.crt/server.crt
Enter pass phrase for /etc/httpd/conf/ssl.key/server.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:
State or Province Name (full name) [Berkshire]:
Locality Name (eg, city) [Newbury]:
Organization Name (eg, company) [My Company Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:
Email Address []:
Donde podemos poner los datos que se ajusten a nuestro servidor.
Si reiniciamos apache, tendremos disponible la conexión por el puerto 443 mediante SSL. Podemos
comprobarlo apuntando nuestro navegador a https://localhost.
```

### 27.6.2. Páginas PHP

Como ya hemos comentado, PHP es un lenguaje de script que se ejecuta en el servidor web y permite que las páginas sean dinámicas. Nos permite conexiones a bases de datos y muchas utilidades más. La comparación con ASP es inmediata, pero PHP es mucho mejor.

Necesitamos el paquete `php-4.2.2-8.0.5` (CD2), y si queremos el manual, también el paquete `php-manual-4.2.2-8.0.5` (CD3).

En el fichero `/etc/httpd/conf.d/php.conf` hay una serie de directivas que tienen que ver con su instalación, y que comentaremos. En la siguiente línea se carga el módulo de php con la orden `LoadModule`:

```
LoadModule php4_module          modules/libphp4.so
```

En esta siguiente directiva, decimos qué páginas de un directorio<sup>33</sup> pueden ser consideradas de inicio (si no se especifica una página concreta). En este caso: `index.php` se añade a la lista que ya existiera (`index.html`, `index.htm`, `index.shtml`...).

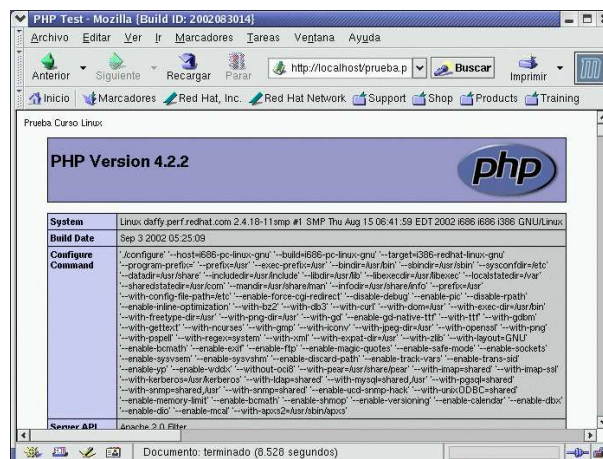
```
DirectoryIndex index.php
```

## ➔ Para practicar

1. Podemos probar que nuestro servidor con el módulo PHP funciona poniendo una página llamada `prueba.php` en el directorio `/var/www/html` con el contenido siguiente:

```
<html><head><title>PHP Test</title></head>
<body>
<?php echo "Prueba Curso Linux<P>"; ?>
<?php phpinfo(); ?>
</body></html>
```

y apuntando nuestro navegador a `http://localhost/prueba.php`



2. Un ejemplo que simplemente esboza lo que podemos hacer con PHP es el siguiente, formado por los ficheros `prueba.html` y `accion.php`

```
$cat prueba.html
<html>
<body>
<form action="accion.php" method="POST">
  Su nombre: <input type="text" name="nombre">
  Su edad: <input type="text" name="edad">
  <input type="submit">
</form>
</body>
```

<sup>33</sup> Cuando especificamos un directorio sin decir la página concreta (`http://localhost` o `http://localhost/pruebas`, siendo el primer caso el directorio raíz de los documentos y el segundo un directorio llamado `pruebas`, dentro de él)

```
<html>
```

```
$cat accion.php
```

```
Hola <?php echo $nombre?>. Tienes <?php echo $edad?>años.
```

Para poder trabajar con páginas PHP hay que modificar una directiva del fichero de configuración de php: `/etc/php.ini`. Esta directiva ha cambiado su valor por defecto en la versión de php de Red Hat 8.0 (con otras versiones de php está en ON). Se trata de hacer que `register_globals = ON`, además hay que reiniciar apache:

```
# service httpd reload
```

para que los cambios sean efectivos. Si se mantiene en OFF impedimos que se creen variables globales con los nombres de las variables pasadas como argumentos al script.

Si ponemos esos ficheros en `/var/www/html` y desde el navegador llamamos a `prueba.html` veremos el resultado de mezclar formularios y php.

## 27.7. Servicio Proxy-cache. Squid

El servidor `SQUID` nos da los servicios de proxy<sup>34</sup> y de caché<sup>35</sup>. La ventaja de todo esto es una utilización mucho más efectiva de la salida a Internet y un control de estos accesos<sup>36</sup>. Si un usuario de Internet se conecta con el Boletín Oficial del Estado a las 8 de la mañana y se descarga el BOE de ese día, los siguientes usuarios que deseen verlo, lo obtendrán a una velocidad de vértigo pues ya se encuentra en nuestro servidor.

Nos aseguramos de tener instalado el servidor squid y lo instalamos si fuese necesario (en el CD2 está `squid-2.4.STABLE7-4.i386.rpm`).

```
$rpm -q -a |grep squid  
squid-2.4.STABLE7-4
```

Si no estuviera instalado, lo instalamos a partir del paquete RPM correspondiente. Los ficheros y directorios más importantes son<sup>37</sup>:

- En el directorio `/etc/squid` se guardan los ficheros de configuración. Específicamente en el fichero `squid.conf` se encuentra la mayor parte de ella.
- Una parte importante de ficheros se encuentran en `/usr/lib/squid`, pero no tendremos que preocuparnos de ellos.
- La documentación se encuentra en `/usr/share/doc/squid-2.4.STABLE4/`
- En `/var/spool/squid` se van a encontrar las páginas “cacheadas”, es decir, las traídas desde Internet y que se almacenan para la próxima vez que las solicite alguien y no hayan cambiado.
- En `/var/log/squid` se guardan los accesos de nuestros usuarios a Internet a través del proxy, así como los posibles errores que hayan ocurrido.

<sup>34</sup>Acceder a través de él a páginas de servidores remotos, esto es, función de intermediario, sin que los clientes (navegadores que se conectan a él y están autorizados), tengan salida directa a Internet, bien por permisos o por tener direcciones IP privadas.

<sup>35</sup>Almacena las páginas de Internet en el disco duro, con lo que si estas páginas no han cambiado, nos las ofrece directamente desde este disco, sin tener que pasar a través de la red, con el consiguiente ahorro de ancho de banda de la línea y mejora en el tiempo en que la página está disponible para el cliente.

<sup>36</sup>Los ficheros de contabilidad que deja, pueden ser monitorizados para impedir accesos a Internet no deseados. Un ejemplo real es el de un organismo que manda semanalmente a los usuarios de Internet un fichero con los accesos en ese periodo. El usuario se siente controlado y es más responsable con sus accesos.

Una herramienta sencilla de configurar y muy útil para obtener estadísticas sobre las páginas visitadas es `sarg`, la podéis conseguir en formato rpm de <http://web.onda.com.br/orso/>.

<sup>37</sup>Mediante la orden `$rpm -q -l squid`, podemos ver qué ficheros componen el paquete.

Una vez instalado, revisamos el fichero de configuración `/etc/squid/squid.conf`. Es un fichero complejo, pero normalmente no tendremos que modificar muchas opciones. Comentaremos únicamente las que podemos necesitar cambiar para un uso normal.

El servidor `squid` puede o bien traer las páginas él mismo de Internet o integrarse en una jerarquía de servidores proxy. La ventaja de esto último es que nos aprovechamos de una amplia red de cachés y además entre ellos hablan un protocolo de transferencia optimizado. Se especificaría de la forma siguiente:

```
# cache_peer hostname type 3128 3130
cache_peer cache.cica.es parent 3128 3130
```

La línea comentada es el patrón que se sigue, que será la etiqueta `cache_peer`, para indicar un “compañero” de nuestro `squid`, cuyo nombre especificamos en el lugar de `hostname`, con un tipo que puede ser `parent` si dependemos de él como “hijos” o `sibling` si somos “hermanos”. Un hijo depende de los padres para todo, mientras que los hermanos cooperan al mismo nivel. Específicamente, en la segunda línea, estamos tomando a `cache.cica.es` como un padre dentro de la jerarquía (el otro tipo es el de hermano) y hablamos con él por los puertos 3128 (acceso http) y 3130 (optimizado por el Internet Cache Protocol).

Si nuestra máquina va a ir a Internet directamente a coger las páginas, no especificaremos ni hijos ni hermanos (seremos huérfanos e independientes).

- Con la etiqueta `cache_mem 8 MB` especificamos la cantidad de memoria que el proceso `squid` va a coger para trabajar en memoria. Cuanta más tengamos, mejor, pero sin llegar a agotar la memoria, pues otros procesos también necesitan “vivir”.
- La opción `cache_dir ufs /var/spool/squid 100 16 256`, nos dice que el directorio donde se guarda el caché de páginas se encuentra en `/var/spool/squid`, que es un sistema de ficheros tipo “Unix” (`ufs`) y que mantendrá 100 Megas de información como máximo. Los otros dos números controlan los índices y no necesitaremos cambiarlos normalmente.

En la sección de `Access Controls` (controles de acceso) podemos especificar quién queremos que salga a Internet a través de nosotros. Las posibilidades son muchas y para todos los gustos. Primero se define una regla de control de acceso (*Access Control List* o ACL) y luego se dan permisos para esa regla.

Como ejemplo, podemos poner sitios públicos a los que permitimos la salida a todos los usuarios:

```
acl boe dstdomain www.boe.es
...
http_access allow boe
```

En las dos líneas anteriores hemos definido una lista de control de acceso llamada `boe`, que se define como el dominio de destino (`dstdomain`) `www.boe.es`. La segunda línea, que se encuentra físicamente separada de ésta en el fichero (primero definimos las ACLs y luego les damos permisos más adelante en el fichero), dice que permitimos (`allow`) el acceso por http (`http_access`) para la regla `boe`, definida anteriormente.

La configuración más trivial y que permite acceso a todos los usuarios es:

```
acl all src 0.0.0.0/0.0.0.0
...
http_access allow all
```

Estas dos reglas se encuentran en sitios distintos del fichero. La primera define una regla de control de acceso. Su nombre es “all” y actúa sobre las direcciones ip de origen (`src`) `0.0.0.0/0.0.0.0`, es decir, todas las direcciones. La segunda, especifica que a la regla “all” se le permite (`allow`) el acceso por protocolo http.

Para permitir solamente el acceso a ciertos PCs (la red `192.168.1.0/24`), podríamos poner:

```
acl alumnos src 192.168.1.0/255.255.255.0
...
http_access allow alumnos
```

Para limitar el acceso a páginas con determinadas cadenas (cadena1, cadena2) en su nombre:

```
acl nopermitidas url_regex cadena1 cadena2 ...
```

```
...
```

```
http_access deny nopermitidas
```

La configuración que viene por defecto es denegar todos los accesos, mediante:

```
http_access deny all
```

Obviamente deberemos al menos incluir algún grupo que pueda acceder, porque si no, nuestro proxy sería innecesario.

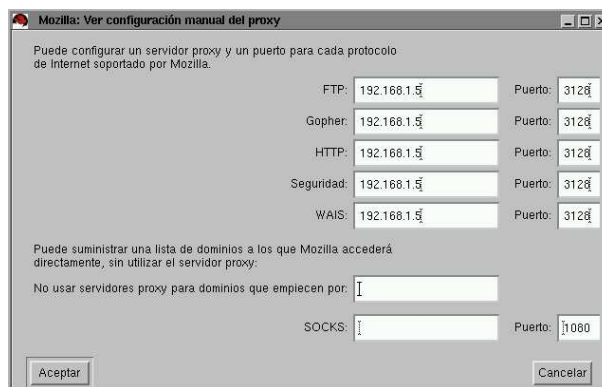
Suponiendo que ya hemos definido nuestra política de acceso, arrancamos el servicio:

```
/etc/rc.d/init.d/squid start
```

La primera vez que lo ejecutemos tardará un ratito porque tiene que construir sus índices para el almacenamiento de páginas.

El cliente lo configuraremos de la siguiente forma:

Si es Mozilla, seleccionamos **Editar**→**Preferencias**→**Avanzadas**→**Proxy**→**Configuración Manual del Proxy**, pulsamos **Ver** y en los distintos protocolos ponemos el host 192.168.1.5 por el puerto 3128 que es por el que escucha el squid peticiones de sus clientes. Podemos ponerlo en todos los protocolos menos en el socks.



Veremos que nuestros accesos a Internet son mucho más rápidos y el control que podemos llegar a tener es muy importante.

#### ➡ Para practicar

Instalar y configurar `squid` en el equipo local para que no permita el acceso a más máquinas que la nuestra y que además limite el acceso a todas las páginas en cuya URL aparezcan las cadenas: `apa` y `sex`.

Comprobar que funciona configurando `mozilla` tal cual aparece en los apuntes.

## Capítulo 28

# Más servicios de red

Cuando trabajas con Linux estás ante un sistema operativo orientado al trabajo con redes de ordenadores.

¿Qué nos empuja a poder afirmarlo tan categóricamente? Ya te darás cuenta poco a poco. (*Manual Avanzado de Linux* de RAÚL MONTERO RIVERO, Ed. Anaya)

### 28.1. Servidor de ftp.

El servidor se encuentra en el paquete `wu-ftp`<sup>1</sup>. Podemos aplicar lo visto para el servidor Web. Para un acceso con clave de usuario se nos ubicará en el directorio `home` del usuario y para un acceso anónimo (sin usuario del sistema) en el directorio `/var/ftp/`. En el directorio `/var/ftp/pub` podemos dejar archivos listos para ser recogidos a través de accesos anónimos.

Para ver cómo funciona podemos ejecutar:

```
$ftp localhost
```

Si no nos sentimos cómodos desde la línea de comandos es mejor usar los programas: `mc`, `gftp` (ya comentados).

El acceso vía ftp (y telnet) es poco seguro (las contraseñas viajan en texto claro por la red, visible por cualquiera que ponga un *sniffer*<sup>2</sup> en ella) así que si se opta por activar este servicio se debería actualizar el paquete a la última versión disponible y controlar los accesos al sistema. Si se utiliza a través de Internet está totalmente desaconsejado.

El demonio FTP se configura a través del fichero `/etc/xinetd.d/wu-ftp`<sup>3</sup> y del fichero `/etc/ftpaccess`, analicemos algunas de las entradas de este fichero:

```
# Denegar el acceso a los usuarios/grupos indicados:
# con uid/gid menor que 99 o mayor que 65534
# al resto se les permite el acceso
deny-uid%-99%65534-
deny-gid%-99%65534-
allow-uid ftp
allow-gid ftp
```

```
# Se define la clase all como aquella formada por los usuarios
# del sistema, los usuarios reales con cuentas de invitados
# y el resto para todas las IP posibles (*)
class all real,guest,anonymous *
```

<sup>1</sup>Tendremos que cargarlo si queremos disponer de un servidor ftp. `wu-ftp-2.6.2-8.i386.rpm`, está en el CD3

<sup>2</sup>Sí, porque “esnifa” paquetes de datos en la red.

<sup>3</sup>Hay que comprobar si está inactivo (`disable=yes`), en ese caso poner `disable=no` y hacer que `xinetd` relea la configuración.

```
# /etc/rc.d/init.d/xinetd reload
```



```
# Correo del administrador
email root@localhost

# Si hay 5 intentos fallidos se cierra la conexión
loginfails 5

# Si un usuario se registra o cambia de directorio se le
# avisa (si existen) de ficheros README
readme README* login
readme README* cwd=*

# Mensaje de bienvenida (hay que crearlos)
message /welcome.msg login
message .message cwd=*

# Permite que se compriman/empaqueten ficheros a todos los usuarios
compress yes all
tar yes all

# Opciones no permitidas a los usuarios que se listan

chmod no guest,anonymous
delete no anonymous
overwrite no anonymous
rename no anonymous

# Se registran en /var/log/xferlog las transferencias de los
# usuarios (separados por comas), tanto de carga como de descarga
log transfers anonymous,guest,real inbound,outbound

# Si se va a cerrar el servidor se avisa a los usuarios conectados de
ello
# según las directrices del fichero especificado (man ftpshut)

shutdown /etc/shutmsg

# Si se conecta un usuario anónimo (anonymous) tendrá que introducir
# como contraseña una dirección de correo según esa norma
# es decir, nombre@host.dominio
passwd-check rfc822 warn

Para permitir el acceso anónimo (anonymous) hay que instalar el paquete del CD1:
anonftp-4.0-12.i386.rpm.
Sólo deberíamos instalarlo si estamos seguros de que la seguridad no se va a resentir.
```

#### ➡ Para practicar: Ftp solo más WWW

1. Se trata de dar a un usuario de la máquina la posibilidad de acceso vía ftp, pero sólo eso, es decir, que no puede ver nada más del sistema y que no pueda ejecutar una shell de comandos.
  - a) Cargar y activar un servidor de ftp (para usuarios reales y anónimos): paquetes wu-ftpd y anonftp.  
Podemos comprobar que el acceso anónimo está permitido con:  
\$ ftp localhost

```

Connected to localhost (127.0.0.1).
220 localhost.localdomain FTP server (Version wu-2.6.2-8) ready.

Name (localhost:root): anonymous
331 Guest login ok, send your complete e-mail address as password.

Password: thales@cica.es
>ls
...
>exit

```

- b) Creación de usuarios invitados (usuarios que tienen limitado su acceso a sólo su \$HOME de usuario)

```

# adduser invftp
# passwd invftp

```

- c) Crear el directorio \$HOME/WWW<sup>4</sup>

```

# mkdir /home/invftp/WWW

```

Modificar la entrada del fichero /etc/passwd para el usuario INVFTP como sigue:

```

invftp:x:510:510:./home/invftp/./WWW:/etc/ftpsolo
y añadir el fichero /etc/shells la línea
/etc/ftpsolo

```

Así conseguimos que su directorio de trabajo vía ftp sea /home/invftp/WWW, que no pueda salir de él y que no pueda ejecutar comandos de la shell.

- d) Copiar los subdirectorios contenidos en /var/ftp en el \$HOME de usuario de invftp

- e) Añadir al fichero /etc/ftpaccess la entrada:

```

guestuser invftp

```

- f) Crear un fichero de bienvenida para este usuario

```

$cat /home/invftp/WWW/welcome.msg
Hola eres fenomenal

```

2. Comprobar que todo funciona (/etc/rc.d/init.d/xinetd restart) conectando con el programa gftp sobre la máquina local y con la cuenta invftp.

3. Configuremos Apache para que lea de ese directorio:

- a) Modificar los permisos de /home/invftp para que se pueda acceder a él

```

# chmod 755 /home/invftp

```

Poner en el directorio /home/invftp/WWW un fichero html de nombre index.html (nos puede servir el de la página 379)

- b) Modificar el fichero de configuración de apache como sigue<sup>5</sup>:

```

<IfModule mod_userdir.c>
    #UserDir public_html
    UserDir /home/*/WWW
</IfModule>

```

<sup>4</sup>Mejor si está en /etc/skel

<sup>5</sup>Sería deseable modificar/configurar algo parecido a

```

<Directory /home/*/WWW>
    AllowOverride FileInfo AuthConfig Limit
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
        Order allow,deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS PROPFIND>
        Order deny,allow
        Deny from all
    </LimitExcept>
</Directory>

```

pero ésta ya es para nota y os remitimos a la documentación de Apache.

Reiniciar apache (`#/etc/rc.d/init.d/httpd restart`) y comprobar que podemos ver el texto del fichero `index.html` desde un navegador con la URL `http://127.0.0.1/~invftp`

## 28.2. Correo electrónico.

El servidor `sendmail` es el que se utiliza comúnmente como agente de transporte de correo<sup>6</sup>. Un agente de transporte habla con los otros agentes de transporte para encaminar los correos a través de Internet y en última instancia deposita los correos en las carpetas de usuario. Los agentes de usuario<sup>7</sup> son los encargados de mostrar al usuario los mensajes que se encuentran en su carpeta de correo y de mandar los correos a un agente de transporte para su envío al destinatario final. Normalmente el agente de usuario habla los protocolos `pop` o `imap` con el agente de transporte para recoger el correo que ha llegado para el usuario<sup>8</sup>. Los servicios `pop` e `imap` están controlados para la seguridad del sistema por la utilidad `tcpwrapper`, con lo que para que los PCs de nuestra red puedan acceder debemos ponerlos en el fichero `/etc/hosts.allow`

```
ipop3d: LOCAL
```

El correo que el agente de usuario manda al agente de transporte para su envío, se hace por el protocolo SMTP (Simple Mail Transport Protocol) y debemos permitir acceso a nuestros usuarios en el fichero `/etc/mail/access`. Para ello, seguimos la secuencia:

```
#cd /etc/mail
#vi access9
```

```
añadimos la línea
192.168.1 RELAY
y ejecutamos
```

```
#make
```

para que el fichero `access.db` se actualice.

Con esta configuración ya tendremos correo en forma local en nuestra red. Para enlazar este correo con Internet debemos realizar configuraciones adicionales.

`Sendmail` lee al arranque un archivo de configuración llamado `/etc/mail/sendmail.cf` y este archivo es bastante difícil de leer.

### 28.2.1. Una configuración práctica.

Vamos a montar un servidor de correo que nos va a permitir<sup>10</sup>:

- Que los usuarios de nuestra Red usen sus cuentas de correo externas desde las distintas máquinas locales.
- Disponer de correo local en nuestra Intranet

Por nuestra parte sólo vamos a dar una serie de recetas y ficheros de ejemplo que permitan disponer de correo local en nuestra máquina. Para configurar nuestro servicio de correo vamos a usar tres<sup>11</sup> programas con funciones diferenciadas:

---

<sup>6</sup>MTA o Mail Transport Agent

<sup>7</sup>MUA o Mail User Agent

<sup>8</sup>En las instalaciones de RedHat normalmente no se instala el paquete `imap`, que es el que trae los servicios de `pop` e `imap`. Debemos instalarlo si vamos a utilizarlos.

<sup>9</sup>o con `mcedit` o con nuestro editor favorito

<sup>10</sup>Para disponer de un servidor de correo de “verdad” además de disponer de una IP fija necesitamos decírselo a nuestro servidor de nombres.

<sup>11</sup>En realidad serán cuatro ya que falta el MUA (*Mail User Agent*), que por comodidad vamos a suponer que es *Ximian Evolution*.

**sendmail** es un MTA (*Mail Transport Agent*), se encarga del encaminamiento del correo entre los diversos sistemas. Es el MTA por “excelencia”.

**fetchmail** sin entrar en “profundidades” vamos a entenderlo como un programa que nos va a permitir bajarnos el correo de los distintos servidores y redistribuir nuestro correo entre los distintos usuarios del sistema. En sistemas con conexión permanente a internet no es muy útil.

**imap** paquete de RedHat que nos va a permitir disponer de los servicios de correo POP2, POP3, SPOP e IMAP. Nos va a permitir disponer de correo local.

### Instalación

Tanto **sendmail** (`sendmail-8.12.5-7.i386.rpm`) como **fetchmail** (`fetchmail-5.9.0-16.i386.rpm`) deberían estar instalados por defecto en nuestro equipo.

Solo nos falta instalar el paquete del CD2 `imap-2001a-15.i386.rpm`:

```
# rpm -ivh imap-2001a-15.i386.rpm
```

### Configuración

**sendmail** Si disponemos de un solo servidor de correo saliente tendremos que editar el fichero `/etc/mail/sendmail.cf` y modificar una línea similar a:

```
# "Smart" relay host (may be null)
DS
Si nuestro servidor de correo saliente es smtp.mileto.es escribiremos:
# "Smart" relay host (may be null)
DSsmtp.mileto.es
Para no limitar el acceso sólo a nuestra máquina:
# SMTP daemon options
#cambiamos la línea comentada por
#O DaemonPortOptions=Port=smtp,Addr=127.0.0.1, Name=MTA
O DaemonPortOptions=Port=smtp,Addr=192.168.1.5, Name=MTA
```

**fetchmail** Vamos a configurarlo como root<sup>12</sup>, el motivo de hacerlo así (no es obligatorio) es que sea éste el encargado de bajarse el correo de los distintos servidores para después distribuirlo a los distintos usuarios.

Para eso crearemos en el directorio del `/root` un fichero de nombre `.fetchmailrc` con las líneas:

```
# valores por defecto
defaults
# recoger todos los mensajes del servidor/es
fetchall
#borrarlos después de bajarlos. Si en vez de flush
#escribimos keep los mensajes no se borran del servidor.
flush
#permite caracteres de 8 bits
pass8bits
#una entrada poll por cada servidor de correo
poll servidor_de_correo_1
    #protocolo usado por el servidor. En general será POP3
    proto pop3
    #nombre de usuario en el servidor de correo
    user "usuario1"
    #contraseña en el servidor de correo
    pass "password1"
```

---

<sup>12</sup>Lo aquí expuesto es igualmente válido si en vez del root es cualquier usuario del sistema.

```
#usuario local al que dirigir el correo
to usuario_local_1

poll servidor_de_correo_2
  proto pop3
  user "usuario2"
  pass "password2"
  to usuario_local_2
#si deseamos que trabaje como demonio añadiremos esta línea
#para controlar nosotros la ejecución del programa comentar esta línea
set daemon 300
```

Veamos un fichero `.fetchmailrc` de ejemplo para dos servidores de correo y reenvío a distintos usuarios del sistema:

```
$ cat .fetchmailrc
defaults
fetchall
flush
#keep
pass8bits
poll tux.iesmurgi.org
  proto pop3
  user "pvillegas"
  pass "contraseña_1"
  to paco
poll mileto.cica.es
  proto pop3
  user "ed00linux"
  pass "contraseña_2"
  to ed00linux

set daemon 300
```

Donde lo único que se ha modificado son las contraseñas de acceso a ambos servidores de correo. Este fichero debe tener unos permisos de lectura y escritura sólo del root y nadie más, así tendremos que usar

```
# chmod 0600 /root/.fetchmailrc
```

## imap

La configuración de `imap` sólo requiere que activemos los servicios necesarios. Ejecutaremos

```
# ntsysv
```




y elegiremos los servicios que queremos activar (al menos ipop3). Si deseamos ver cómo funciona con imap o con una comunicación segura (SSL) activaríamos tanto spop, imap e imaps.


Y reiniciar xinetd:

```
#/etc/rc.d/init.d/xinetd restart
```

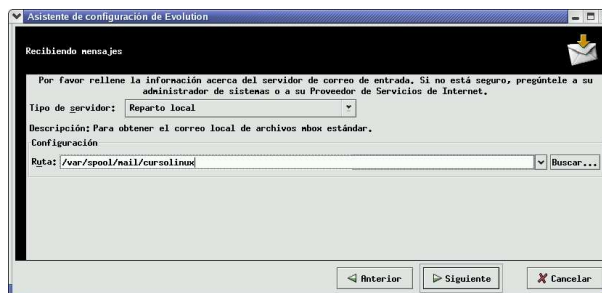
## MUA

**Máquina servidor con Ximian** Vamos a suponer que el MUA a usar es **Ximian** y que es la primera vez que lo ejecutamos, si no es así podemos modificar nuestros datos o añadir cuentas nuevas desde **Herramientas**→**Configuración del correo**. Si usamos otro, lo dicho aquí es igualmente válido sólo que ajustándolo a ese gestor de correo.

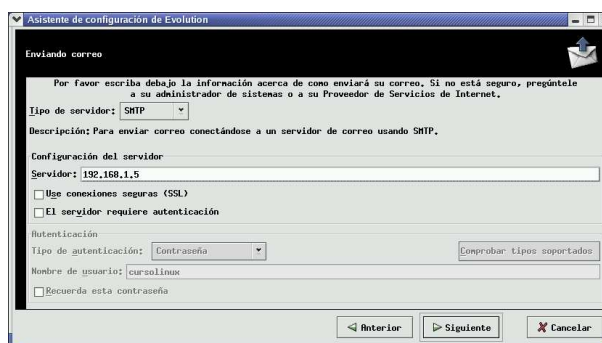
Si iniciamos el programa () por primera vez, comienza el proceso de configuración de nuestra cuenta de correo, tras la primera ventana de bienvenida, accedemos a



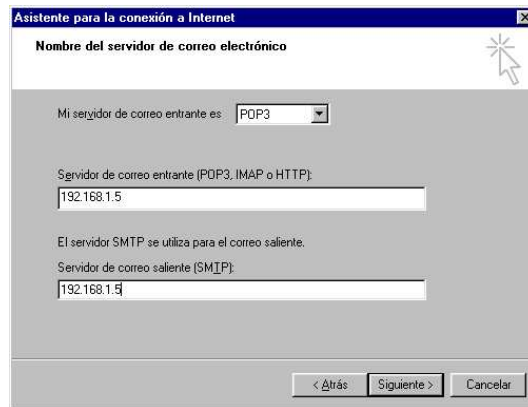
Una vez rellenos los campos mínimos necesarios pasamos a configurar el servidor de correo entrante (optaremos por Reparto local) y él sólo configurará la ruta



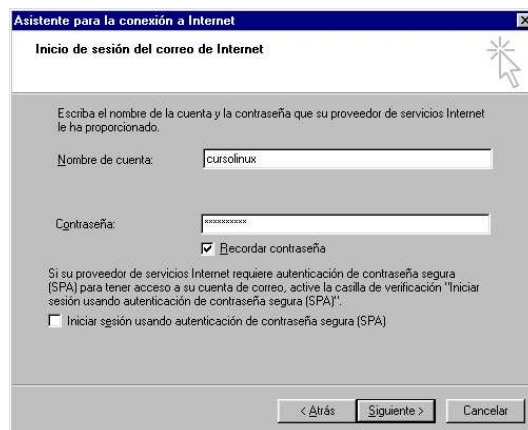
después, tendremos que optar por servidor SMTP e introducir la IP adecuada (estamos suponiendo que es 192.168.1.5)



**Máquinas clientes** Supongamos que no tenemos más remedio que trabajar con el Outlook: si es otro MUA sólo hay que adaptarlo a lo aquí expuesto. La única diferencia con el apartado anterior es que tanto en servidor de correo entrante como correo saliente hay que optar por la IP del servidor de correo local (por ejemplo 192.168.1.5).



y que en la ventana que sigue tendremos que poner el nombre de usuario y contraseñas de la máquina Linux



## RELAY

Para que se puedan enviar correos desde estas máquinas hay que permitirselo en la máquina Linux. Para aquellas IP locales a las que optemos por permitir que envíen correos añadiremos una entrada en el fichero `/etc/mail/access` del tipo:

```
IP_local          RELAY
```

por ejemplo, si la máquina tiene de IP: 192.168.1.11 quedaría:

```
192.168.1.11      RELAY
```

después de guardar, para que los cambios tengan efecto ejecutaremos en el directorio `/etc/mail` el comando

```
# make
```

para generar, a partir del fichero de texto sin formato `/etc/mail/access`, el fichero de base de datos mucho más eficiente `/etc/mail/access.db`

### 28.2.2. Poner todo en marcha.

Cada vez que nosotros enviemos un correo con Netscape, se almacenará en `/var/spool/mqueue` en espera de que nos conectemos a Internet<sup>13</sup>. Como sendmail está cargado como demonio, él sólo enviará los mensajes cuando nos conectemos. Si queremos controlar nosotros cuándo se envían, es decir, para enviarlos YA, escribiremos:

```
#sendmail -q
```

Para recibir el correo sólo tenemos que ejecutar como el usuario que tiene el fichero `.fetchmailrc` el comando<sup>14</sup>:

```
$fetchmail
```

Los correos así bajados se almacenan en `/var/spool/mail` en espera de que los leamos con Netscape.

Si hemos optado por dejar `fetchmail` a la escucha y deseamos matarlo hay que usar

```
fetchmail --quit
```

Podemos comprobar que funciona mandando un correo al exterior desde una máquina local (sea el linux o no). Si el correo se envía a un usuario registrado del Linux debe ser de la forma: `usuario@nombre_máquina_linux`

### 28.2.3. Montemos un servidor de correo con nuestro propio dominio

Podemos dar una vuelta de tuerca más y si tenemos nuestro propio dominio<sup>15</sup>, utilicémoslo para el correo de nuestro centro.

Veamos un poco más detalladamente cómo funciona el encaminamiento del correo. Cuando alguien va a mandarnos un correo a la dirección `pepito@centrovirtual.org`, se pregunta a los servidores de nombres (DNS) por un tipo de registro especial, el registro MX<sup>16</sup>. El servicio DNS responderá que para el dominio `centrovirtual.org`, el intercambiador de correo es, por ejemplo, `correo.centrovirtual.org`. Pueden devolverse varios nombres con una determinada preferencia. Para el envío del correo se escogerá aquella dirección que esté activa y que tenga mayor preferencia. Suponemos que nos devuelven `mail.centrovirtual.org`, que se corresponde con una dirección pública de nuestro servidor linux del centro. Lo que hemos conseguido es que desde Internet, cualquier dirección de correo que sea del tipo `@centrovirtual.org`, venga a nuestro servidor.

Veamos una sesión de ejemplo con el comando `nslookup` de interrogación a los servidores de nombres:

```
[root@localhost root]# nslookup
```

```
Note: nslookup is deprecated and may be removed from future releases.
Consider using the 'dig' or 'host' programs instead. Run nslookup with
the '-sil[ent]' option to prevent this message from appearing.
```

Decimos que el tipo de registro que queremos es MX (por defecto se pregunta sólo por los A)

```
>set type=MX
```

Preguntamos por el dominio de correo `iesmurgi.org`.

```
>iesmurgi.org
```

```
Server: 150.214.4.34
```

<sup>13</sup>Sendmail está configurado (por defecto) de forma que si pasan más de 4 horas sin entregar el correo nos avisa con una advertencia (no significa que después no lo entregue). La línea del fichero `/etc/sendmail.cf` que permite controlar esto es:

```
Timeout.queuewarn=4h
```

Si no queremos tener esas advertencias deberíamos comentar la línea (#) o poner un espacio de tiempo mayor. En mi caso he optado por poner 1 día:

```
Timeout.queueereturn=1d
```

<sup>14</sup>Si hemos escrito la línea

```
set daemon 300
```

se quedará cargado como demonio y nos bajará el correo cada 5 minutos. Si no la hemos añadido tendremos que ejecutar este comando cada vez que deseemos bajarnos el correo.

<sup>15</sup>Que hemos contratado con un proveedor, por ejemplo, `centrovirtual.org`

<sup>16</sup>De *Mail exchanger* o Intercambiador de Correo. Debemos decirle a nuestro proveedor cuál será el nuestro o, en algunos casos, nosotros podremos editar esta configuración.

Cuando preguntamos por la dirección de un host, por ejemplo `www.centrovirtual.org`, estamos pidiendo al sistema DNS un registro tipo A para ese nombre de máquina.



```
Address: 150.214.4.34#53
iesmurgi.org mail exchanger = 10 tux.iesmurgi.org.
```

Nos ha devuelto que para el dominio `iesmurgi.org`, el intercambiador de correo con preferencia 10 (y único) es `tux.iesmurgi.org`. Si cambiamos a preguntar por la dirección IP de ese host, lo haremos así<sup>17</sup>.

```
>set type=A
>tux.iesmurgi.org
Server: 150.214.4.34
Address: 150.214.4.34#53
Name: tux.iesmurgi.org
Address: 80.32.193.162
```

Ya el correo se dirigirá a nosotros. Ahora debemos preparar el agente de transporte para que reciba el correo de forma adecuada. Veremos el caso de Sendmail<sup>18</sup>. Debemos tener instalados los paquetes `sendmail-8.12.5-7.i386.rpm` (CD1), `sendmail-cf-8.12.5-7.i386.rpm` (CD3) y `sendmail-doc-8.12.5-7.i386.rpm` (CD3).

Históricamente Sendmail se ha considerado una bestia negra de la configuración. No hay más que ver el fichero `/etc/mail/sendmail.cf` y avanzar un poco en su contenido<sup>19</sup>. Pero afortunadamente, una configuración básica no nos llevará demasiado trabajo y podremos mantener la mayoría de los valores por defecto. Para nuestra ayuda disponemos del paquete `sendmail-cf`.

En el directorio `/etc/mail` se encuentra la configuración de sendmail. El fichero `sendmail.cf` es el fichero principal de la configuración. Puede generarse un fichero `sendmail.cf` a partir del fichero `sendmail.mc`. Aquí veremos algunas modificaciones de `sendmail.mc` para conseguir nuestro objetivo: disponer de nuestro servidor de correo en Internet, tanto para enviar como para recibir.

```
$more sendmail.mc
divert(-1)
```

Las líneas que comienzan por `dnl` son comentarios, de tipo texto u opciones, que si nos interesan, podemos descomentar y utilizar

```
dnl This is the sendmail macro config file. If you make changes to this
file,
```

```
dnl you need the sendmail-cf rpm installed and then have to generate a
dnl new /etc/sendmail.cf by running the following command:
dnl
```

Comentario que nos dice cómo se genera un fichero `sendmail.cf` a partir del `sendmail.mc`

```
dnl m4 /etc/mail/sendmail.mc >/etc/sendmail.cf
dnl
```

Incluimos un fichero de configuración

```
include( '/usr/lib/sendmail-cf/m4/cf.m4' )
```

Tipo de sistema operativo y distribución. Sendmail-cf viene preparado para generar configuraciones que sirvan para multitud de máquinas y sistemas operativos.

```
VERSIONID( 'linux setup for Red Hat Linux' ) dnl
OSTYPE( 'linux' )
```

Con la orden `define`, creamos determinadas variables y les asignamos valores. Por ejemplo, en una de ellas decimos que el fichero de alias (`ALIAS_FILE`) se podrá encontrar en `/etc/aliases`

```
define( 'confDEF_USER_ID', '8:12' ) dnl
undefine( 'UUCP_RELAY' ) dnl
undefine( 'BITNET_RELAY' ) dnl
define( 'confAUTO_REBUILD' ) dnl
define( 'confTO_CONNECT', '1m' ) dnl
define( 'confTRY_NULL_MX_LIST', true ) dnl
define( 'confDONT_PROBE_INTERFACES', true ) dnl
define( 'PROCMAIL_MAILER_PATH', '/usr/bin/procmail' ) dnl
```

<sup>17</sup>Todo esto se hace “automáticamente” por los servidores de correo y normalmente no lo vemos.

<sup>18</sup>Existen otros como Qmail

<sup>19</sup>Qué diablos significará: `R$* <@ $* >$* $@ $1 <@ $2 >$3`. ¿Alguien ha derramado el sobre de la sopa de letras en este fichero?



```

define('ALIAS_FILE', '/etc/aliases')dnl
dnl define('STATUS_FILE', '/etc/mail/statistics')dnl
define('UUCP_MAILER_MAX', '2000000')dnl
define('confUSERDB_SPEC', '/etc/mail/userdb.db')dnl
define('confPRIVACY_FLAGS', 'authwarnings,noverfy,noexpn,restrictgrun')dnl
define('confAUTH_OPTIONS', 'A')dnl
dnl TRUST_AUTH_MECH('DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
dnl define('confAUTH_MECHANISMS', 'DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
dnl define('confTO_QUEUEWARN', '4h')dnl
dnl define('confTO_QUEUERETURN', '5d')dnl
dnl define('confQUEUE_LA', '12')dnl
dnl define('confREFUSE_LA', '18')dnl

```

Otras líneas añaden características (FEATURE) o modos de trabajo de nuestro servidor. Normalmente no sólo añaden una línea, sino un conjunto de ellas, incluyendo valores, reglas y modificando el comportamiento del servidor.

```

dnl FEATURE(delay_checks)dnl
FEATURE('no_default_msa', 'dnl')dnl
FEATURE('smrsh', '/usr/sbin/smrsh')dnl
FEATURE('mailertable', 'hash -o /etc/mail/mailertable.db')dnl

```

Por ejemplo, esta característica (virtusertable) nos permite crear usuarios de correo en dominios virtuales y se encarga de añadir todo lo que sea necesario al fichero sendmail.cf para que esto funcione.

```

FEATURE('virtusertable', 'hash -o /etc/mail/virtusertable.db')dnl
FEATURE(redirect)dnl
FEATURE(always_add_domain)dnl
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
FEATURE(local_procmail, '', 'procmail -t -Y -a $h -d $u')dnl
FEATURE('access_db', 'hash -o /etc/mail/access.db')dnl
FEATURE('blacklist_recipients')dnl
EXPOSED_USER('root')dnl
FEATURE('relay_mail_from')dnl
dnl This changes sendmail to only listen on the loopback device 127.0.0.1
dnl and not on any other network devices. Comment this out if you want
dnl to accept email over the network.

```

A esta línea nos referiremos más adelante.

```

DAEMON_OPTIONS('Port=smtp,Addr=80.24.23.22, Name=MTA')
dnl NOTE: binding both IPv4 and IPv6 daemon to the same port requires
dnl a kernel patch
dnl DAEMON_OPTIONS('port=smtp,Addr>:::1, Name=MTA-v6, Family=inet6')
dnl We strongly recommend to comment this one out if you want to protect
dnl yourself from spam. However, the laptop and users on computers that
do

```

```

dnl not have 24x7 DNS do need this.
FEATURE('accept_unresolvable_domains')dnl
dnl FEATURE('relay_based_on_MX')dnl
MAILER(smtp)dnl
MAILER(procmail)dnl
Cwlocalhost.localdomain

```

Si en el directorio /etc/mail ejecutamos como root el comando #make, se actualizarán todas las dependencias que haya sin actualizar. En este caso, si el fichero sendmail.mc se hubiera modificado, se generará a partir de él un fichero sendmail.cf, que es el que tiene la configuración de verdad.

## Luchemos todos contra el spam

El SPAM o correo no solicitado, es un problema dentro de la red. Sin embargo, lo es aún más cuando alguien utiliza nuestra máquina para realizarlo. Por ello, hay una serie de medidas de seguridad que protegen a nuestro servidor de ese peligro y que hacen que en la configuración inicial debamos tener en cuenta una serie de aspectos.

En el fichero `sendmail.mc` existe una línea como la siguiente:

```
DAEMON_OPTIONS( 'Port=smtp,Addr=127.0.0.1, Name=MTA' ) dnl
```

que le indica a sendmail varias cosas. En qué puerto tiene que escuchar<sup>20</sup>, que en este caso pone smtp (25) y la dirección de la interfaz de red en la que escuchará, que debemos cambiar a la dirección de nuestra máquina, porque en la 127.0.0.1 sólo escuchará en el interfaz interno y no podremos conectarnos a él.

Cuando ejecutamos `#make` en el directorio `/etc/mail` y se actualiza el fichero `sendmail.cf`, quedará una línea de este modo, donde ya hemos cambiado a la dirección de nuestra máquina

```
O DaemonPortOptions=Port=smtp,Addr=80.24.23.22, Name=MTA
```

Veamos el fichero `/etc/mail/access`. En éste se le dice, de quién o dónde permitimos que nos envíen correo para que sendmail lo reenvíe. Esta es la función de RELAY o intermediario.

```
localhost.localdomain RELAY
localhost RELAY
127.0.0.1 RELAY
192.168.0.51 RELAY
From:peperez@centrovirtual.org RELAY
```

Estamos viendo que permitimos que utilice nuestra máquina como pasarela de correo (aparte de la interfaz interna), la dirección 192.168.0.51 que corresponde a un PC de nuestro centro. Aquí pondremos todos los puestos desde los que permitamos envío de correo, o la máscara correspondiente (192.168.0.\*) y puede servirnos para que los alumnos envíen su correo desde el centro. Si queremos que desde su casa los profesores puedan enviar correo con la dirección de correo del centro, tenemos la posibilidad de poner las direcciones de correo a las que se lo permitimos. En este caso, el método de dirección IP no nos serviría porque, normalmente, se conectan desde distintos proveedores de acceso y con direcciones dinámicas, que cambian en cada conexión.

## 28.3. Otros servicios

### 28.3.1. Telnet

Llamado también servicio de acceso remoto.

**Telnet** nos permite conectarnos a un ordenador remoto como si estuviéramos físicamente allí: nos da la posibilidad de conectarnos a un ordenador remoto de Internet haciendo que nuestro ordenador local sea una terminal de aquel. La principal utilidad que tiene esto es la de poder ejecutar programas en la máquina remota, así tenemos a nuestra disposición programas o servidores muy potentes capaces de realizar tareas que nuestro PC no sería capaz de realizar.

Para acceder al sistema remoto se nos solicitará la identificación para poder entrar al sistema. Por ejemplo<sup>21</sup> para acceder a la máquina MILETO escribiremos:

```
$telnet mileto.cica.es
```

Con RedHat podemos acceder vía el cliente `telnet` a cualquier máquina remota, pero para ser servidor tenemos que cargar el paquete adecuado, éste es:

```
telnet-server-0.17-23.i386.rpm
```

que se encuentra en el CD1 de la distribución. Además hay que habilitar el servicio (con `ntsysv`, ...), por ejemplo, editando el fichero `/etc/xinetd.d/telnet` y cambiando `disable=yes` por `disable=no`. Después hay que decir a `xinetd` que relea la configuración con

```
#/etc/init.d/xinetd reload
```

<sup>20</sup>Lo hace como demonio independiente

<sup>21</sup>Previamente deberemos haber establecido la conexión con nuestro proveedor de Internet.

Comentar que igual que pasa con ftp, telnet “pasea” las contraseñas en texto plano por Internet. Una forma segura de telnet en nuestra intranet puede ser la de usar la seguridad adicional de tcpwrappers de la forma<sup>22</sup>:

```
$cat /etc/hosts.allow
in.telnetd: 192.168.1. 127.0.0.1
$cat /etc/hosts.deny
in.telnetd: ALL
O bien usando el propio xinetd, añadiendo al fichero /etc/xinetd.d/telnet la línea:
only_from = 192.168.0.0/16 127.0.0.1
y después:
#/etc/init.d/xinetd reload
```

### ➡ Para practicar

1. Cargar un servidor de telnet, modificar el texto de bienvenida (/etc/issue.net) y conseguir que los accesos estén limitados a la máquina local y a la subred de clase C 192.168.1.0.
2. Es interesante probar la posibilidad que nos ofrece Linux de trabajar en modo gráfico con programas situados en otro equipo, para esto tendremos que:

Desde un Xterm de la máquina local ejecutaremos<sup>23</sup>

```
$ xhost +máquina_remota
```

después haremos un telnet a la máquina remota y una vez conectados escribiremos

```
$ export DISPLAY=máquina_local:0
```

por último ya sólo tenemos que ejecutar el comando que deseemos, por ejemplo, podéis probar con

```
$ mozilla &
```

## 28.3.2. Ssh

**ssh** Protocolo *Secure Shell*, se usa para conexiones de red cifradas y autenticadas de forma más segura.

En RedHat disponemos de este servicio gracias a los paquetes `openssh*`. Haciendo uso de este juego de herramientas podemos conseguir que se cifre el tráfico (incluyendo passwords) de una conexión de red, y así evitamos la posibilidad de ser “espiados” cuando establecemos una comunicación entre nuestro ordenador y una máquina remota.

Para ampliar sobre el tema os remitimos:

- Capítulo 9 del *Manual oficial de referencia de Red Hat Linux*
- Capítulo 14 del *Manual oficial de personalización de Red Hat Linx*
- La web de OpenSSH en <http://www.openssh.com/>
- Un artículo sobre este tema en la Web <http://gaceta.piensa.com>

Por defecto se cargan los paquetes<sup>24</sup>:

```
openssh-3.4p1-2
openssh-clients-3.4p1-2
openssh-askpass-3.4p1-2
openssh-askpass-gnome-3.4p1-2
```

Con ellos podemos conectarnos como cliente usando `ssh`. Para ser servidor se necesita el paquete del CD1:

<sup>22</sup>Que no se olvide pulsar **INTRO** después de escribir la líneas

<sup>23</sup>Donde `máquina_remota` es o bien la dirección IP de la máquina remota, o bien, el nombre de esa máquina

<sup>24</sup>Además se necesita el paquete `openssl`

`openssh-server-3.4p1-225`

y habilitar el servicio, por ejemplo con:

```
# /etc/init.d/sshd start
```

La conexión ssh es fácil de usar, basta con escribir:

```
$ssh "máquina_remota"
```

Soporta de forma completa la ejecución de aplicaciones en el entorno X. Cuando un cliente se conecta a un servidor SSH intercambian claves de cifrado y a partir de ahí el servidor autentifica al cliente, bien con RSA o bien mediante contraseña. Cuando se ha iniciado la conexión, el servidor lanza un servidor X ficticio de forma que las aplicaciones que se ejecutan en el servidor SSH se conectan al servidor X ficticio y de "forma segura" SSH reenvía los datos al servidor X real. Es más seguro y fácil de usar que `telnet` y para ejecutar una aplicación gráfica (una vez realizada la conexión usando `ssh`) sólo hay que ejecutar el programa desde la `xterm` que estemos usando para la conexión.

Una información añadida, clientes para Windows (gratuitos)<sup>26</sup>:

- Para ssh, sftp, scp en modo comando:

`putty` <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

- Para sftp en modo gráfico:

`SecEx Security Express Lite` <http://www.bytefusion.com/>

➡ **Para practicar:** Trabajar y montar un servidor de ssh

1. Activar un servidor ssh, para eso:

- a) Instalar el paquete del CDI<sup>27</sup>

```
# rpm -i openssh-server-3.4p1-2.i386.rpm
```

- a) Activar el servicio:

```
# /etc/rc.d/init.d/sshd start
```

- b) Comprobar que funciona:

```
# ssh localhost
```

y comprobar que podemos ejecutar aplicaciones gráficas sin más. Después con `sftp`:

```
# sftp localhost
```

2. Desde W... instalar los programas `putty` (no necesita instalación) y `SecEx` e iniciar una comunicación segura con un servidor de prueba

## 28.4. Linux como enrutador

Con el uso de Squid solucionamos el problema de que las máquinas de nuestra red, usando Linux como "caché", puedan salir a Internet para obtener páginas web. Pero no podríamos conectarnos vía ssh desde una máquina local que no tenga acceso directo a Internet con otra máquina externa. Podemos solventar esto usando las:

**ipchains** son de los núcleos 2.2 (si deseamos trabajar con ellas hay que instalar el paquete del CD3 `ipchains-1.3.10-16.i386.rpm`).

**iptables** las de los núcleos 2.4 (Red Hat trabaja con ellas por defecto)

La idea de ambos es común: permiten crear reglas de filtrado de paquetes, es decir, administrar un firewall. Tanto su uso como las diferencias entre ambos programas aparecen documentadas en:

- Capítulo 7 del *Manual Oficial de Personalización de Red Hat Linux*

<sup>25</sup>Se debería actualizar a la última versión disponible para no tener problemas de seguridad.

<sup>26</sup>Hay más: `winSCP2`, `iXplorer`, ...

<sup>27</sup>Mejor si se instala la última versión y no éste <https://rhn.redhat.com/errata/rh8-errata.html>

- Capítulo 18 del *Manual de referencia de Red Hat Linux*
- *Guía de Administración de Redes*
- Manual sobre Iptables accesible desde la web del curso (24 páginas en castellano).

En este apartado sólo vamos a ver cómo conseguir que nuestra máquina linux reenvíe y enmascare hacia el exterior el tráfico local sin entrar en nada más. Para ampliar sobre el tema mirar los manuales antes comentados así como la documentación que acompaña ambos programas. Se va a ejemplificar sobre una conexión a internet usando una ADSL conectada en el interfaz `eth0` en la que Linux hace de enrutador (`eth1`). Para usarlo con una conexión vía módem sólo hay que cambiar el interfaz sobre el que se permite el enmascaramiento y todo queda igual.

El núcleo de linux permite que los datos atraviesen el firewall sólo si cumplen unas determinadas reglas llamadas filtros. Estos filtros examinan la cabecera de los datos (paquetes) y a partir de la información que obtengan pueden:

**accept** aceptar el paquete

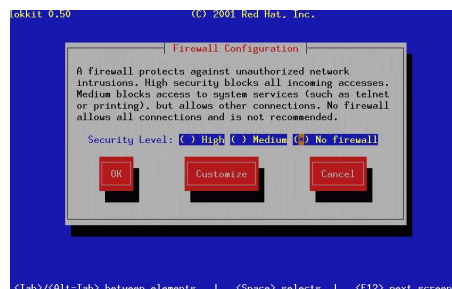
**reject** descartar el paquete e informar al origen de que se ha descartado

**deny** descartar el paquete “pasando” de él (hace como que no lo ha recibido).

### 28.4.1. iptables

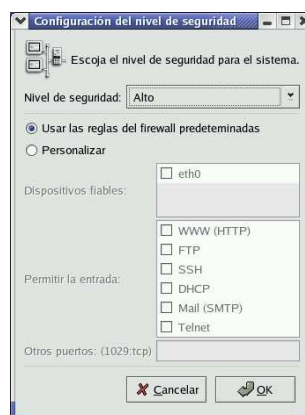
Como se ha comentado, por defecto es el sistema que carga Red Hat. Se puede configurar con:

```
# lokkit
```

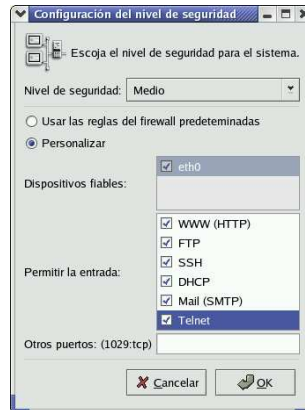


o de forma gráfica con (  → **Configuración del sistema** → **Security Level**):

```
#redhat-config-securitylevel
```



La salida de ambos programas se almacena en `/etc/sysconfig/iptables28` y, es a partir de ese fichero como se configura el *firewall* en el arranque. Si optamos por permitir que el interfaz de red `eth0` permita todas las opciones de configuración (servir páginas web, ...)



sería de la forma:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Lokkit-0-50-INPUT - [0:0]
-A INPUT -j RH-Lokkit-0-50-INPUT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 80 --syn -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 21 --syn -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 22 --syn -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 25 --syn -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 23 --syn -j ACCEPT
-A RH-Lokkit-0-50-INPUT -i lo -j ACCEPT
-A RH-Lokkit-0-50-INPUT -i eth0 -j ACCEPT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 0:1023 --syn -j REJECT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 2049 --syn -j REJECT
-A RH-Lokkit-0-50-INPUT -p udp -m udp --dport 0:1023 -j REJECT
-A RH-Lokkit-0-50-INPUT -p udp -m udp --dport 2049 -j REJECT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 6000:6009 --syn -j REJECT
-A RH-Lokkit-0-50-INPUT -p tcp -m tcp --dport 7100 --syn -j REJECT
COMMIT
```

Si optamos por seleccionar ninguna regla de filtrado este fichero aparecería vacío. Para optimizar las reglas de filtrado podemos usar `lokkit` y después depurar el fichero generado.

Tanto el reenvío como el enmascaramiento son servicios a nivel de núcleo (afortunadamente están ya compilados en nuestros núcleos). Para habilitar el reenvío podemos hacerlo de dos formas:

- En “caliente”:

```
# echo 1 >/proc/sys/net/ipv4/ip_forward
```

- Para que sea siempre la opción por defecto:

Modificando una línea del fichero `/etc/sysctl.conf`, debería quedar:

```
net.ipv4.ip_forward = 1
```

<sup>28</sup>Si optamos por *Ningún Firewall* se borra el fichero

Podemos comprobar que esta opción está activa si obtenemos 1 al ejecutar:

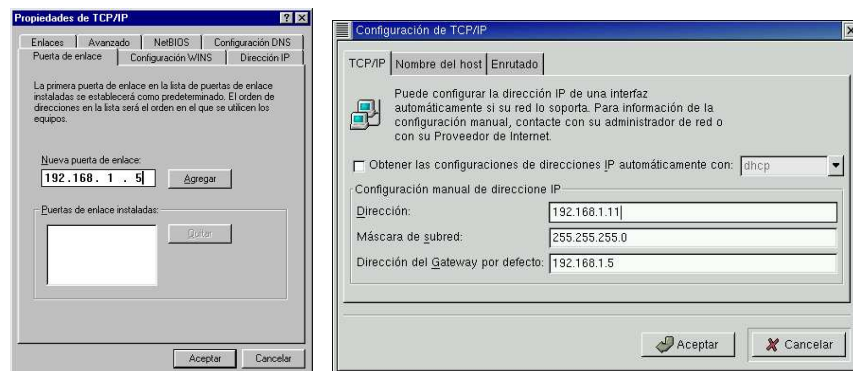
```
# cat /proc/sys/net/ipv4/ip_forward
1
```

Si tenemos esto ya listo, y queremos permitir que las máquinas de la red de clase C 192.168.1.0 puedan salir al exterior, unas posibles reglas (para conexión ADSL por el interfaz eth0 e interfaz de red para la red local eth1) serían<sup>29</sup>:

```
# iptables --flush
# iptables --table nat --flush
# iptables --table nat --append POSTROUTING --out-interface eth0 -j MASQUERADE

# iptables --append FORWARD --in-interface eth1 -j ACCEPT
```

Para ver que funciona tendremos que poner como pasarela por defecto en las otras máquinas la IP de la máquina Linux que se encarga de hacer de firewall (por ejemplo 192.168.1.5). Habrá que configurar las máquinas windows y/o las Linux (con neat<sup>30</sup>) que queramos que salgan con:



Una vez comprobado que todo funciona (por ejemplo podemos hacer un ping a una dirección IP externa desde la máquina Windows) almacenaremos las reglas actuales (que ya sabemos que están bien) en el fichero /etc/sysconfig/iptables usando el comando:

```
# iptables-save >/etc/sysconfig/iptables
```

Si deseamos ver qué reglas hay activas podemos ejecutar

```
# iptables -L
```

Al hacerlo así, la próxima vez que iniciemos Linux tendremos nuestro firewall Linux preparado para enmascarar las direcciones de todo el tráfico de las máquinas locales destinado a direcciones externas.



Para poder resolver nombres hay que poner las IP de los servidores de nombres en cada una de las máquinas en las que añadamos la puerta de enlace.

Podemos comprobar los cambios viendo el fichero /etc/sysconfig/iptables.

Si usamos ahora lokkit reescribirá el fichero anterior y perderíamos el trabajo realizado.

## 28.5. Sistema de ficheros NFS

Para conocer más de este tema podeis mirar en el *Howto Servidor De Raíces NFS Mini-COMO*, el libro comentado en la introducción de esta entrega *Guía de Administración de Redes* y como no:

- \* Capítulo 17 del *Manual oficial de referencia de Red Hat Linux*

- \* Capítulo 15 del *Manual oficial de personalización de Red Hat Linx*

<sup>29</sup>Las dos primeras son para limpiar las reglas del iptables y no son fundamentales. Con las dos últimas activamos el NAT con enmascaramiento en las reglas del iptables.

<sup>30</sup>Mirar en la página 361



El sistema de ficheros NFS (*Network File System*) es la forma en que podemos compartir sistemas de ficheros entre máquinas Linux. Con él podemos acceder a sistemas remotos de igual forma que si son sistemas de ficheros locales.

Para disponer de los servicios NFS tenemos que tener activos los demonios `nfsd` y `mountd` que forman parte del paquete

```
nfs-utils-1.0.1-2.i386.rpm
```

En consecuencia, lo primero que tendremos que comprobar es si tenemos cargado ese paquete<sup>31</sup> y si están activos. Se pueden activar con

```
# service nfs start
```

Si deseamos que se activen al inicio lo mejor es activar en el inicio el demonio `nfs` usando el panel de control o, directamente como root, usar las herramientas:

- En modo gráfico: `serviceconf`
- En modo alfanumérico: `ntsysv`

Una vez que todo esté correcto controlaremos el proceso de montaje con los ficheros:

- `/etc/fstab` en la máquina cliente (máquina CICA) en donde montemos el sistema de ficheros.

y

- `/etc/exports` en la máquina servidor (máquina THALES) del sistema de ficheros

El primero (esto es opcional) nos va a permitir no tener que pasar todos los parámetros al comando `mount` cuando queramos montar un sistema de ficheros remoto, incluso nos permitiría que el sistema remoto se monte de forma automática al arrancar la máquina. Una posible entrada para ese fichero que permite montar el sistema de ficheros remoto `/var/tux` de la máquina THALES en `/mnt/thales` de la máquina CICA es:

```
thales:/var/tux /mnt/thales nfs noauto,rw,user
```

Donde lo único que añadir a lo ya visto en entregas anteriores es que el origen de montaje es de la forma:

```
máquina_remota:origen
```

Para que podamos acceder al directorio `/var/tux` de la máquina THALES tenemos que permitirlo mediante el fichero `/etc/exports` de esa máquina. Su formato es:

```
directorio hostname(opciones32)
```

En este fichero tiene que aparecer una línea para cada directorio que se quiera exportar así como las opciones con las que se quiere exportar. Por ejemplo si sólo deseamos que la máquina CICA acceda al subdirectorio antes mencionado pondríamos en ese fichero:

```
$cat /etc/exports  
/var/tux cica(rw)33
```

---

<sup>31</sup>Se instala por defecto.

<sup>32</sup>Estos parámetros son opcionales

<sup>33</sup>Es importante resaltar que entre el nombre de la máquina y las opciones no puede haber espacios, Si ponemos un espacio ¿qué se conseguiría?, ¿quién tendría de verdad permisos de lectura y escritura y quién sólo de lectura?



y con las opciones entre paréntesis indicamos que permitimos que se escriba en él.

Si modificamos el fichero `/etc/exports` tenemos que reiniciar los demonios NFS para que actualicen la nueva configuración. Podemos hacerlo con:

```
#/etc/rc.d/init.d/nfs restart34
```

Para conocer más de este fichero:

```
$ man exports
```

Una vez configurados esos ficheros, para montar el sistema de ficheros `/home/tux` de la máquina THALES en CICA escribiríamos:

```
$ mount /mnt/thales
```

y listo.<sup>35</sup>

Para desmontar el sistema de ficheros lo haremos con el comando `umount`.

## 28.6. Compartir impresoras entre máquinas Linux

Compartir una impresora en una red Linux es bastante sencillo y se controla con el fichero `/etc/lpd.perms`, que por defecto no permite las conexiones.

Para un control más estricto de qué máquinas vamos a permitir que impriman usando nuestro linux:

```
man lpd.perms
```

Si deseamos permitir conexiones de otras máquinas hemos de comentar la línea:

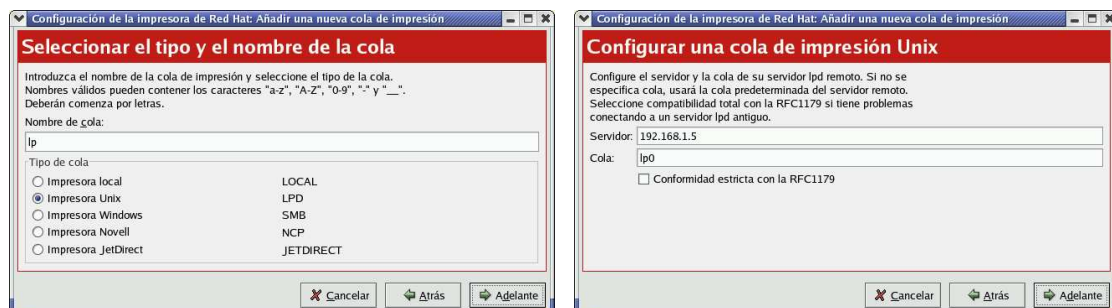
```
# REJECT SERVICE=X NOT SERVER
```

Si hacemos algún cambio en ese fichero y queremos que se active:

```
#/etc/init.d/lpd reload
```

Después en la máquina remota, añadiremos la impresora en modo gráfico con `redhat-config-printer` las opciones para la configuración:

- Impresora Unix
- Servidor: 192.168.1.5
- Cola: nombre de la cola de impresión (`lp0`).
- Modelo: Impresora PostScript para que no se “pisen” los filtros.



<sup>34</sup>También funcionará:

```
#killall -HUP rpc.nfsd rpc.mountd
```

<sup>35</sup>En el caso de que no pongamos ninguna entrada en el fichero `/etc/fstab` de la máquina CICA tendríamos que escribir:

```
# mount -t nfs thales:/var/tux /mnt/thales
o sólo
#mount thales:/var/tux /mnt/thales
```



Si mandamos la página de prueba podemos ver si todo ha ido bien.

# Capítulo 29

## Samba

Samba es la idea de Andrew Tridgell, ... Unos cuantos años después, él lo expandió como su servidor SMB particular y comenzó a distribuirlo como producto por Internet bajo el nombre de servidor SMB. Sin embargo, Andrew no pudo mantener ese nombre -ya pertenecía como nombre de producto de otra compañía-, así que intentó lo siguiente para buscarle un nuevo nombre desde Unix:

```
grep -i 's.*m.*b' /usr/dict/words
```

y la respuesta fue:

```
salmonberry samba sawtimber scramble
```

De ésta manera nació el nombre de Samba.

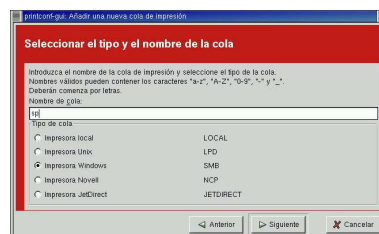
(Usando Samba, ROBERT ECKSTEIN y otros)

### 29.1. ¿Qué es Samba?

La página principal de Samba es:

<http://www.samba.org>

Mediante Samba y a través del protocolo TCP/IP podemos compartir y utilizar recursos de sistemas de ficheros Unix e impresoras con otros sistemas operativos<sup>1</sup> (discos duros e impresoras). Samba es rápido y sencillo de configurar. Linux con Samba puede trabajar como servidor y como cliente. Como servidor ofrece recursos (discos e impresoras) para que los utilicen las máquinas windows. Como cliente utiliza los servicios ofrecidos por las máquinas windows (discos e impresoras<sup>2</sup>)



Con Samba disponemos también de servicios de dominios NT.<sup>3</sup>

Samba debe sus “orígenes” a Andrew Tridgell. Necesitaba poder compartir archivos desde el DOS a su servidor UNIX y consiguió el primer programa sobre 1992 que si bien funcionó dejaba bastante que

<sup>1</sup>Sistemas Windows 3.11, 9x, NT, 2000, XP y OS/2

<sup>2</sup>Os remitimos a la documentación comentada más adelante para esto

<sup>3</sup>Si proporciona servicios de archivo y de impresión, es mejor como NT que un NT y es gratis, la pregunta es obvia.

desear. En 1994 y tras retomar el proyecto inicial pero ahora con la idea de interconectar en la misma red Windows y Linux apareció de forma “oficial” Samba. Con Samba disponemos de los servicios:

- Servicios de archivos y de impresión.
- Autenticación y autorización.
- Servicios de resolución de nombres.
- Servicios de exploración.

Para conseguir estos servicios tenemos a nuestra disposición los programas<sup>4</sup>:

**smbd** demonio SMB (*Session Message Block*, Bloque de mensajes de sesión), se encarga de los servicios de archivos, de impresión y autenticación y autorización

**nmbd** demonio de servidor de nombres NetBIOS.

**winbindd** demonio para resolver nombres con servidores NT.

Además de estos demonios en los paquetes que componen el programa Samba tenemos entre otros:

**nmblookup** se usa para consultar nombres de NetBIOS y mapearlos a direcciones IP.

**smbclient** cliente tipo ftp .

**smbmount** para montar sistemas compartidos SMB en nuestra máquina Linux.

**smbumount** para desmontar un sistema de archivos SMB ya montado.

**smbadduser** para añadir usuarios.

**smbpasswd** para cambiar la contraseña de acceso SMB tanto local como remota.

**smbprint** smbclient reducido que permite imprimir en los recursos de impresión compartidos SMB.

**smbstatus** utilidad para mostrar las conexiones SMB activas.

**smbtar** para hacer copia de seguridad de los sistemas de archivos compartidos SMB en una unidad de cinta de nuestra máquina Linux.

**swat** utilidad para configurar SAMBA usando un navegador<sup>5</sup>.

**testparm** Revisa/prueba los archivos de configuración de SAMBA.

**testprns** Para revisar el nombre de impresora para usarlo con SAMBA.

Para afinar mejor la configuración y estudiar los más de 120 parámetros os remitimos a

- la extensa documentación que acompaña al programa  
`/usr/share/doc/samba-2.2.5`
- la traducción del libro de O'Reilly *Usando SAMBA* disponible en (imprescindible):  
<http://es.tldp.org/Manuales-Lucas/USANDO-SAMBA/>  
<http://www.sobl.org/modules.php?name=Downloads>
- al cómo *Samba-Como*  
<http://lucas.hispalinux.es/COMO-INSFLUG/COMOs/Samba-Como/Samba-Como.html>

---

<sup>4</sup>No están todas las utilidades que contienen los paquetes que componen samba.

<sup>5</sup>Hay que instalarlo

- un documento traducido por PEDRO P. FÁBREGA llamado *Configuración y Uso de Samba* que podéis conseguir en:  
<http://www.arrakis.es/~pfabrega>  
en él se explican todos los parámetros del archivo de configuración de SAMBA.
- a una página personal  
<http://teleline.terra.es/personal/garzones/garzalin.html>
- además, en Inglés, tenemos los Howtos (están más actualizados que los que hay en castellano):  
<http://www.tldp.org/HOWTO/Samba-Authenticated-Gateway-HOWTO.html>  
<http://www.tldp.org/HOWTO/SMB-HOWTO.html>  
<http://www.tldp.org/HOWTO/Windows-LAN-Server-HOWTO.html>
- y como no, a las manpages de los programas.

## 29.2. Instalación

El programa consta de tres paquetes que están a nuestra disposición en el CD2<sup>6</sup>:

- Paquete “principal” de la aplicación. Al instalarlo dispondremos de un servidor SMB:  
samba-2.2.5-10.i386.rpm
- Si está instalado dispondremos de varios clientes SMB que complementan el sistema de ficheros SMB. Permiten acceder a recursos compartidos SMB (archivos e impresoras)  
samba-client-2.2.5-10.i386.rpm
- Proporciona ficheros necesarios para los paquetes anteriores  
samba-common-2.2.5-10.i386.rpm

Si instalamos sólo los dos últimos ¿qué disponibilidad tenemos? Muchísima, podemos ser clientes de máquinas windows. Dispondríamos ya (entre otros ficheros) de las utilidades:

/usr/bin/nmblookup	/usr/bin/smbtar
/usr/bin/smbclient	/usr/bin/smbumount
/usr/bin/smbmnt	/usr/bin/make_printerdef
/usr/bin/smbmount	/usr/bin/make_smbcodepage
/usr/bin/smbprint	/usr/bin/testparm
/usr/bin/smbpool	/usr/bin/testprns

y del fichero de configuración de Samba  
/etc/samba/smb.conf

Instalemos el servidor Samba en nuestra máquina, montamos el CD2 y:

```
# rpm -ivh samba-2.2.5-10.i386.rpm
error: Failed dependencies:
    samba-common = 2.2.5 is needed by samba-2.2.5-10
    perl(CGI) is needed by samba-2.2.5-10
Suggested resolutions:
    perl-CGI-2.81-55.i386.rpm
```

---

<sup>6</sup>SAMBA 2.2 solventa la incompatibilidad con clientes y servidores 2000.



```
samba-common-2.2.5-10.i386.rpm
```

Instalemos los paquetes señalados (son todos del CD2), samba y samba-client en el orden adecuado:

```
# rpm -ivh perl-CGI-* samba-*
y pongámoslo en funcionamiento7:
# service smb start
```

Una vez instalados los paquetes estamos casi listos para funcionar, ya que los «demonios» que requiere se ponen en marcha por defecto al arrancar el sistema operativo<sup>8</sup>. Antes de activarlos tendremos que configurar la máquina Linux y la máquina Windows.

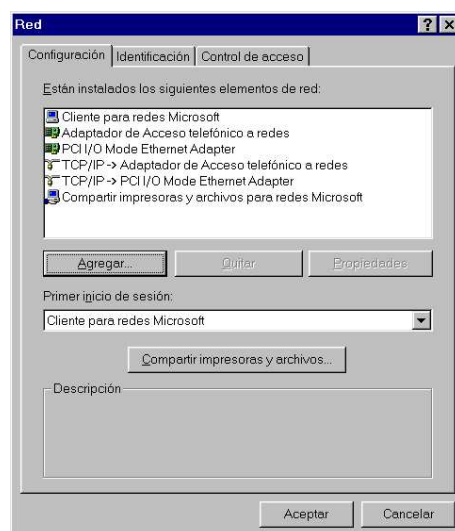
Partiremos de una red privada con grupo de trabajo THALES con la siguiente configuración:

Sistema Operativo	Nombre	IP
Linux (RedHat 8)	secre	192.168.1.5
Windows 95	bag	192.168.1.11
Windows 95 OSR2/Windows 98	compa	192.168.1.12

El proceso consiste:

### 29.3. Configuración de la máquina Güindows 9x

Para trabajar con Samba tendremos que tener cargados los protocolos TCP/IP, por tanto, en Red añadiremos esos protocolos



y asignaremos una dirección IP a nuestras máquinas, por ejemplo, 192.168.1.11, en máscara de Red pondremos 255.255.255.0

Pondremos un nombre a nuestra máquina (BAG) y el grupo de trabajo en el caso de que no estuviese ya definido (THALES). Repetiremos este proceso con la máquina COMPA.

<sup>7</sup>Deberíamos activarlo para los niveles de inicio en los que trabajemos para que lo active en el arranque.

<sup>8</sup>Si deseamos activarlos sin reiniciar el sistema escribiremos:

```
# /etc/rc.d/init.d/smb start
Starting SMB services: [ OK ]
Starting NMB services: [ OK ]
```



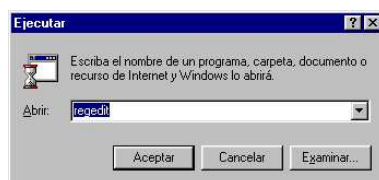
### Problemas con las contraseñas.



Si todos nuestros equipos con Windows son posteriores a Win95 OSR2, no tendremos que hacer nada de esto y lo mejor es trabajar con contraseñas encriptadas. Incluso si no estamos seguros lo mejor es dejar este tema aparcado. Sólo en el caso de que tengamos una máquina con un Windows antiguo con la que no conseguimos acceder al servidor SMB y sin embargo sí podemos usar el resto de servicios es cuando deberíamos plantearnos si el problema puede estar aquí. Para saber cómo conseguirlo mirar en la página 413

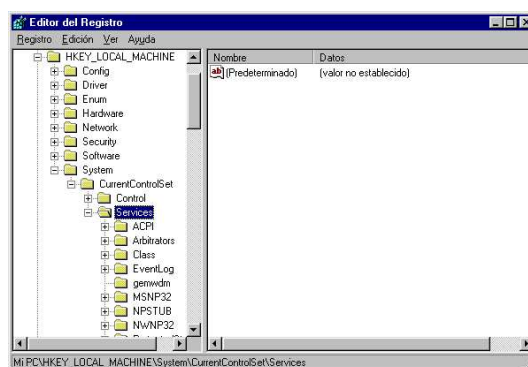
Hay un problema que puede que tengamos que solucionar antes de seguir. La problemática reside en que Microsoft cambió el sistema de trabajo con las contraseñas a partir de Windows 95 OSR2<sup>9</sup>. En versiones anteriores, las contraseñas se pasan como texto plano pero a partir de esta versión (incluyendo NT) las contraseñas están encriptadas de manera predeterminada. ¿Qué significa esto?, pues que si no podemos trabajar con contraseñas encriptadas debemos hacérselo saber a nuestro Windows 98 (máquina COMPA), para eso tendremos que modificar el registro<sup>10</sup> de Windows siguiendo los pasos:

1. En Windows, tendremos que ejecutar el editor del registro, regedit



y después llegar hasta la entrada

**HKEY\_LOCAL\_MACHINE→System→CurrentControlSet→Services→ VxD→VNETSUP**



2. Si ya tenemos una entrada llamada EnablePlainTextPassword, tendrá el valor de 0 ó de 1.
  - a) Un valor de 0 significa que el sistema está usando contraseñas encriptadas.
  - b) Para desactivarlo, cambiar el valor a 1.
3. Si no tenemos una entrada con ese texto, pulsemos sobre **Edición→Nuevo→Valor DWORD** y tras crearlo definámoslo con el valor 1.<sup>11</sup>

<sup>9</sup>Véase /usr/share/doc/samba-2.2.5/docs/textdocs/Win95.txt

<sup>10</sup>La edición del registro es "peligrosa" y puede que después tengamos problemas con el sistema operativo.

<sup>11</sup>Es decir, no trabajaremos con contraseñas encriptadas. Esto nos va a obligar a comentar varias líneas del fichero /etc/samba/smb.conf. Son las líneas:

```
; encrypt passwords = yes
; smb passwd file = /etc/samba/smbpasswd
```



4. Para finalizar tendremos que guardar los cambios y reiniciar Windows.

Notar que con la máquina BAG no tenemos que hacer ningún cambio.

## 29.4. Configuración de la máquina Linux

### 29.4.1. Análisis del archivo de configuración de Samba

En Red Hat, el archivo de configuración de Samba es `/etc/samba/smb.conf`<sup>12</sup>. Mediante este archivo podemos controlar la gran cantidad de opciones disponibles del programa y tratar a fondo cada una de ellas está más allá del objetivo de esta entrega.

Se divide en varias secciones, cada una de ellas determinada por un nombre entre corchetes: `[impresoras]`, `[global]`, etc. En cada sección encontramos una serie de parámetros compuestos por pares de clave/valor. Los comentarios comienzan con punto y coma (;) o mediante una almohadilla (#).<sup>13</sup> Ejemplifiquemos esto con una parte de la sección `[global]`:

```
[global]
# La almohadilla indica que estamos en un comentario
remote announce = 192.168.1.255 192.168.2.44
# El siguiente parámetro está marcado como comentario y
# el segundo no, es decir, está activo
; local master = no
os level = 33
```

Si modificamos el fichero de configuración tendremos que reiniciar el servidor<sup>14</sup> con:

```
# /etc/rc.d/init.d/smb restart
```

Para comenzar a trabajar y conocer las posibilidades que nos ofrece el programa sólo vamos a modificar las opciones más usuales de este fichero.

#### Sección `[global]`

En esta sección configuraremos parámetros para todo el servidor SAMBA así como algunos valores predeterminados a las otras secciones. Veamos algunas de las opciones más usuales<sup>15</sup>.

Comencemos por ajustar el grupo de trabajo<sup>16</sup> en el que nos encontramos. Por ejemplo, si nuestro grupo de trabajo es THALES<sup>17</sup>, escribiremos:

```
workgroup=THALES
```

Con el parámetro

```
server string = Samba Server
```

indicamos el nombre que identificará al servidor cuando lo consultan los clientes SAMBA.

La línea que sigue aparece marcada como comentario. Si la activamos con el valor

```
hosts allow = 192.168.1. 127.
```

conseguimos que el acceso al servidor de Samba esté restringido a los hosts o redes especificados. En este caso el acceso está limitado a la red 192.168.1.0/24 y al host local. Si deseamos limitar el acceso a una

<sup>12</sup>Antes de tocar este archivo, como viene siendo habitual, deberíamos hacer una copia por si acaso.

<sup>13</sup>Normalmente aparecen marcados los parámetros como comentarios utilizando el punto y coma, y se deja la almohadilla para comentarios normales.

<sup>14</sup>Es deseable actualizar el fichero `/etc/hosts` con las direcciones IP y el nombre de cada máquina a la que vamos a acceder. De esta forma con sólo escribir el nombre del equipo al que queremos acceder, el servidor buscará en ese fichero el nombre del equipo y dirección IP correspondiente.

<sup>15</sup>Para conocer todas las opciones véase el libro *Usando Samba* ya comentado.

<sup>16</sup>Deberá estar limitado como máximo a nueve caracteres, sin espacios y todos en mayúsculas.

<sup>17</sup>Por defecto el grupo es MYGROUP.



red de clase B escribiríamos 192.168. y si es de clase A, 192. Con `hosts deny` podemos negar el acceso a determinadas máquinas o subredes (en caso de duda “gana” `hosts allow`). Por ejemplo, con

```
hosts allow = 192.168 EXCEPT 192.168.1
```

permitimos el acceso a la red de clase B 192.168. pero denegamos el acceso a la subred de clase C 192.168.1

Por defecto el archivo `smb.conf` permite que estén disponibles todas las impresoras definidas en `/etc/printcap` como recursos compartidos.

```
# si se quiere cargar automáticamente la lista de impresoras en lugar
# de configurarlas por separado, será necesario esto
printcap name = /etc/printcap
load printers = yes
```

```
# No será necesario especificar el tipo de sistema de impresión a menos
# que no sea estándar. Los sistemas con soporte en la actualidad incluyen:
# bsd, sysv, pip, lprng, aix, hpux, qnx
printing = lprng
```

Si se quiere especificar una cuenta de usuario *guest* (invitado) para el acceso anónimo a los servicios en un servidor Samba descomentaremos esta línea. Esto no es imprescindible, por omisión se permite el acceso de un usuario *guest* mediante la cuenta *nobody*<sup>18</sup>.

```
# Quitar la marca de comentario aquí si se desea una cuenta de usuario
guest,
# se debe añadir esto a /etc/passwd, de no ser así, se utiliza el usuario
# nobody
; guest account = pcguest
```

Las líneas que siguen nos indican el lugar en dónde se almacenarán los logs del sistema SMB así como el tamaño máximo que pueden tener<sup>19</sup>:

```
log file = /var/log/samba/%m.log
max log size = 50
```

Con la entrada que sigue nos garantizamos que sólo los usuarios de la máquina Linux tienen acceso vía SMB

```
# Modo de seguridad. La mayoría querrá nivel de seguridad de usuario.
Ver
# security_level.txt para más detalles.
security=user
```

Si en vez de `user` escribimos `share` podrán acceder al servidor usuarios no registrados en el sistema. Es mejor si se desea un acceso menos restringido al servicio de impresión o crear una zona pública limitada por la seguridad del recurso.

Con las líneas que siguen (están comentadas por defecto y no deberían descomentarse salvo que tengamos problemas) establecemos el número de caracteres (no distinguen entre mayúsculas y minúsculas) que comprobaremos de los nombres de usuario así como de su contraseña.

```
# El nivel de password permite que concuerden _n_ caracteres de la contraseña
# para todas las combinaciones de mayúsculas y minúsculas.
; password level = 8
; username level = 8
```

Como ya ha hemos visto, deberíamos descomentar las líneas que aparecen a continuación si trabajamos con Windows 95 OSR2 y siguientes. De esta forma trabajaríamos con contraseñas encriptadas por el servidor Samba:

```
# Si se quiere usar encriptación de la password, consúltese, por favor
# ENCRYPTION.txt, Win95.txt y WinNT.txt en la documentación de Samba.
# NO activar esta opción a menos que se hayan consultado dichos documentos
encrypt passwords = yes
```

<sup>18</sup>Sobre la forma de dar de alta un usuario véase la orden `smbadduser`

<sup>19</sup>Con un valor de 0 no ponemos ningún límite en cuanto al tamaño de los ficheros.



```
smb passwd file = /etc/smbpasswd
```

Permite a Samba actualizar el fichero de contraseñas de Linux cuando un usuario cambia su contraseña encriptada<sup>20</sup> (fichero `/etc/samba/smbpasswd`).

```
# The following are needed to allow password changing from Windows to
# update the Linux system password also.
# NOTE: Use these with 'encrypt passwords' and 'smb passwd file' above.
# NOTE2: You do NOT need these to allow workstations to change only
# the encrypted SMB passwords. They allow the Unix password
# to be kept in sync with the SMB password.
unix password sync = Yes
passwd program = /usr/bin/passwd%u
passwd chat = *New*password*%n\n *Retype*new*password*%n\n ...
```

Podemos conseguir que Linux autentifique a los clientes Windows en la red. Por defecto, estas líneas están comentadas:

```
# Actívese esto si se desea que Samba sea un servidor de inicio de sesión
de
# dominio para estaciones de trabajo de Windows95.
; domain logons = yes

# Si se activan los inicios de sesión de dominio puede ser necesario un

# script de inicio de sesión por máquina o por usuario ejecútase un archivo
# específico de procesamiento por lotes de inicio de sesión por máquina
; logon script = %m.bat

# ejecútase un archivo específico (de procesamiento por lotes de inicio
de
# sesión por cada usuario
; logon script = %U.bat
```

### Sección [homes]

Con esta sección podemos controlar de qué forma accederán los clientes al directorio principal de usuario en el servidor Linux. La configuración aquí establecida permite que los parámetros sean válidos para todos los usuarios y no hay que especificar una configuración para cada usuario por separado.

```
comment = Home Directories
browseable = no
writable = yes
valid users = %S
create mode = 0664
directory mode = 0775
```

Analicemos cada una de las líneas anteriores:

**comment** cadena de identificación que se muestra a los clientes que examinan el servidor Samba.

<sup>20</sup>Para ampliar sobre este tema y comprender las líneas no comentadas del fichero no estudiadas mirar `/usr/share/doc/samba-2.2.5/docs/html/docs/ENCRYPTION.html`, el capítulo 16 de *The Official Red Hat Linux Customization Guide* y el capítulo 7 de *The Official Red Hat Linux Reference Guide*



**browseable** al establecerlo a no conseguimos que el explorador de Windows no nos muestre los `$HOME` de otros usuarios del sistema Linux. Si no somos ningún usuario registrado del sistema no veríamos ningún `$HOME`. Sin embargo, si nos conectamos como un usuario del sistema Linux aparecerá una carpeta de nombre ese usuario a la que podremos acceder con nuestra contraseña. Si lo establecemos a *yes* aparecerán todas las carpetas de los `$HOME` de usuario, aunque no podríamos acceder a ellas salvo que la validación sea la correcta (nombre de usuario y contraseña)

**writable** permite que un usuario pueda crear y modificar archivos en su directorio `$HOME` cuando inicia una conexión SAMBA. Podemos conseguir esto mismo sustituyendo esta línea por

```
read only = no
write ok = yes
```

**valid users** lista de usuarios que pueden conectarse a un recurso (en este caso, la variable `%S` contiene el nombre actual del recurso).

**create mode** determinamos que los permisos de archivo para todos los archivos creados en el directorio compartido sean 0664.

**directory mode** los permisos para todos los directorios creados en el recurso compartido serán 0775.

### Sección [printers]

Con SAMBA podemos configurar de dos formas distintas la forma en que nuestras impresoras están disponibles para la red:

1. Creando una sección específica para compartir en `/etc/printcap` para cada impresora que se quiera compartir. No es el método que vamos a usar.
2. Usar la sección especial `[printers]` para compartir todas las impresoras definidas en el archivo `/etc/printcap`.

Con esta sección definimos cómo se controlan los servicios de impresión en el caso de que no haya entradas específicas en el archivo de configuración de SAMBA: si disponemos de un servicio, SAMBA buscará en primer lugar un servicio con ese nombre, si no hay ninguno con ese nombre se usa esta sección para permitir al usuario conectarse con cualquier impresora definida en `/etc/printcap`.

Con la sección `[printers]` que se lista a continuación damos a compartir todas las impresoras del sistema y además permitimos imprimir a cualquiera, excepto a la cuenta *guest*.

```
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
# Set public = yes para permitir al usuario guest imprimir
# es equivalente a guest ok = yes
guest ok = no
writable = no
printable = yes
```

Los parámetros `comment` y `browseable` ya se han visto en la sección `[homes]`.

Con el parámetro `path` determinamos en qué directorio temporal<sup>21</sup> se copian los archivos antes de imprimirlos, en este caso `/var/spool/samba`.

Con el parámetro `printable` en *yes* permitimos la escritura en el fichero de *spool* de impresión (si los permisos lo permiten). Con el parámetro `writable` a no restringimos a que sólo se permita la escritura de trabajos de impresión en el directorio de *spool* definido con el parámetro `path`. Es decir, definimos el servicio como no “escribible” pero sí como “imprimible”.

<sup>21</sup>Los permisos de este directorio están configurados para permitir la lectura y la escritura, lo mismo que el directorio `/tmp`.  
`drwxrwxrwt 2 root root 4096 jun 9 20:05 /var/spool/samba`



### Secciones personalizadas

Usando las secciones personalizadas podemos compartir impresoras o directorios de una manera no genérica. La idea que subyace en estas secciones consiste en poder compartir directorios para grupos de usuarios o permitir que determinados directorios sean de acceso público. A su vez, si tenemos varias impresoras conectadas a nuestra máquina y no queremos darlas todas a compartir, podemos usar esta sección para dar a compartir sólo una impresora en concreto.

Con el ejemplo que se lista a continuación permitimos acceso al servicio `trabajos` a los usuarios del grupo llamado `clase`:

```
[trabajos]
comment = Directorio compartido de la clase de informática
path = /home/trabajos
browseable = yes
writable = yes
printable = no
valid users = @clase
```

Las líneas de este ejemplo significan:

- damos a compartir el directorio `/home/trabajos`
- La identificación de este servicio es: `Directorio compartido de la clase de informática`
- Al estar `browseable` en `yes`, se mostrará la carpeta del recurso compartido en el explorador de Windows siempre que accedamos al servidor SMB.
- Permitimos que se pueda escribir en él.
- Con `printable = no` indicamos que no es un servicio de impresión.
- Con el parámetro `valid users = @clase` restringimos el acceso a este directorio a miembros del grupo `clase`.

Veamos cómo compartir una impresora dedicada en la que sólo puede imprimir el usuario `CURSOLINUX`:

```
[Impresora]
comment = Impresora a compartir
# con print ok = yes se consigue el mismo efecto
# que con la línea que sigue
printable= yes
printer = lp
path = /var/spool/samba
public = no
writable = no
valid users = cursolinux
```

Para finalizar, ¿por qué no dar a compartir nuestra unidad de CD a los usuarios `THALES` y `MILETO` de nuestra máquina:

```
[Cdrom]
path = /mnt/cdrom
writable = no
printable = no
valid users = thales, mileto
public = no
```

Cuando terminemos de configurar nuestro sistema podemos usar:

```
$ testparm
```

y comprobar que todo está perfectamente.

## 29.4.2. Swat

Vamos a comentar una herramienta que permite configurar Samba usando el navegador Web, se trata de SWAT (Samba Web Administration Tool). SWAT se incluye como parte del paquete estándar de Samba. La idea de este programa (hay más con esta misma filosofía) consiste en facilitar la configuración de un servidor Samba<sup>22</sup>

Antes de poder usar esta utilidad tenemos que configurar nuestro sistema para que permita acceder a ella, lo instalamos (CD3)

```
# rpm -ivh samba-swat-2.2.5-10.i386.rpm
```

y le decimos a xinetd que permita trabajar con él. Para eso en el fichero `/etc/xinetd.d/swat` sustituiremos la línea

```
disable = yes
```

por

```
disable = no
```

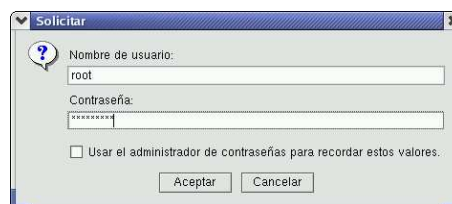
Después, tendremos que reiniciar xinetd para activar los cambios:

```
# /etc/rc.d/init.d/xinetd reload
```

Para usar el programa SWAT tendemos que iniciar un navegador y nos conectaremos a la URL

```
http://localhost:901/
```

Tras la ventana que nos pide los datos del root



accederemos a la página principal de la aplicación:



<sup>22</sup>Tiene una "pega", y es que cuando se usa escribe un archivo de configuración sin comentarios.

Los iconos de la parte superior de la pantalla nos permiten acceder a diferentes páginas de SWAT:



**HOME** página principal, en ella entre otras cosas tenemos enlaces a la documentación del paquete Samba. Entre otros el libro de O'Reilly (pero en Inglés) comentado en esta misma sección.



**GLOBALS** desde aquí podemos manipular la sección globals del archivo `etc/samba/smb.conf`. Podemos modificar los valores de los distintos parámetros, obtener ayuda sobre ellos y/o mantener el valor predeterminado. Para grabar los cambios pulsaremos sobre **Commit Changes**.



**SHARES** usando esta página podremos añadir, modificar o borrar recursos compartidos. Por defecto la pantalla inicial muestra sólo los parámetros de uso más frecuente del archivo `smb.conf`. Si pulsamos sobre **Advanced View** tendremos la posibilidad de configurar alguno de los parámetros menos usados.



**PRINTERS** para configurar las impresoras.



**STATUS** desde aquí podemos comprobar el estado de Samba. Además de ver cómo están las “cosas” podemos arrancar y parar los demonios de Samba y ver las conexiones activas de nuestro servidor Samba.



**VIEW** para examinar el contenido del archivo `smb.conf`. Si deseamos ver todas las variables disponibles y sus valores hay que hacer pulsar en **Full View**.



**PASSWORD** con este enlace aparecerá la pantalla mediante la cual tendremos la posibilidad de cambiar cuentas de usuarios locales y cuentas del controlador del dominio primario.

## 29.5. A “bailar” la Samba

### 29.5.1. Acceder desde una máquina Linux a una Windows

Como es evidente sólo podremos acceder a aquellos recursos autorizados en la máquina Windows.

Si optamos sólo por usar esta posibilidad no necesitamos tener instalado el paquete “principal” de la aplicación<sup>23</sup>. Trabajando de esta forma Samba no comparte ningún recurso con otro sistema, se limita a acceder a los recursos compartidos en los servidores de recursos la red. Si trabajamos sólo como cliente no tenemos que tener activos los demonios `smbd` o `nmbd` aunque sí debemos ajustar a nuestro grupo de trabajo el archivo `smb.conf`, la única línea necesaria en ese fichero sería:

```
workgroup=THALES
```

Veamos algunas de las posibilidades de las que disponemos ya:

#### **smbclient**

Una de las utilidades más interesantes de las que acompañan a Samba es `smbclient`. Con él podemos acceder desde Linux a los recursos compartidos en máquinas Windows con métodos que incluirían FTP, NFS y los “comandos r”, como `rcp`.

`smbclient` nos permite disponer de un interfaz similar a un FTP, por tanto, es una utilidad cuyo objetivo es accesos temporales a un recurso compartido.<sup>24</sup>

Veamos algunos ejemplos sobre su uso. Partiremos de la base de que estamos conectados en red, que nuestra máquina Linux se denomina `SECRE` y que podemos acceder a varias máquinas con Win9x (máquinas `BAG` y `COMPA`).

<sup>23</sup>Sólo hay que instalar `samba-client` y `samba-common`.

<sup>24</sup>Si lo que deseamos es mantener una conexión “permanente” es mejor usar `smbmount`.

Antes de nada y como no sabemos qué máquinas están a nuestra disposición ejecutemos:

```
# nmblookup -d 2 thales
added interface ip=192.168.1.5 bcast=192.168.1.255 nmask=255.255.255.0
querying thales on 192.168.1.255
Got a positive name query response from 192.168.1.5 ( 192.168.1.5 )
Got a positive name query response from 192.168.1.11 ( 192.168.1.11 )
Got a positive name query response from 192.168.1.12 ( 192.168.1.12 )
192.168.1.5 thales<00>
192.168.1.11 thales<00>
192.168.1.12 thales<00>
```

Vemos qué máquinas del grupo de trabajo especificado responden positivamente

Vemos que la máquina de IP 192.168.1.11 está encendida. Veamos qué recursos tiene compartidos. Para eso usaremos el comando `smbclient` con el parámetro `-L`. Por ejemplo, si nuestra máquina Windows se llama bag una posible salida es:

```
$ smbclient -L bag
added interface ip=192.168.1.5 bcast=192.168.1.255 nmask=255.255.255.0
```

Password:

Sharename	Type	Comment
-----	----	-----
ERASE	Disk	
ADMIN\$	Disk	
PRINTER\$	Disk	
EPSON	Printer	
D	Disk	
C	Disk	
IPC\$	IPC	Comunicación remota entre procesos

Server	Comment
-----	-----
Workgroup	Master
-----	-----

nos muestra una lista con los recursos disponibles, las máquinas que comparten recursos y los grupos de trabajo. Comentar que la máquina Linux verá a la máquina Windows aunque el nombre del grupo de trabajo no sea el mismo.

Si usamos:

```
$ smbclient -L secre -U cursolinux
added interface ip=192.168.1.5 bcast=192.168.1.255 nmask=255.255.255.0
```

```
Anonymous login successful
Domain=[THALES] OS=[Unix] Server=[Samba 2.2.5]
```

Sharename	Type	Comment
-----	----	-----
IPC\$	IPC	IPC Service (Samba Server)

Server	Comment
-----	-----



SECRE	Samba Server
Workgroup	Master
-----	-----
THALES	SECRE

veremos la lista de los recursos compartidos disponibles en el servidor SMB SECRE para el usuario *cursorlinux*.

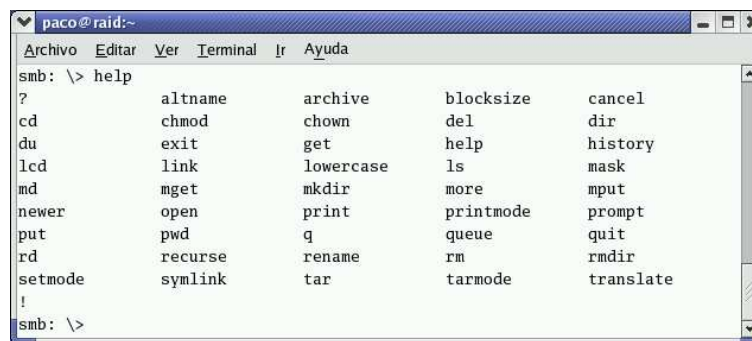
De la salida de `smbclient -L bag` hemos podido comprobar qué recursos compartidos tenemos a nuestra disposición en esa máquina, accedamos a alguno de ellos, para eso escribimos:

```
$ smbclient //bag/c
```

si deseamos acceder al disco C. Tras introducir la contraseña veremos:

```
smb: \>
```

Aquí podemos introducir comandos, para saber cuáles tenemos disponibles podemos usar la orden `help`<sup>25</sup>



Para ampliar sobre el significado de cada uno de ellos sólo debemos usar de nuevo `help` seguido del comando del que queremos ayuda:

```
smb: \>help get
```

```
HELP get:
```

```
<remote name>[local name] get a file
```

Comentemos algunos de los más usuales:

**? [comando]** muestra ayuda sobre ese comando o lista de comandos

**help [comando]** igual que ?

**! [comando de shell]** ejecuta un comando de la shell

**cd [directorío]** cambia el directorío remoto

**lcd [directorío]** cambia el directorío local

**md [directorío]** crea un directorío

**mkdir [directorío]** igual que md

**rd [directorío]** borra el directorío

**rmdir [directorío]** igual que rm

**del [archivos]** borra el archivo

**dir [archivos]** lista los archivos

**ls [archivos]** igual que dir

**get [rarchivo] [larchivo]** copia el archivo remoto (*rarchivo*) en el archivo de nombre *larchivo*

**mget [archivos]** copia los archivos que que coinciden con el nombre especificado (normalmente se usan comodines)

**newer [archivo]** sólo tomará los archivos posteriores al especificado.

**put [larchivo] [rarchivo]** copia desde la máquina local el archivo *larchivo* en la máquina remota con el nombre *rarchive*

<sup>25</sup>Son muy parecidos a los del ftp.

**mput [archivos]** copia en el servidor los archivos de la máquina local que coinciden con el nombre especificado (normalmente se usan comodines)

**printmode [modo]** determina el modo de impresión (text o graphics)

**print [archivo]** imprime el archivo especificado en la

máquina remota

**queue** muestra la cola de impresión.

**exit** sale del programa

**quit** igual que exit

Por ejemplo, si tras hacer un listado de la máquina remota:

```
smb: \>ls
```

comprobamos que hay un archivo de nombre `curso.txt`, podemos bajarlo a la máquina local usando:

```
smb: \>get curso.txt
```

### smbmount y smbmount

Si deseamos tener posibilidad de montar un sistema de archivos compartido en el árbol del sistema de archivos de Linux tenemos que usar `smbmount`<sup>26</sup>

Para montar algún recurso de la máquina Windows usaremos el comando:

```
# smbmount //máquina_windows/recurso destino_montaje
```

Veamos su utilidad con un ejemplo. Recordemos que en la máquina BAG disponíamos del recurso C, montémoslo en nuestro sistema de archivos. Para eso creemos un directorio destino de montaje:

```
# mkdir /mnt/bag
```

ya sí, usemos ahora

```
# smbmount //bag/C /mnt/bag
```

Password:

Al usar este comando, estamos consiguiendo que el recurso compartido sea montado en `/mnt/bag` y que lo veamos como cualquier otra parte del sistema de archivos Linux.



- Podemos acceder a máquinas Windows sin necesidad de usar el paquete Samba. Esto es posible gracias al soporte del sistema de archivos `smbfs` del núcleo. Si nuestro núcleo está compilado con esta opción (los de RedHat lo están) podemos montar recursos compartidos de máquinas Windows sin tener activos los demonios del paquete Samba y sólo necesitamos el programa

<sup>26</sup>Tanto `smbmount` como `smbumount` las podrán usar los usuarios “normales” si les activamos el bit `setuid`. Para ampliar sobre su uso os remitimos a las manpages de ambos programas.

smbmount. Si optamos por esta opción, para montar el recurso C de la máquina BAG escribiremos:

```
#mount -t smbfs //bag/c /mnt/bag
```

- Debemos tener cuidado con que los archivos de texto de Linux y de Windows son diferentes. Si creamos archivos de texto en un recurso montado vía SAMBA desde Linux, éste utilizará el formato de fin de línea del sistema Linux y después tendremos problemas para leerlos desde windows.

Para desmontarlo usaremos el comando smbmount

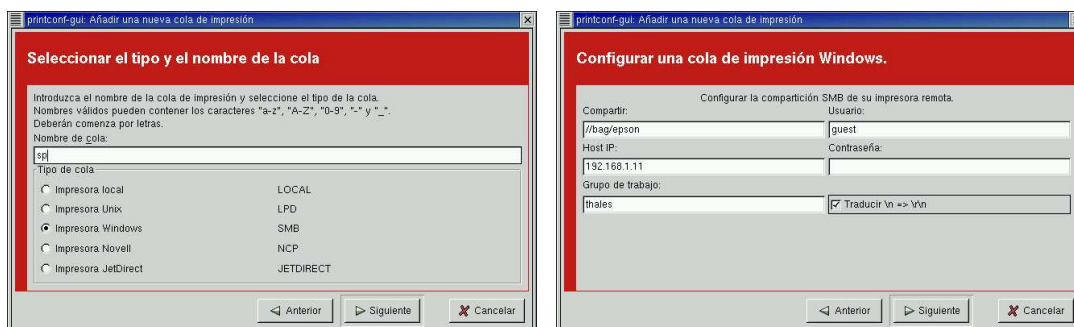
```
#smbmount /mnt/bag
```

o sólo umount

```
# umount /mnt/bag
```

**Agregar una impresora.** Para poder imprimir vía red usando una impresora conectada a una máquina Windows podemos usar `redhat-config-printer` (notar que previo debe de estar dada a compartir). Supongamos que disponemos de la impresora EPSON en la máquina BAG, añadámosla a Linux:

```
# redhat-config-printer
```



Optaremos por impresora Windows, y después tendremos que completar:

**Compartir** nombre del recurso compartido en windows, por ejemplo `//bag/epson`

**Usuario** `guest` u otro de la máquina Windows

**Host IP** IP de la máquina windows (192.168.1.11)

**Contraseña** Si es necesario se pone, si no nada de nada.

Por último marcar `\n=\r\n` para no tener problemas con los finales de línea. Eso sí, tendremos que seleccionar el filtro adecuado.

### 29.5.2. Acceder desde Windows a la máquina Linux

Un papel más interesante para Linux y Samba en una red Windows es el de servidor de recursos. Para conseguir esto debemos tener instalado el paquete `samba-2.2.5-10.i386.rpm` y tener correctamente configurado el fichero `/etc/samba/smb.conf`<sup>27</sup>.

<sup>27</sup>De nuevo recordar que si nuestra máquina dispone de las últimas versiones de Windows hay que descomentar las líneas:

```
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd
```

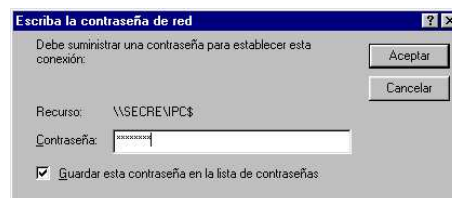
Para acceder (sin hacer más cambios que los ya comentados<sup>28</sup>) desde una máquina Windows a una Linux sólo lo podremos hacer si somos usuarios registrados en el sistema Linux. Además, en Linux debemos dar de alta a ese usuario y definir con qué contraseña puede acceder a nuestro sistema. Si queremos dar de alta al usuario *cursolinux* de la máquina Linux (con igual nombre en la máquina Windows<sup>29</sup>) usaremos:

```
# smbadduser cursolinux:cursolinux
```

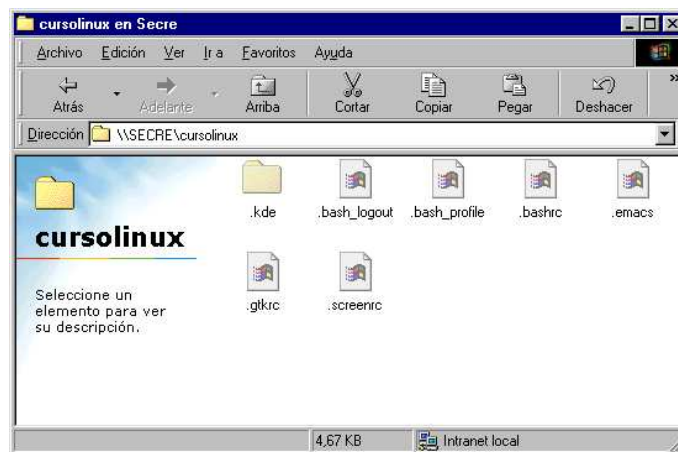
```
-----  
ENTER password for cursolinux  
New SMB password:  
Retype new SMB password:  
Added user cursolinux.
```

El único comentario a hacer sobre esta orden es que tenemos que separar con : la identidad del usuario en la máquina Linux y en la máquina windows. Es decir, nuestro usuario tendrá el mismo nombre en los dos sistemas y en este caso será *cursolinux*. Podremos cambiar la contraseña después usando el comando `smbpasswd`.

En la máquina Windows tendremos que entrar como ese usuario y si en el inicio no ponemos la contraseña, ésta se nos pedirá cuando intentemos acceder a los servicios de Red:



Una vez validada la contraseña (interesa desmarcar la casilla de guardar contraseña ya que si no, nos arriesgamos a que nos fastidien el Linux desde Windows), los directorios a los que tengamos acceso en nuestro sistema Linux se nos mostrarán como carpetas de Windows y podremos trabajar con ellas de la forma habitual.



<sup>28</sup>Puede haber también directorios públicos, para los cuales no hace falta que el usuario esté dado de alta como usuario del linux. A los usuarios registrados en Linux además les muestra su directorio `home`.

También, si existe un NT que valida los usuarios en la red, se puede poner ese NT como *password server* y es el que valida los usuarios.

<sup>29</sup>Si los nombres son distintos usaremos:

```
smbadduser nombre_linux:nombre_windows
```

Con respecto a las impresoras comentar que accederemos a ellas igual que una red Windows.

Es importante resaltar que si el grupo de trabajo no está bien configurado en el fichero `/etc/smb.conf` no veremos a la máquina Linux cuando accedamos a la red.

O sea que: **se puede montar una unidad de Red en Windows usando Linux, instalar programas sobre esa unidad, cortar, copiar, pegar, imprimir, etc sin que el cliente Windows se entere; usando la seguridad, precio y potencia de un Linux.**

#### ➡ Para practicar

Vamos a partir de que en la máquina Linux hay un usuario genérico de nombre INVITADO, ajustarlo a vuestro caso particular.

1. Instalar el servidor samba y conseguir que:
  - a) El usuario INVITADO pueda acceder desde Windows al equipo Linux.
  - b) El acceso esté limitado a la red local.
  - c) Haya un recurso compartido público de sólo lectura (`/home/samba`) que sea accesible por todos los equipos de la red. Para eso:

```
# mkdir /home/samba
# chmod 755 /home/samba
y en /etc/samba/smb.conf:
security=share
y además
[public]
comment = Directorio Público
path = /home/samba
public =yes
writable = no
printable = no
```
2. Construir una sección personalizada de nombre `[Practicas]` para el fichero `smb.conf` de manera que el directorio `/home/practicas` sea un recurso compartido de sólo lectura para el grupo `clase`. Además, deberá poder verse (la carpeta que da acceso a él) desde el navegador de windows por todos los usuarios del sistema si bien no podrán acceder a él.

## Capítulo 30

# Manteniendo nuestro sistema seguro

Las computadoras son mucho más útiles cuando están conectadas a una red, pero, desgraciadamente, estas redes hacen que las computadoras estén expuestas a un gran número de accesos no autorizados, y los sistemas Linux no son inmunes a este tipo de actividades. (*Hackers en Linux*, BRIAN HATCH y otros)

### 30.1. Modelos de seguridad.

¿Qué es más seguro, un sistema windows o un sistema Linux?

Esta respuesta, dependiendo de la persona que la responda tendrá una respuesta diferente. Pongamos un ejemplo para visualizarlo. Un sistema windows se basa en un sistema de código fuente propietario, que por cuestiones económicas (de la empresa propietaria), no está disponible para su revisión. Es decir, se comporta como una caja negra y totalmente opaca a nuestros ojos. Linux, y en general los programas de código fuente abierto, tienen disponible su código para que cualquiera lo pueda revisar. Es decir, se comportaría como una caja transparente, de la que siempre podemos ver su interior.

Con respecto a la seguridad, podemos visualizar, a modo de ejemplo, un fallo de seguridad como una bomba en el interior de la caja que es el código fuente del sistema. En un sistema de código abierto podemos ver, en la caja transparente, que existe una bomba dentro de la caja y desactivarla, es decir, arreglar el problema de seguridad.

En el modelo de código cerrado, esa bomba puede existir en el interior durante años y años sin ser detectada. O ser descubierta por alguien y utilizada en su propio beneficio. Y estamos confiando procesos críticos de aeropuertos u hospitales a un sistema que no podemos ver, ni decir si es realmente seguro.

A la tarea de descubrir fallos de seguridad se dedican tanto los “chicos buenos” como los “chicos malos”. A un sistema de código abierto se le achaca que los chicos malos pueden detectar fallos de programación basándose en el conocimiento del código fuente. Esto es verdad, y precisamente es este modelo de detección de problemas de seguridad y arreglo de dichos problemas de una forma mucho más rápida, lo que permite que los sistemas de código abierto sean más seguros.

Existen miles de programadores que pueden aportar una solución a ese fallo de seguridad. En un sistema de código cerrado, sólo nos queda esperar a que la empresa dedique a unos cuantos de programadores (si la planificación de la siguiente actualización del sistema o la cuenta de resultados lo permite) a arreglar el problema y saque un parche o solución para éste.

### 30.2. Amenazas a la seguridad de nuestro servidor.

Veamos algunas amenazas frecuentes a la seguridad de nuestros sistemas y cómo combatir contra ellas.



### Servicios no usados y Puertos abiertos

Al instalar el sistema operativo, hay aplicaciones que se incluyen en el tipo de instalación que seleccionamos (estación de trabajo, servidor...) y que normalmente no necesitamos. Si estas aplicaciones se encuentran activadas por defecto, nos encontramos con que como mínimo están consumiendo unos recursos de nuestra máquina (memoria, cpu...) que nos vendrían muy bien si estuvieran libres y además, son una posible puerta de entrada a nuestro sistema. Debemos asegurarnos de desactivar estos servicios no usados. Mediante la utilidad `ntsysv` podemos ver cuáles se encuentran activos y desactivarlos en caso necesario.

### Servicios no actualizados

Las aplicaciones que se incluyen en las distintas distribuciones, suelen tener incorporados los últimos parches de seguridad. Pero eso se refiere a la fecha de cierre de la distribución. Con el tiempo, aparecen nuevos modos de ataque que hacen que ese sistema que era considerado seguro, se convierta en inseguro.

Cuando se encuentran nuevos fallos de seguridad, se publican en los sitios web dedicados a la seguridad, como <http://www.securityfocus.com> o el *Computer Emergency Response Team* (CERT <http://www.cert.org>) tanto la descripción del problema como el parche correspondiente o el modo de hacer que nuestro sistema no sea vulnerable. Es un mecanismo efectivo de alertar a la comunidad de estos nuevos problemas, pero queda bajo la responsabilidad del administrador el aplicar las medidas oportunas.

En un sistema RedHat, podemos revisar la lista de paquetes que han sido actualizados con respecto a los que se incorporaban a la distribución originalmente. Simplemente podemos ir a cualquiera de los sitios en los que se encuentran (<ftp://ftp.rediris.es/mirror/redhat/updates>, <ftp://ftp.cica.es/mirrors/Linux/redhat-updates> o <http://updates.redhat.com>) y descargárnoslos<sup>1</sup>.

Una vez que nos hemos descargado las nuevas versiones de los paquetes rpm, podemos utilizar el comando siguiente para actualizar nuestro sistema.

```
#rpm -Fhv *.rpm
```

La opción `-F` (de *Freshen*, refrescar) en este caso es útil porque actualiza los sistemas que se encontraban ya instalados en el sistema, pero no hace nada con los que no estuvieran instalados anteriormente.

### Servicios inherentemente inseguros

Incluso el administrador más cauto puede caer víctima de los fallos de seguridad si los servicios que se eligen son inherentemente inseguros. Por ejemplo, el servicio telnet, que hace pasar en claro los nombres de usuario y las contraseñas por la red, puede ser adecuado para una red interna sin conexión a Internet<sup>2</sup>, pero su utilización a través de Internet puede ser un suicidio.

Afortunadamente, las distribuciones actuales se encuentran preparadas para utilizar los protocolos seguros (https, ssh...) e ir olvidando los protocolos inseguros como el ftp, telnet...

### Malas palabras de paso

Si las palabras de paso de nuestros usuarios (y de nosotros mismos) son débiles, es decir, fácilmente deducibles, no tenemos políticas de cambio de palabras de paso, o no se limita el número de reintentos fallidos, tenemos la puerta principal de nuestra casa entreabierta para que intente entrar cualquiera.

### Monitorización de nuestra máquina

Consiste en comprobar que los ficheros de nuestro sistema no han sido manipulados por ningún intruso o usuario malintencionado, que puede haber puesto puertas traseras o caballos de troya para entrar en él.

Existen varias cosas que podemos hacer en este aspecto.

<sup>1</sup>El servicio RedHat Network pretende hacer esto de manera automática.

<sup>2</sup>Incluso ahí es cuestionado su uso, porque el enemigo puede estar dentro de casa.



**Controlar permisos de ficheros.** Mediante el comando `find` podemos hacer varias comprobaciones. Con la orden siguiente, comprobamos los ficheros que tienen SUID o SGID activado. Debemos tomar nota de cuáles son los que el sistema tiene y estar atentos a los que puedan incorporarse. Especialmente peligrosos serían los que tienen como propietario o grupo a `root` porque se ejecutarán con los permisos de superusuario.

```
#find / -type f -perm +6000 -ls
```

Mediante la utilidad `rpm` podemos comprobar la integridad de nuestros ficheros. Veámoslo con un ejemplo. La opción `-V` del comando `rpm` verifica la integridad de los ficheros respecto a su estado inicial, que se guarda en la base de datos de paquetes `rpm`.

Comprobamos a qué paquete pertenece el fichero `/bin/false`

```
#rpm -q -f /bin/false
coreutils-4.5.3-19
```

Hacemos una copia del fichero para devolverlo posteriormente a su estado original

```
[root@siridus bin]# mv false false.ori
```

Copiamos otro fichero en lugar del fichero original (`/bin/false`)

```
[root@siridus bin]# cp login false
```

Ejecutamos el comando `rpm` en modo de verificación de integridad. La salida detecta que el fichero `/bin/false` ha sido modificado en tamaño (S), contenido (*hash MD5*) y fecha (T).

```
[root@siridus bin]# rpm -Va coreutils
S.5....T /bin/false
```

Devolvemos todo a su situación original

```
[root@siridus bin]#mv false.ori false
```

Otras utilidades que simplemente comentamos son `Logwatch` y `tripwire`.

**Logwatch** es un programa que analiza los logs o trazas del sistema y nos los presenta de forma más resumida, mostrándonos lo más importante desde el punto de vista de la seguridad.

**Tripwire** es un programa que comprueba la integridad de los ficheros de nuestro sistema<sup>3</sup>. Funciona de la siguiente manera. Cuando lo instalamos, crea una base de datos segura y cifrada con las firmas<sup>4</sup> de los ficheros de nuestro sistema. Cada día, se comprueban los ficheros con sus firmas y alerta de las modificaciones que se puedan haber producido.

## Monitorización de red

Veremos el caso más básico de monitorización de red, que consiste en comprobar los puertos que se encuentran abiertos en las máquinas de nuestra red.

Para ello utilizaremos la utilidad `nmap` que es un escáner de puertos que se instala por defecto en el modo servidor de RedHat<sup>5</sup>. Nos dice qué puertos se encuentran abiertos en las máquinas.

Para usar `nmap` en su modo por defecto tecleamos:

```
[root@siridus bin]# nmap -O localhost
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on localhost (127.0.0.1):
(The 1598 ports scanned but not shown below are in state: closed)
Port State Service
22/tcp open  ssh
111/tcp open  sunrpc
6000/tcp open  X11
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 0.162 days (since Tue May 20 19:14:28 2003)
Nmap run completed -- 1 IP address (1 host up) scanned in 7 seconds
```

<sup>3</sup>Parecido a lo que hemos visto mediante el comando `rpm` en modo verificación de integridad, pero más potente.

<sup>4</sup>La firma de un fichero consiste en aplicarle una función matemática de hash que consigue una marca del fichero, mucho más pequeña que éste, pero que al modificar el fichero, aunque sea levemente, el hash difiere sustancialmente del original.

<sup>5</sup>Si no se encuentra instalado, sólo tenemos que dirigirnos al CD2 e instalar `nmap-3.00-1.i386.rpm`





Vemos que los puertos abiertos en nuestra máquina son el 22 (ssh), 111 (sunrpc) y 6000 (X11). Si observamos algún puerto innecesario debemos desactivarlo. La opción `-O` intenta detectar el sistema operativo de la máquina objetivo.

Las opciones del comando son las siguientes:

`nmap -s` (tipo de escaneo) `-p` (rango de puertos) objetivo

Los tipos de escaneo pueden ser:

**T:** TCP Connect

**S:** TCP SYN

**F:** TCP FIN

**N:** TCP NULL

**X:** TCP Xmas Tree

**U:** UDP Scan

**R:** RPC Scan

Debemos comprobar periódicamente con nmap los puertos que se encuentran activos en nuestras máquinas, desactivando los que no sean necesarios. Puertos extraños abiertos pueden indicar que el sistema ha sido crakeado.

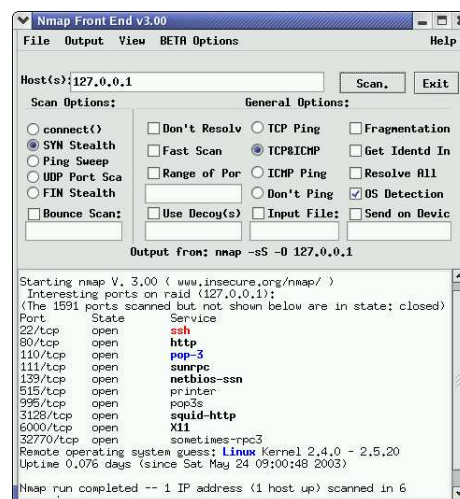
#### En modo gráfico:

Si instalamos el paquete del CD3

```
# rpm -ivh nmap-frontend-3.00-1.i386.rpm
```

dispondremos de un *front-end* para nmap, se ejecuta con

```
#xnmap
```



y para ver qué puertos tenemos abiertos sólo tenemos que indicar nuestra IP en el recuadro correspondiente (aparece por defecto) y pulsar sobre **scan**.

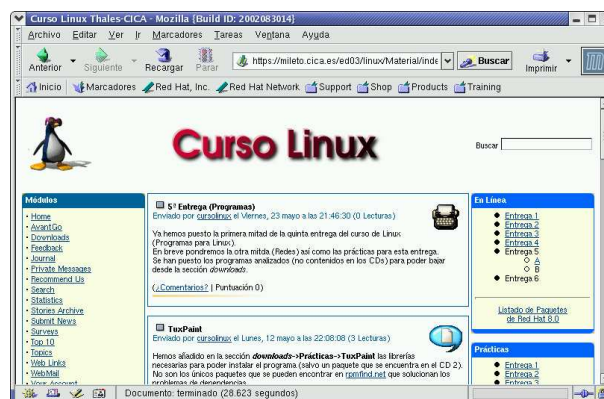
# Capítulo 31

## Apéndice.

Aunque PHP compite con ASP de Microsoft, Cold Fusion de Allaire, JSP de Sun, e incluso un primo de código fuente abierto llamado `mod_perl`, realmente se encuentra por encima de prácticamente todos sus competidores por al menos una año. (*Servidor Apache 2*, MOHAMMED J. KABIR)

### 31.1. PHP con MySQL: instalación de phpnuke

Como práctica final, vamos a ver la potencia de instalar MySQL (servidor de base de datos) y, junto con PHP y el software PHPNuke (<http://phpnuke.org/>), montar una web altamente configurable. El mejor ejemplo de qué es



la página Web del curso.

Para eso necesitamos:

1. Tener instalado Apache.
2. Tener instalado PHP
3. Instalar los paquetes (CD3) que permiten disponer de la base de datos MySQL<sup>1</sup>

```
mysql-3.23.52-3.i386.rpm
mysql-server-3.23.52-3.i386.rpm
mysql-devel-3.23.52-3.i386.rpm
# rpm -ivh mysql-*
error: Failed dependencies:
```

---

<sup>1</sup>El último no es necesario

```
perl-DBI is needed by mysql-3.23.52-3
perl-DBD-MySQL is needed by mysql-3.23.52-3
perl(DBI) is needed by mysql-3.23.52-3
Suggested resolutions:
perl-DBD-MySQL-2.1017-3.i386.rpm
perl-DBI-1.30-1.i386.rpm
```

Ambos paquetes están en el CD2, montémoslo. Si intentamos instalarlos nos aparecerá otro problema de dependencias que se resuelve con un paquete del mismo CD2<sup>2</sup>:

```
# rpm -ivh postgresql-libs-7.2.2-1.i386.rpm
# rpm -ivh perl-DBD-Pg-1.13-5.i386.rpm perl-DBI-1.30-1.i386.rpm
```

Tras solucionar los problemas de dependencias de los paquetes del CD2, montemos el CD3 y finalicemos la instalación<sup>3</sup> con:

```
# rpm -ivh perl-DBD-MySQL-2.1017-3.i386.rpm mysql-*
```

y activémosla:

```
# service mysqld start
```

Lo mejor es usar `ntsysv` para que se active en el arranque.



4. Instalar el módulo que permite a PHP disponer de soporte de base de datos MySQL (CD3)
 

```
# rpm -ivh php-mysql-4.2.2-8.0.5.i386.rpm
```
5. Descomprimir el PHPNuke<sup>4</sup> en `/var/www`<sup>5</sup> y una vez descomprimido (véase el fichero INSTALL), tras situarnos en el directorio `/var/www/sql` ejecutemos:
 

```
# mysqladmin create nuke
```

 para crear la base de datos nuke, y
 

```
# mysql nuke <nuke.sql
```

 para crear las tablas de esta base de datos según se establece en el fichero `nuke.sql`.
6. Cambiar los permisos del fichero `/var/www/html/ultramode.txt` a 666
7. Comprobar que todo está bien, apuntando con nuestro navegador a `http://127.0.0.1/index.php`

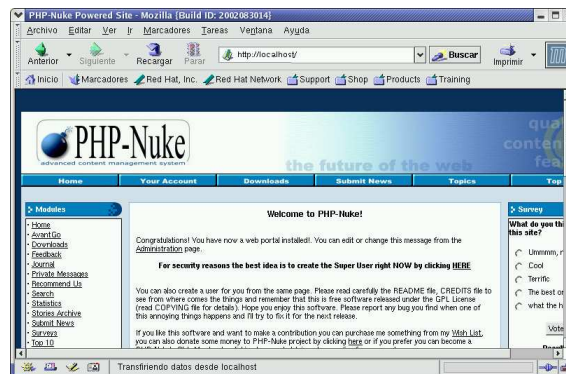
<sup>2</sup> postgresql-libs-7.2.2-1.i386.rpm

<sup>3</sup> Es importante instalar los paquetes a la vez, tal cual aparece en la línea, ya que presentan problemas de dependencias recíprocas.

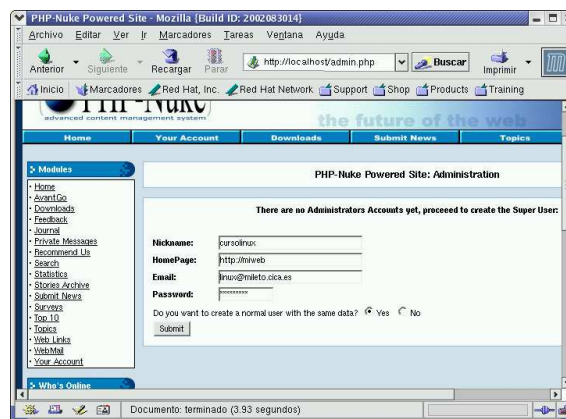
<sup>4</sup> [http://www.phpnuke.org/modules.php?op=modload&name=Downloads&d\\_op=viewdownload&cid=1](http://www.phpnuke.org/modules.php?op=modload&name=Downloads&d_op=viewdownload&cid=1)

<sup>5</sup> Esto es opcional pero así se queda ya puesto en su sitio:

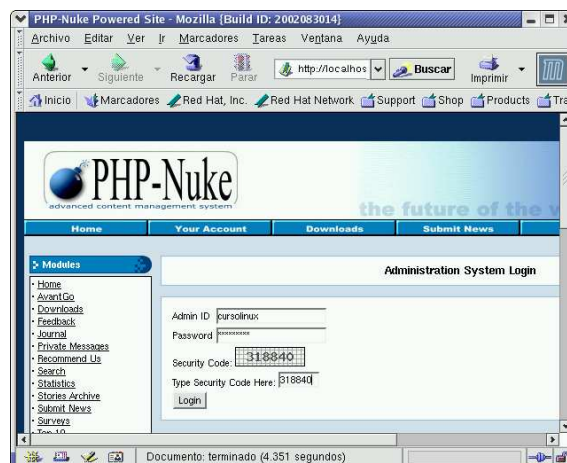
```
# cp PHP-Nuke-6.5.tar.gz /var/www
# cd /var/www
# tar -xzf PHP-Nuke-6.5.tar.gz
```



8. Lo último (el resto para casa: configurar, poner en castellano, etc), crear el superusuario del sistema usando `http://localhost/admin.php`



Una vez pasado el trámite, accedamos como administrador (es conveniente guardar a buen recaudo los datos del administrador y el código)



y listo, la configuración y mejora es cosa vuestra.

# **Parte VII**

## **Compilación del nucleo**

## Capítulo 32

# Introducción a C.

Afortunadamente, cuanta más gente se acerca a GNU/Linux, más se interesan por el Kernel. Pero leer las fuentes no es siempre suficiente. Puede ser fácil que entendamos el código, pero olvidemos los conceptos, la filosofía y decisiones de diseño que se encuentran tras este código. (*Índice de documentación para gente interesada en escribir y/o comprender el Kernel de Linux*, JUAN MARIANO DE GOYENECHE, <http://www.dit.upm.es/~jmseyas/linux/kernel/hackers-docs.html>)

En esta entrega aprenderemos a compilar y actualizar el núcleo de Linux y en qué ocasiones puede sernos útil. Para ello veremos primero algunos conceptos de programación necesarios para comprender mejor este proceso.

### 32.1. Lenguajes de Programación

A continuación comentamos algunas características de los principales lenguajes de programación que podemos utilizar con la distribución RedHat 8.0.

Entre las características de los lenguajes que comentaremos se encuentran: si son compilados o interpretados (lenguajes de script), si es Orientado a Objetos y su utilización principal.

**Perl:** Lenguaje de script, soporta orientación a objetos. Muy utilizado para scripts de administración de sistemas, programación web mediante cgi, muy potente en su manejo de cadenas de texto.

**Python:** Lenguaje de script, soporta orientación a objetos. Utilizado para prototipos, aplicaciones de script, web.

**TCL:** Lenguaje de script, no soporta orientación a objetos. Utilizado para aplicaciones de script. Sus extensiones para entorno gráfico se denominan TK. Normalmente aparecerán unidos como TCL/TK

**PHP:** Lenguaje embebido en páginas web, que se ejecuta en el servidor para originar páginas web dinámicas (como el ASP de Microsoft, pero mejor ;-). Muy utilizado en web y para conexión a bases de datos.

**Java:** Lenguaje orientado a objetos y cuya principal característica es la portabilidad, al generarse un código intermedio (bytecode) que se ejecutará en cualquier máquina virtual java, independientemente de la plataforma hardware. Ideal para aplicaciones que se distribuyen y ejecutan vía web. Promete ser el lenguaje universal de programación.

**Lisp:** Lenguaje de programación funcional, utilizado sobre todo en Inteligencia Artificial. Variantes suyas son Elisp, Clisp y Scheme.

**Fortran:** Lenguaje veterano donde los haya, fue muy utilizado para aplicaciones de cálculo matemático y científico. Sus versiones que aún se utilizan son f77 y f90/95

**C:** Es el lenguaje de la programación de sistemas por excelencia. Su eficiencia es muy elevada. El kernel de Linux en su mayor parte está programado en C.

**C++:** Lenguaje que partiendo de C, incorpora capacidad de orientación a objetos.

**Programación en shell:** Ya conocemos su versatilidad y potencia a la hora de alegrar la vida del administrador de sistemas y usuarios en general.

**Objetivo-C:** Variante de C, con extensiones de orientación a objetos.


**Pascal:** Lenguaje muy utilizado para la enseñanza de la programación. Se incluye como un conversor de pascal a C (p2c).

## 32.2. Introducción a C.

Al ser C el lenguaje en el que está construido en su mayor parte el núcleo del sistema operativo, le dedicaremos una mención especial.

El lenguaje de programación C, fue desarrollado por Brian Kernighan y Dennis Ritchie, con el propósito de escribir sistemas operativos (de hecho este sistema operativo era Unix) y debido a su portabilidad y eficiencia es uno de los más utilizados. La mayoría de los programas que utilizamos ya sea en plataformas Unix/Linux o Windows, están escritos en C. Como referencia, el propio kernel de Linux está escrito en su inmensa mayoría en C y sólo muy pequeñas partes que deben ser muy eficientes, están escritas en lenguaje ensamblador (el que habla la propia CPU de la máquina). Estas pequeñas partes son las que hay que cambiar cuando se desea construir un sistema Linux sobre otra arquitectura de máquina (como Intel, Sparc, Alpha, RISC, OS/390...). Si deseamos recompilar el kernel deberemos tener unos pequeños conocimientos de C y del entorno de compilación.

Aparte del núcleo de linux (kernel), gran cantidad de programas que queramos instalar en nuestro sistema vendrán en código fuente que la mayor parte de las veces será C. Por ello, vamos a ver una pequeña introducción a C. Lo haremos a través de ejemplos, empezando con un programa simple en este lenguaje de programación.

 **Nota:** Necesitamos disponer del compilador de C y las herramientas de apoyo necesarias. Para ello, debemos tener instalados los paquetes:

`cpp-3.2-7` → preprocesador del compilador, necesario para el paquete que sigue.

`gcc-3.2-7` → compilador de C

`glibc-kernheaders-2.4-7` → encabezados del kernel, necesario para instalar el paquete siguiente

`glibc-devel-2.2.93-5` → librerías y ficheros de *include*<sup>1</sup> para desarrollo

`make-3.79.1-14` → utilidad para comprobar dependencias en compilación.

Crearemos un directorio llamado `proyecto1`, en el que nos situaremos y realizaremos las prácticas.

```
$cd $HOME
$mkdir proyecto1
$cd proyecto1
```

Editamos el fichero `hola.c` con el contenido que viene a continuación. Debemos guardarlo en formato texto, no con códigos de algún procesador de textos. Para ello podemos utilizar el que nos resulte más cómodo.

---

<sup>1</sup>Que no se asuste quien no conozca todavía conceptos del lenguaje de programación C.

```
#include <stdlib.h>
#include <stdio.h>
int main (int argc, char *argv[])
{
    printf ("Hola Mundo!\n");
    return 3;
}
```

Este programa nos presentará por pantalla el mensaje "Hola Mundo!" (es una costumbre el hacer este programita cuando comenzamos con un lenguaje de programación).

Las dos primeras líneas le dicen que incluya funciones especificadas en los ficheros de cabecera (esos son los ficheros "include"), que en este caso son `stdlib.h` y `stdio.h`. Estos ficheros se encuentran en una ubicación estándar que es `/usr/include`. Podemos verlos con

```
$more /usr/include/stdio.h
```

Después se define una función que se llama `main`<sup>2</sup> y de la que podemos decir que devuelve un valor entero (`int`) y recibe dos argumentos: un entero con el nombre `argc` y un array (`[]`) de punteros (\*) a caracteres (`char`), llamado `argv`<sup>3</sup>. El contenido de la función se encuentra delimitado por llaves de apertura({) y cierre (}).

Una vez dentro del `main`, lo primero que hace es llamar a la función `printf`<sup>4</sup> con una cadena de texto ("Hola mundo!") terminada con un retorno de carro (`\n`). Esta función, como podemos imaginar, lo que hace es mostrar en pantalla la cadena indicada.

Además el programa nos devuelve el código 3 como resultado de su ejecución. Por convención, el código de retorno de un programa que se ha ejecutado correctamente es 0 (cero) y si es distinto de cero, nos indica que se ha producido algún error. Para este ejemplo nos hemos saltado la norma para que veamos el código de retorno más claramente.

Guardamos el fichero. A esto le llamamos *código fuente*, escrito en un lenguaje de alto nivel, que en este caso es C. Para poderlo ejecutar necesitamos compilarlo mediante el comando:

```
$ gcc -Wall -o hola hola.c
```

Este comando nos dice que compilemos (orden `gcc`) el código fuente `hola.c`, para generar el código ejecutable `hola` (`-o hola`) con la opción de que nos avise de todos los posibles errores o avisos (`-Wall`) que puedan ocurrir. Si no hemos puesto por ejemplo las comillas que cierran la cadena:

```
printf ("Hola Mundo!\n);
```

Nos saldrán mensajes como los siguientes,

```
hola.c:6: unterminated string or character constant
```

```
hola.c:6: possible realstart of unterminated constant
```

indicándonos que ha habido algún error y no se generará el fichero ejecutable `hola`. Si todo ha ido correctamente, terminará sin errores y podemos ver con `$ls -l` los ficheros:

```
-rwxr-xr-x 1 ed03linux ed03linux 11702 May 27 09:31 hola
-rw-r--r-- 1 ed03linux ed03linux 133 May 27 09:26 hola.c
```

Vemos que `hola` se ha creado con permisos de ejecución automáticamente. Si lo ejecutamos, mediante:

```
$/hola
Hola Mundo!
```

Vemos que la salida ha sido un simple pero gratificante mensaje. Si inmediatamente vemos el código de retorno mediante el comando

```
$echo $?
3
```

<sup>2</sup>Es la función principal, como su propio nombre indica, y a la que se llama directamente cuando se ejecuta un programa en C.

<sup>3</sup>No os asustéis, no es estrictamente necesario que comprendáis ésto.

<sup>4</sup>Que se encuentra definida en `/usr/include/stdio.h`, por eso la necesidad del `#include` con ese fichero.



Vemos que nos ha devuelto un tres (el valor que le dijimos que devolviese). Hay que señalar que si ejecutamos otra vez

```
$echo $?
0
```

ahora nos ha devuelto un 0, porque estamos preguntando por el valor de retorno del último comando ejecutado, que ahora es el comando `echo` anterior. Ya hemos compilado nuestro primer programa en C.

El proceso que hemos realizado, en realidad consta de dos fases. Una de compilación, que consistiría en comprobar que nuestro fichero es correcto léxica y sintácticamente; y otra de enlazado (o linkado) en la que unimos todas las partes que necesita el ejecutable, incluidas las funciones de librerías de las que hace uso.

Podemos hacer estas dos funciones de una en una, mediante los comandos:

```
$gcc -c hola.c
```

Con la opción `-c` le indicamos que solamente compile, pero no enlace. Nos generará el fichero `hola.o` que contiene el resultado de compilar el fichero `hola.c`, pero sin enlazar con las librerías que necesita.

```
$gcc -o hola hola.o
```

Con este comando decimos que coja el código objeto (que así se denomina el compilado, pero no enlazado) y que genere el ejecutable (éste ya sí estará enlazado) `hola`.

### 32.2.1. Librerías en C.

Como vimos antes, la última etapa de la construcción de un programa es el enlazado (linkado) para unir todas las piezas que necesita el ejecutable. Por economía de espacio y de recursos, las funciones que varios programas necesitan poder hacer (acceso a ficheros, funciones matemáticas...), se ubican en librerías y los programas lo que hacen es utilizar esas librerías<sup>5</sup>. Las librerías suelen encontrarse en `/lib` o `/usr/lib`.

Las librerías pueden ser *estáticas* o *dinámicas*:

- Cuando usamos una librería estática, en el momento del linkado se copian físicamente las partes de la librería que utiliza el programa dentro del ejecutable que se genera, siendo éste de un tamaño considerable.
- En las librerías dinámicas, lo que se hace es poner en el ejecutable un enlace o nota, diciendo que cuando se ejecute el programa, tiene que cargar las funciones de la librería correspondiente.

El uso de librerías dinámicas (también llamadas librerías compartidas) hace que los ejecutables sean de menor tamaño y en general consumen menos memoria. Además, si esta librería se encuentra ya en memoria, no hace falta cargarla de nuevo y puede ser compartida por todos los programas ejecutables que la utilizan.

Por defecto, Linux intenta enlazar con librerías dinámicas, y si no puede, con las versiones estáticas si existieran. De muchas librerías existen la versión estática y la dinámica en el sistema.

Las librerías estaticas suelen tener nombres del tipo `libnombre.a`<sup>6</sup>, mientras que las librerías dinámicas suelen denominarse `libnombre.so.x.y.z`<sup>7</sup>, donde `x.y.z` es el número de versión. Veamos un ejemplo:

```
$ ls -l /usr/lib/libusb*
-rw-r--r-- 1 root root 181640 ago 30 2002 /usr/lib/libusb.a
lrwxrwxrwx 1 root root 17 nov 3 2002 /usr/lib/libusb-0.1.so ->libusb-0.1.so.4.1.1
lrwxrwxrwx 1 root root 17 nov 3 2002 /usr/lib/libusb-0.1.so.4 ->libusb-0.1.so.4.1.1
-rwxr-xr-x 1 root root 133705 ago 30 2002 /usr/lib/libusb-0.1.so.4.1.1
```

<sup>5</sup>El término adecuado sería el de bibliotecas (del inglés *libraries*), pero el uso de librerías está ya demasiado extendido como para intentar cambiarlo.

<sup>6</sup>La extensión `.a` proviene de *archive*

<sup>7</sup>La extensión `.so` proviene de *shared object*, objeto compartido

La librería `libusb` existe en versión estática (`libusb.a`) y en dinámica (`libusb-0.1.so.4.1.1`). Además, existen dos enlaces a la librería dinámica, `libusb-0.1.so` y `libusb-0.1.so.4`.

Podemos ver qué librerías dinámicas utiliza nuestro programa *hola* mediante la orden:

```
$ldd hola
        libc.so.6 =>/lib/libc.so.6 (0x40022000)
        /lib/ld-linux.so.2 =>/lib/ld-linux.so.2 (0x40000000)
```

Vemos que utiliza la librería `libc.so.6` y la `ld-linux.so.2`

La librería estándar de C, donde se encuentran muchas de las funciones comunes de C (como el `printf` que utilizamos antes en nuestro programa `hola.c`) se encuentra en `/lib/libc.so.6`.

Para poder ver cuáles son las funciones que contiene una librería, podemos ejecutar los comandos:

```
nm /lib/libcrypto.so.2
```

En este comando nos ha mostrado todos los símbolos que define.

```
nm /lib/libcrypto.so.2 | grep ' T ' | cut -f3 -d' ' | grep -v '^_' | sort
-u | less
```

Este, mucho más elaborado y nos sirve para conocer la potencia de las tuberías en la shell, realiza lo siguiente:

1. como antes, muestra los símbolos
2. pero filtra para que salgan solamente los que tienen como atributo una T (que son los realmente definidos en la librería)
3. con la orden `cut` nos muestra el tercer campo (`-f3`) teniendo en cuenta que el separador es el espacio en blanco (`-d' '`)
4. nos omite (`-v`) los que empiezan por el carácter subrayado (`'^_'`)
5. los muestra ordenados alfabéticamente (`sort -u`)
6. y nos los muestra página a página (`less`)

No está mal para una orden.

En nuestro primer ejemplo hemos utilizado funciones de librerías ya existentes, pero también podemos crear nuestras propias librerías.

Creemos para ello dos ficheros con las funciones deseadas. El primero, llamado `simple_math_sqrt.c` y cuyo código es:

```
#include <stdlib.h>
#include <stdio.h>

static int abs_error (int a, int b)
{
    if (a > b)
        return a - b;
    return b - a;
}

int simple_math_isqrt (int x)
{
    int result;
    if (x < 0) {
        fprintf (stderr, "simple_math_sqrt: raiz cuadrada num. negativo\n");

        abort ();
    }
}
```



```
}    result = 2;
    while (abs_error (result * result, x) >1) {
        result = (x / result + result) / 2;
    }
    return result;
}
```

y el segundo, `simple_math_pow.c` con el código fuente siguiente:

```
#include <stdlib.h>
#include <stdio.h>

int simple_math_ipow (int x, int y)
{
    int result;
    if (x == 1 || y == 0)
        return 1;
    if (x == 0 && y <0) {
        fprintf (stderr, "simple_math_pow: elevando cero a potencia negativa\n");
        abort ();
    }
    if (y <0)
        return 0;
    result = 1;
    while (y >0) {
        result = result * x;
        y = y - 1;
    }
    return result;
}
```

Queremos llamar a la librería `simple_math`. Es considerado buena práctica nombrar las funciones de la librería como `simple_math_?????`, es decir, el nombre de la librería más el nombre de la función, separados por símbolos de subrayado.

Podemos compilar el código con:

```
$gcc -Wall -c simple_math_sqrt.c
$gcc -Wall -c simple_math_pow.c
```

La opción `-c` significa compilar solamente, no generar código ejecutable. Los ficheros generados son `simple_math_sqrt.o` y `simple_math_pow.o`. Este formato, como vimos anteriormente, se llama código objeto y es un paso previo al ejecutable.

Necesitamos archivar estos ficheros objeto en una librería, cosa que hacemos mediante:

```
$ar rc libsimple_math.a simple_math_sqrt.o simple_math_pow.o
$ranlib libsimple_math.a
```

El comando `ranlib` indexa el archivo para una más rápida recuperación de las funciones. Podemos comprobar lo que hay dentro de la librería mediante:

```
$nm libsimple_math.a
```

Ya podemos utilizar la librería. Creamos un fichero `test.c`:

```
#include <stdlib.h>
#include <stdio.h>

int main (int argc, char *argv[])
{
    printf ("%d\n", simple_math_ipow (4, 3));
    printf ("%d\n", simple_math_isqrt (64));
    return 0;
}
```

y ejecutamos:

```
$gcc -Wall -c test.c
$gcc -o test test.o -L. -lsimple_math
```

El primer comando compila el fichero `test.c` y genera el código objeto `test.o`. El código de las funciones `simple_math_ipow` y `simple_math_isqrt` no se encuentra en `test.c`, con lo que si de alguna forma compiláramos y convirtiéramos en ejecutable `test.c`, a la hora de ejecutar estas funciones no podría encontrarlas. Es precisamente el aviso que nos da al compilar.

El segundo comando realiza la función de enlazado (*linking*) y une el código de `test.o` con el de la librería `simple_math`. Ya sí que encontrará el código de las funciones. La opción `-L.` le dice que busque en el directorio actual (`.`) las librerías, además de unos sitios predefinidos como son `/lib` y `/usr/lib`. La opción `-lsimple_math` le dice que incluya la librería `libsimple_math.a`. El `lib` delante del nombre de la librería y la extensión (`.a`) se añaden automáticamente. Como no encuentra la librería dinámica coge la estática.

### 32.2.2. Trabajando con proyectos. Makefile.

¿Qué ocurre cuando hacemos un pequeño cambio a uno de los ficheros fuente de nuestra aplicación? Pues que debemos realizar todo el proceso de compilación y enlazado de nuevo si queremos tener la seguridad de estar trabajando con las últimas versiones de nuestros ficheros.

Una posible solución es poner todos los comandos en un fichero de script y así no tenemos que teclear demasiado. Pero aún así, si nuestro proyecto es de un tamaño mediano o grande, el tiempo que necesitará será elevado (y ello para muchas tareas que ya tiene hechas de antes, con el consiguiente desperdicio de tiempo de procesamiento y recursos de la máquina).

Se necesita una utilidad que recompila sólo los códigos objetos cuyos programas fuente hayan sido modificados, pero no todos los demás. Esta utilidad se llama `make`.

El programa `make` mira en un fichero del directorio actual llamado `Makefile`<sup>8</sup> y se encarga de realizar las acciones oportunas para satisfacer las dependencias. Los ficheros `Makefile` contienen una lista de reglas y dependencias que dicen cómo construir los programas<sup>9</sup>.

Dentro de un fichero `Makefile` hay que especificar una lista de dependencias del estilo "esto depende de esto otro" que hacen que el programa `make` funcione y los comandos necesarios para conseguir el fin (llamado objetivo) requerido. Nuestro objetivo final en el proceso del ejemplo anterior era que el programa ejecutable `test` dependiera<sup>10</sup> de la librería `libsimple_math.a` y el fichero objeto (`test.o`). ¿Cómo le decimos esto a `make`? Pues creamos un fichero llamado `Makefile` con las líneas siguientes<sup>11</sup>:

```
test: libsimple_math.a test.o
    gcc -Wall -o $@ test.o -L. -lsimple_math
```

<sup>8</sup> Aunque podemos especificar otro nombre, éste es el valor por defecto y nos referiremos a ellos con este nombre: `Makefile`

<sup>9</sup> Y en general cualquier otra cosa que tenga dependencias

<sup>10</sup> Es decir, se regenera cuando cambien los ficheros de los que depende

<sup>11</sup> El carácter que debemos poner delante de `gcc` es un tabulador, y no 8 espacios.



Esto significa: los ficheros `libsimple_math.a` y `test.o` deben existir y "estar puestos al día" antes que el ejecutable `test`. En este caso `test` se denomina objetivo (*target*).

En la segunda línea decimos cómo construir `test` cuando han cambiado `libsimple_math.a` o `test.o`. Cuando las dependencias `libsimple_math.a` y `test.o` están actualizadas, pero `test` no está actualizado, sabemos cómo actualizarlo: enlazando la librería con `test.o`<sup>12</sup>. Los caracteres `$$` se refieren al objetivo, que en este caso es `test`.

La siguiente dependencia que vamos a estudiar es que `libsimple_math.a` depende de `simple_math_sqrt.o` y `simple_math_pow.o`. En las líneas siguientes indicamos la dependencia y los comandos que necesitamos ejecutar para poner al día el objetivo<sup>13</sup>.

```
libsimple_math.a: simple_math_sqrt.o simple_math_pow.o
    rm -f $$
    ar rc $$ simple_math_sqrt.o simple_math_pow.o
    ranlib $$
```

Por último, los ficheros objeto `simple_math_sqrt.o` y `simple_math_pow.o` dependen de los ficheros fuente `simple_math_sqrt.c` y `simple_math_pow.c` respectivamente.

```
simple_math_sqrt.o: simple_math_sqrt.c
    gcc -c -o simple_math_sqrt.o simple_math_sqrt.c
simple_math_pow.o: simple_math_pow.c
    gcc -c -o simple_math_pow.o simple_math_pow.c
```

Aquí podemos ver claramente el sentido de "estar actualizado". El fichero `simple_math_sqrt.o` depende de `simple_math_sqrt.c`. Necesitaremos "actualizarlo" bien cuando no exista el objetivo (primera vez que lo ejecutemos) o cuando hayamos modificado `simple_math_sqrt.c` para hacer algún cambio (la fecha de última actualización es un valor que se guarda en Linux junto a cada fichero). Si la fecha de última actualización de `simple_math_sqrt.c` es posterior a la fecha de última actualización de `simple_math_sqrt.o`, entonces debemos actualizarlo. Existe un modo más conciso de poner estas dos reglas, útil cuando tenemos muchos ficheros fuente en C.

```
.c.o:
    gcc -Wall -c -o $*.o $<
```

Esto significa que cualquier fichero `.o` (fichero objeto) puede ser construido a partir de un fichero `.c` (fichero fuente) con el mismo nombre usando el comando

```
gcc -Wall -c -o $*.o $<
```

En esta línea, `$*.o` (`$*` es la base del nombre del fichero, sin la extensión) se expande como el nombre del fichero objeto y `$<` significa el fichero del cual depende `$*.o`, para cada fichero en particular.

Los ficheros `Makefile` pueden tener sus reglas en cualquier orden, pero es preferible poner las reglas principales al principio.

Existe una regla que siempre deberíamos poner: la regla *all*

```
all: libsimple_math.a test
```

El objetivo *all* (todo) es la regla que `make` trata de satisfacer por defecto. Es decir, si tecleamos `$make`, el comando automáticamente cogerá el fichero `Makefile` en el directorio en el que nos encontramos e intentará satisfacer el objetivo *all*. En nuestro caso, intentará satisfacer `libsimple_math.a`

<sup>12</sup>Es decir, el comando `gcc -Wall -o $$ test.o -L. -lsimple_math`

<sup>13</sup>Recordemos que delante de las líneas con `rm`, `ar` y `ranlib` hay un tabulador



y test, como objetivos fundamentales de los que depende *all*. Nuestro fichero Makefile finalmente, haciendo algunas optimizaciones, quedaría como:

```
# Los comentarios comienzan con el carácter(#) como en los scripts.
# Makefile para construir libsimple_math.a y test.

OBJS=simple_math_sqrt.o simple_math_pow.o
LIBNAME=simple_math
CFLAGS=-Wall
all: test
test: libsimple_math.a test.o
    gcc ${CFLAGS} -o $@ test.o -L. -l${LIBNAME}
libsimple_math.a: ${OBJS}
    rm -f $@
    ar rc $@ ${OBJS}
    ranlib $@
.c.o:
    gcc ${CFLAGS} -c -o $.o $<
limpiar:
    rm -f *.o *.a test
```

Posee algunas características más, pero en principio para el nivel del curso no necesitamos saberlas.

Si tecleamos `$make`, se intentarán satisfacer las dependencias del objetivo *all*, pero sólo realizando las acciones necesarias. Lo que ya esté puesto al día no será necesario repetirlo.

```
$make
gcc -Wall -c -o simple_math_sqrt.o simple_math_sqrt.c
gcc -Wall -c -o simple_math_pow.o simple_math_pow.c
rm -f libsimple_math.a
ar rc libsimple_math.a simple_math_sqrt.o simple_math_pow.o
ranlib libsimple_math.a
gcc -Wall -c -o test.o test.c
test.c: In function 'main':
test.c:11: warning: implicit declaration of function 'simple_math_ipow'

test.c:13: warning: implicit declaration of function 'simple_math_isqrt'

gcc -Wall -o test test.o -L. -lsimple_math
```

Si lo realizamos otra vez, observamos que ya está todo hecho y así nos lo comunica make.

```
[proyecto1]$ make
make: No se hace nada para 'all'.
```

Si borramos solamente el fichero test y ejecutamos el comando make, solamente hará lo necesario, que es regenerar el fichero ejecutable test.

```
$make
gcc -Wall -o test test.o -L. -lsimple_math
```

Hay una regla desconectada del objetivo principal (*all*), que es limpiar. Este objetivo se puede llamar explícitamente mediante:

```
$make limpiar
```

con el resultado de borrar los códigos objeto, la librería y el ejecutable test. Con esto empezamos desde el principio.

Los ficheros `Makefile` y el ejecutable `make` se utilizan para muchas más tareas que sólo programas en C. Cualquier dependencia que necesitemos que se cumpla, la podemos expresar en un fichero `Makefile` y hacer que `make` nos la mantenga.

## Capítulo 33

# Compilación del kernel

Una de las razones principales del crecimiento explosivo que Linux ha experimentado es la disponibilidad del código fuente de casi todas las partes del sistema, incluyendo el *kernel* del sistema operativo. Esta libertad permite a personas de toda la Internet modificar o ampliar la funcionalidad del sistema, según se lo permita su talento de programación. Esta libertad ha atraído la atención de alguna de las mentes más brillantes de Internet, las cuales contribuyen a hacer de Linux un sistema operativo flexible y estable. (*Administración de Red Hat Linux*, TOMAS SCHENK, Prentice Hall)

Alguna vez puede ser necesario cambiar la versión del kernel de Linux. Puede ser para:

1. obtener nuevas características
2. arreglos de errores cuando estén disponibles
3. porque hayamos añadido un nuevo elemento hardware no soportado por el actual o
4. queremos habilitar ciertas características.

Con el uso de módulos (partes que se cargan en el kernel conforme se necesitan), la necesidad de compilar nuestro kernel se reduce a casos muy limitados.

Como ya sabemos, existen dos líneas de desarrollo del kernel. Unas son las consideradas versiones "estables" y otras las denominadas "de desarrollo". La diferencia entre las dos líneas es que en los kernels de desarrollo se introducen características nuevas, no totalmente probadas, pero que pueden satisfacer a los usuarios ávidos de nuevas funcionalidades y a los que no les importe demasiado que su máquina pueda rebotarse (aunque esto no suceda frecuentemente). A cambio obtienen casi de inmediato características que están en la "cresta de la ola".

Las versiones estables introducen nuevas características y drivers para hardware solamente después de que hayan sido suficientemente probadas en las versiones de desarrollo. Las dos líneas utilizan números de versiones separados para distinguirlas. Los núcleos estables son nombrados con el segundo número par (eg. 2.0.35, 2.2.14, 2.4.3), mientras que los de desarrollo tienen el segundo número impar (eg. 2.1.120, 2.3.14, 2.5.1).

Siempre que vayamos a instalar un nuevo kernel, deberíamos asegurarnos de que tenemos una posibilidad de arrancar del antiguo por si algo fuera mal. Podemos crear un disco de arranque con el kernel que estamos utilizando antes de hacer cualquier cambio:

```
#mkbootdisk --device /dev/fd0 2.4.18-14
```

Por otro lado y a tenor de la rapidez con que aparecen/cambian los núcleos, antes de "tocarlo" no está de más mirar en la página:

<http://www.redhat.com/support/docs/howto/kernel-upgrade/kernel-upgrade.html>

por si hay alguna cuestión de interés que se haya cambiado respecto a lo que aquí se expone.





### 33.1. Subiendo de versión nuestro kernel mediante un RPM

La forma más fácil de subir nuestro kernel es utilizando un kernel ya precompilado y en formato RPM. Estos núcleos soportan una gran variedad de hardware y características comunes. Aunque el cambiar el kernel debe ser siempre hecho con cuidado, esta forma no implica demasiado riesgo y sería la siguiente:

```
#rpm -ivh kernel-2.4.18-24.8.0.i386.rpm1
#cd /boot
#ls -l
```

Comprobamos el nombre del nuevo kernel y si el enlace `vmlinuz` apunta a él.

El comando `rpm` se encarga de llamar automáticamente al comando `/sbin/new-kernel-pkg`. Éste hace dos tareas. Se encarga de generar una imagen `initrd`<sup>2</sup> y de actualizar nuestro gestor de arranque<sup>3</sup>.

Si nuestro gestor de arranque es GRUB, se actualizará la entrada en el fichero `/etc/grub.conf` para arrancar de este nuevo kernel.

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this
file
# NOTICE: You do not have a /boot partition. This means that
# all kernel and initrd paths are relative to /, eg.
# root (hd0,1)
# kernel /boot/vmlinuz-version ro root=/dev/hda2
# initrd /boot/initrd-version.img
#boot=/dev/hda
default=1
timeout=10
splashimage=(hd0,1)/boot/grub/splash.xpm.gz
title Red Hat Linux (2.4.18-24.8.0)
    root (hd0,1)
    kernel /boot/vmlinuz-2.4.18-24.8.0 ro root=LABEL=/
    initrd /boot/initrd-2.4.18-24.8.0.img
title Red Hat Linux (2.4.18-14)
    root (hd0,1)
    kernel /boot/vmlinuz-2.4.18-14 ro root=LABEL=/
    initrd /boot/initrd-2.4.18-14.img
title win
    rootnoverify (hd0,0)
    chainloader +1
```

Vemos que se han añadido automáticamente las líneas correspondientes para el kernel 2.4.18-24.8.0 que acabamos de instalar.



### Sólo si usamos LILO

En el caso de que sea LILO el gestor de arranque que tenemos, se modificará el fichero `/etc/lilo.conf`, el fichero de configuración de LILO. En caso contrario, no sabría cómo encontrar el nuevo núcleo.

Un posible fichero de configuración podría ser:

<sup>1</sup>Vemos que la opción es `i` (instalar) y no `U` (actualizar), de forma que el kernel antiguo siga en nuestro sistema. Es una alternativa para que en caso de fallo dispongamos del kernel anterior.

Además, dependiendo del micro del que dispongamos deberíamos (así optimizamos el rendimiento del núcleo) instalar el apropiado para nuestra máquina: 386, 586, 686, athlon, etc

<sup>2</sup>Necesario si necesitamos soporte de módulos en tiempo de arranque. Por ejemplo si hemos compilado el soporte SCSI como módulo y nuestro disco de arranque es SCSI. También el sistema de ficheros ext3 lo necesita.

<sup>3</sup>GRUB o LILO



```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
linear
default=linux

image=/boot/vmlinuz
    label=linux
    read-only
    root=/dev/hda2

image=/boot/vmlinuz-2.4.18-14
    label=viejo
    initrd=/boot/initrd-2.4.18-14.img
    read-only
    root=/dev/hda2

other = /dev/hda1
    label = w
```

Debemos ejecutar `/sbin/lilo` para que se actualice. No debe devolver mensaje de error, como que no encuentra el fichero, sólo las etiquetas de los sistemas a arrancar.

Rebotando el sistema, tendremos nuestro nuevo kernel en acción.

Podemos comprobarlo por el mensaje que nos muestra nuestra máquina

```
Red Hat Linux release 8.0 (Psyche)
Kernel 2.4.18-24 on an i686
login:
```

O mediante el comando `uname`

```
$ uname -a
Linux localhost.localdomain 2.4.18-14 #1 Tue Mar 7 20:35:25 EST 2002 i686
GNU/Linux
```

## 33.2. Construyendo un kernel a medida.

Si nuestro caso es uno de los siguientes:

1. tenemos un hardware no soportado en los kernels precompilados,
2. queremos ajustar nuestro kernel para modificar alguno de los parámetros por defecto,
3. optimizar el sistema, no cargando soporte para hardware o características que no necesitamos y así ahorrar memoria, o



4. simplemente somos amantes de Linux y hasta que no hemos compilado un kernel no nos hemos demostrado hasta dónde somos capaces de llegar;

entonces estamos en el sitio correcto. (Además, compilar un kernel es una buena forma de probar que todo en nuestra máquina funciona correctamente: hardware, instalación... porque realmente se fuerza al sistema a trabajar duro)

Los fuentes del kernel podemos instalarlos desde un paquete RPM, en nuestro caso necesitaremos `kernel-source` (los fuentes del kernel, están el CD2), `glibc-kernheaders` (ficheros de cabecera necesarios, CD1), si es un portátil `kernel-pcmcia-cs` (CD1). En este caso vamos a compilar el kernel que viene con la distribución de RedHat 8.0. Instalamos

```
#rpm -Uvh kernel-source-2.4.18-14.i386.rpm
```

Y en caso de ser necesario,

```
#rpm -Uvh kernel-pcmcia-cs-3.1-31-9.i386.rpm
```

Después de instalar los paquetes del código fuente, nos trasladamos (como superusuario) al directorio `/usr/src`. Viendo el contenido<sup>4</sup>, observamos que se ha creado un directorio `linux-2.4.18-14` y un enlace llamado `linux-2.4`.

```
lrwxrwxrwx 1 root root 12 Nov 22 23:48 linux-2.4 ->linux-2.4.18-14
drwxr-xr-x 19 root root 4096 Apr 29 17:16 linux-2.4.18-14
```

### Estructura del árbol de los fuentes

En un sistema Linux, las fuentes se ubican normalmente bajo el directorio `/usr/src/linux`, si bien RedHat usa el directorio `/usr/src/linux-version_nucleo` y un enlace simbólico que apunta hacia él (`/usr/src/linux-2.4`). En el siguiente texto los directorios están dados en forma relativa a éste. El código dependiente de la arquitectura está bajo `arch/`. Hasta el presente éste contiene `arch/i386/` para los procesadores 386, 486 y Pentium de Intel, `arch/mips/` para la arquitectura MIPS y `arch/sparc/` para la SPARC de Sun Microsystems.

El kernel de Linux no es más que un programa C estándar. Hay sólo dos grandes diferencias. La función de entrada para los programas escritos en C es normalmente `main(int argc, char **argv)`. En el kernel de Linux ésta es `start_kernel(void)`. El entorno del programa no existe todavía (hablamos del momento de boot cuando la máquina arranca y está por correr el kernel de Linux). Esto significa que hay que hacer un par de maniobras antes de que la primera función en C sea llamada. Los códigos en ensamblador que se encargan de esta labor están contenidos en el directorio `arch/i386/asm/`. La rutina en ensamblador apropiada carga el kernel en la dirección absoluta `0x100000` (1 Mbyte). Entonces instala las rutinas de servicio de interrupción, las tablas de descriptoras globales y las tablas descriptoras de interrupciones, que son utilizadas únicamente durante el proceso de inicialización. En este punto el procesador se pasa a modo protegido.

El directorio `init/` contiene todo lo necesario para inicializar el kernel. Además de otras funciones aquí está `start_kernel()`. Su tarea es la de inicializar el kernel de manera correcta, tomando en cuenta los parámetros de boot que le fueron pasados. El primer proceso es creado sin el uso de las llamadas al sistema (todavía no tenemos el sistema cargado), es decir, a mano. Este proceso es el famoso `idle`, es el que usa el tiempo del procesador cuando ninguna tarea lo requiere.

Los directorios `kernel/` y `arch/i386/kernel/` contienen, como sus nombres sugieren, las partes centrales del kernel. Aquí es donde las principales llamadas al sistema (*system calls*) son alojadas. Aquí también se encuentran implementadas otras áreas como pueden ser el manejo del tiempo, el planificador (*scheduler*), la DMA, el controlador de interrupciones y el controlador de señales.

El código que maneja la memoria está almacenado en los directorios `mm/` y `arch/i386/mm/`. Esta parte del código es la encargada de asignar y liberar memoria a los procesos. Aquí también se aloja todo el código que se encarga de la Paginación.

<sup>4</sup>Véase 33.2 para conocer más sobre la estructura de este directorio.



El sistema de archivos virtual (`virtual file system`) está en el directorio `fs/`. Los diferentes sistemas de archivos soportados se encuentran en subdirectorios de éste. Los sistemas de archivos más importantes son `Ext2`, `Ext3` y `Proc`.

Todos los sistemas operativos requieren de un driver para su correspondiente componente de hardware. Estos están alojados en el directorio `drivers/`.

En el directorio `ipc/` se encuentran los fuentes de la implementación del IPC (comunicación entre procesos) del System V.

La implementación de varios protocolos de red, además del código para los sockets y los dominios de Internet están almacenados en `net/`.

Algunas funciones estándar en C están implementadas en `lib/`, así la programación del kernel puede utilizar los hábitos de la programación en C.

Los módulos generados cuando el kernel es compilado están contenidos en el directorio `modules/`. Este directorio está normalmente vacío hasta que el kernel es compilado por primera vez.

Probablemente el directorio más importante que utilizan los programadores del núcleo es `include/`. Aquí se alojan todos los ficheros de cabecera (*headers* o *includes*) que son específicos del kernel. El directorio `include/asm-386/` contiene los *headers* que son específicos del kernel y para la plataforma Intel 386.



Si necesitamos drivers adicionales que no estén incorporados al fuente del núcleo y vengan en forma de parche, podremos ponerlos ahora. Como ejemplo, supongamos que necesitamos un driver para el controlador *RAID Mylex DAC960*. Descargamos el paquete que se llama `DAC960-2.0.0-Beta4.tar.gz`<sup>5</sup>. Realizamos las acciones siguientes:

```
# cd /usr/src/  
# tar zxvpf DAC960-2.0.0-Beta4.tar.gz  
# mv README.DAC960 DAC960.[ch] /usr/src/linux/drivers/block  
# patch -p0 <DAC960.patch
```

(Estos pasos normalmente no serán necesarios, sólo cuando necesitemos un driver que no está integrado en el kernel todavía. Los incluimos como referencia para casos avanzados)

El próximo paso es configurar las opciones de nuestro kernel. Es el paso más importante, porque podemos no configurar opciones necesarias para algún elemento hardware del que dispongamos y que no funcionaría. Por eso, es necesario un conocimiento del hardware que tenemos. Continuemos con el proceso.

```
# cd /usr/src/linux  
# make mrproper6  
# make config
```

Nos irán saliendo las distintas opciones para que las habilitemos o no en nuestro kernel. Las opciones son seleccionar (**Y**), mediante la cual insertamos en el kernel lo necesario para que funcione esa característica. No seleccionar (**N**), con la que quitamos el soporte del kernel para ese elemento. O cargar como módulo (**M**). Esta opción hace que el kernel soporte esa característica en cuestión, pero no se incorpore directamente a él. Se pone en el sistema de ficheros el código compilado (específicamente en `/lib/modules/version`) y el kernel cuando necesite soportar dicha característica cargará en memoria el módulo. Con ello ahorramos memoria, tenemos un kernel eficiente y en el momento adecuado cargará el soporte adicional. Cuando pasado un tiempo no se necesite más el módulo, el propio kernel lo descargará. No todas las opciones se pueden poner en forma de módulos. También se puede seleccionar ayuda sobre lo que hace cada opción en particular.

Los métodos que tenemos disponibles para realizar la configuración son:

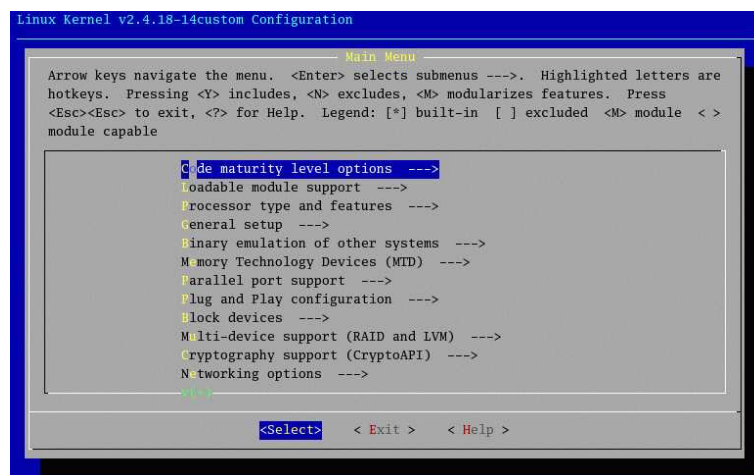
<sup>5</sup>Habría que asegurarse que el parche que incorporamos es el correspondiente a la versión del kernel que vamos a compilar.

<sup>6</sup>Sí Mr. Proper el que limpia, ¿adivinas lo que hace? Exacto. Limpia todos los ficheros para que podamos empezar desde cero

**make config** En forma de texto que interactivamente nos pregunta por el valor. Los componentes son presentados uno detrás de otro y contestamos **Y** (yes), **N** (no), o **M** (module). La opción por defecto se presenta en mayúsculas.

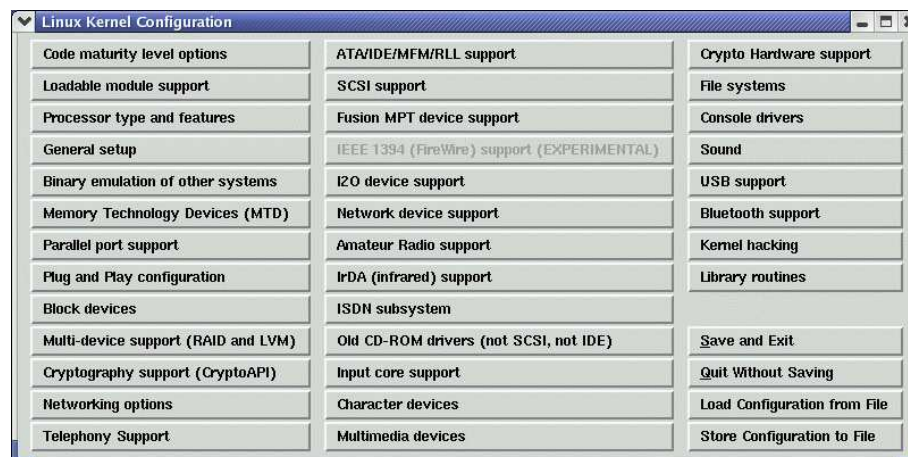
```
[root@pcl linux-2.4]# make config
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in configs/kernel-2.4.18-athlon.config
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL)
[Y/n/?]
```

**make menuconfig** En forma de menús en modo texto. Hay que señalar que necesitamos los paquetes `ncurses` y `ncurses-devel` para poderla utilizar.



**make xconfig** Programa en modo gráfico X Window. Para poder acceder a él tenemos que tener instalado el paquete del CD2 `tk-8.3.3-74.i386.rpm`. Así que antes de ejecutarlo comprobemos si está instalado

```
$rpm -q tk
tk-8.3.3-74
```





**make oldconfig** Es un script no interactivo que activará opciones por defecto de la distribución RedHat. Es útil para comenzar con algo que funciona y luego modificar sólo unas cuantas opciones.

Comentamos aquí las opciones del primer nivel que nos encontramos al ejecutar `make xconfig`. Dentro de ellas, tendremos las opciones específicas que podemos activar (**Y**), desactivar (**N**) o cargar como módulo (**M**).

- *Code maturity level options* (opciones de madurez del código): nos permite seleccionar si solamente nos presenta opciones de hardware plenamente soportadas y depuradas, o por el contrario nos presenta opciones todavía en desarrollo o no completamente probadas.
- *Loadable module support* (soporte para módulos cargables): permite activar el soporte de módulos cargables.
- *Processor type and features* (tipo de procesador y características): nos permite seleccionar el tipo de procesador de que disponemos (386, 486, Pentium, AMD...) y otras características como soporte multiprocesador.
- *General setup* (configuración general): como puede ser el soporte de PCI, tipos de formatos binarios, puerto paralelo...
- *Memory Thecnology Devices (MTD)* Soporte para dispositivos de memoria (memorias flash...)
- *Parallel port support* Soporte del puerto paralelo y sus diversos interfaces
- *Plug and Play configuration* (soporte de Plug and Play): permite su activación o desactivación.
- *Block devices* (dispositivos de bloque): opciones relacionadas con los dispositivos de bloques (discos duros, cintas, cd-rom...) y sus interfaces. Destacar que para las grabadoras de cd-rom IDE hay que habilitar la opción de emulación SCSI.
- *Multi-device support (RAID and LVM)* Soporte avanzado de discos virtuales RAID y manejo de volúmenes lógicos. Utilizado en servidores para mejorar la seguridad y facilidad de manejar grandes cantidades de espacio en disco.
- *Cryptography suport (CryptoAPI)* Soporte de criptografía
- *Networking options* (opciones de red): configuramos el TCP/IP y otros protocolos de red y sus opciones. Hay que destacar que aquí no es donde se configura el soporte para las tarjetas de red, que será en la opción Network device support, sino los protocolos de red.
- *Telephony support* (soporte de telefonía): tarjetas que permiten Voz sobre IP (VoIP)
- *ATA/IDE/MFM/RL support* Soporte de discos duros, cd-rom, en general, los que no sean SCSI
- *SCSI support* (soporte SCSI): permite seleccionar las opciones generales de bus SCSI y en un submenú, el soporte para las tarjetas SCSI concretas.
- *Fusion MPT device support* Soporte SCSI sobre fibra óptica
- *IEEE 1394 (FireWire) support* Soporte de Firewire, también conocido como i-Link, presente en cámaras digitales...
- *I2O device support* (Soporte de la arquitectura Intelligent Input/Output<sup>7</sup>)
- *Network device support* (soporte para dispositivos de red): aquí vienen las tarjetas de red
- *Amateur Radio support* (soporte para radioaficionados): no las personas, sino los equipos.

---

<sup>7</sup>IIO o I2O



- *IrDA (infrared support)*: soporte para dispositivos con conexiones por infrarrojos.
- *ISDN subsystem*: subsistema RDSI (¿estará aquí nuestra tarjeta?).
- *Old CD-ROM drivers (not SCSI, not IDE)*: soporte para los antiguos cd-rom que tenían su propia tarjeta controladora. No son ni IDE, ni SCSI.
- *Input core support* Soporte USB para Human Interface Device
- *Character devices* (dispositivos de caracteres): como los puertos serie, consola...
- *Multimedia devices* Fundamentalmente capturadoras de vídeo y sintonizadores de audio
- *Crypto Hardware support* Soporte para aceleradores criptográficos
- *Filesystems* (sistemas de ficheros): para acceder a sistemas de ficheros de casi cualquier tipo
- *Console drivers* (drivers de consola): soporte gráfico para la consola (no X-Window)
- *Sound* (sonido): para escuchar mientras trabajas, disfrutas o sufres, según la ocasión.
- *USB support*: Soporte para USB
- *Bluetooth support* Soporte para Bluetooth, para redes de área personal hasta 10m
- *Kernel hacking*: para verdaderos expertos. Ayuda a solucionar problemas en caso de un kernel panic (el kernel se nos muere).
- *Library routines* De compresión y descompresión zlib
- El resto de botones son para guardar, restaurar la configuración y salir de la utilidad.



Si el número de opciones nos parece abrumador, podemos intentar encontrar información en castellano en:

- PROYECTO NULIES<sup>8</sup>: <http://nulies.hispalinux.es/>. Responsable de traducir las opciones de configuración disponibles al castellano. Actualmente está “parado” y sólo podemos bajarnos ficheros que permiten realizar todo el proceso de selección de opciones<sup>9</sup> en castellano para núcleos hasta la versión 2.4.3.
- Otra página para estar al día: <http://es.kernelnewbies.org/>

Continuemos con la compilación de éste. Una vez definidas las opciones que queremos, retomemos nuestro trabajo.<sup>10</sup>

```
/usr/src/linux# make clean11
/usr/src/linux# make dep12
/usr/src/linux# make bzImage13
```

Compilamos ahora los módulos

```
/usr/src/linux# make modules14
```

<sup>8</sup>Núcleo de Linux en castellano

<sup>9</sup>Tanto las opciones disponibles como la ayuda emergente de cada opción

<sup>10</sup>Debemos comprobar que tenemos instalado el paquete dev86 y el binutils, pues son necesarios para los siguientes pasos.

<sup>11</sup>Limpiamos por si hubiera algo antiguo

<sup>12</sup>Se resuelven las dependencias

<sup>13</sup>Aquí se compila realmente el kernel

<sup>14</sup>Los dos últimos pasos dependiendo de nuestro ordenador, llevarán un tiempo considerable



Para no machacar los módulos del kernel actual en caso de que sean de la misma versión, deberemos renombrarlos.

```
# mv /lib/modules/2.4.18-14 /lib/modules/2.4.18-14-backup
```

Y luego, instalar los nuevos módulos:

```
/usr/src/linux# make modules_install  
/usr/src/linux# make install
```

Al kernel que hemos compilado nosotros desde las fuentes y que proviene de un paquete RPM<sup>15</sup>, para diferenciarlo del kernel precompilado que viene con la distribución, se le añade al final la palabra *custom*.

Mediante el comando `new-kernel-pkg` realizaríamos las dos tareas que faltan, como se comentó anteriormente: generar el `initrd`<sup>16</sup> y actualizar el gestor de arranque, GRUB o LILO.

```
#new-kernel-pkg --mkinitrd --depmod --install 2.4.18-14custom
```

---

<sup>15</sup>como `kernel-source-2.4.18-14.i386.rpm`

<sup>16</sup>Ram Disk de inicio, necesario para cargar módulos en tiempo de arranque. Si tenemos ext3 como sistema de ficheros, es necesario. Lo podríamos generar directamente con el comando `mkinitrd`:

```
#mkinitrd /boot/initrd-2.4.18-14.img 2.4.18-14
```



## Apéndice A

# Licencia de Documentación Libre GNU (traducción)

### A.1. GFDL

#### Versión 1.1, Marzo de 2000

Esta es la GNU Free Document License (GFDL), versión 1.1 (de marzo de 2.000), que cubre manuales y documentación para el software de la Free Software Foundation, con posibilidades en otros campos. La traducción<sup>1</sup> no tiene ningún valor legal, ni ha sido comprobada de acuerdo a la legislación de ningún país en particular. Vea el original <http://www.gnu.org/copyleft/fdl.html>

Los autores de esta traducción son:

- IGOR TÁMARA <mailto:ikks@bigfoot.com>
- PABLO REYES [mailto:reyes\\_pablo@hotmail.com](mailto:reyes_pablo@hotmail.com)
- Revisión : VLADIMIR TÁMARA P. <mailto:vtamara@gnu.org>

Copyright © 2000

Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Se permite la copia y distribución de copias literales de este documento de licencia, pero no se permiten cambios.

## 0. Preámbulo

El propósito de esta licencia es permitir que un manual, libro de texto, u otro documento escrito sea "libre" en el sentido de libertad: asegurar a todo el mundo la libertad efectiva de copiarlo y redistribuirlo, con o sin modificaciones, de manera comercial o no. En segundo término, esta licencia preserva para el autor o para quien publica una manera de obtener reconocimiento por su trabajo, al tiempo que no se consideran responsables de las modificaciones realizadas por terceros.

Esta licencia es una especie de "copyleft" que significa que los trabajos derivados del documento deben a su vez ser libres en el mismo sentido. Esto complementa la Licencia Pública General GNU, que es una licencia de copyleft diseñada para el software libre.

Hemos diseñado esta Licencia para usarla en manuales de software libre, ya que el software libre necesita documentación libre: Un programa libre debe venir con los manuales que ofrezcan la mismas libertades que da el software. Pero esta licencia no se limita a manuales de software; puede ser usada para cualquier trabajo textual, sin tener en cuenta su temática o si se publica como libro impreso. Recomendamos esta licencia principalmente para trabajos cuyo fin sea instructivo o de referencia.

---

<sup>1</sup>N. del T. Derechos Reservados en el sentido de GNU <http://www.gnu.org/copyleft/copyleft.es.html>

## 1. Aplicabilidad y definiciones

Esta Licencia se aplica a cualquier manual u otro documento que contenga una nota del propietario de los derechos que indique que puede ser distribuido bajo los términos de la Licencia. El "Documento", en adelante, se refiere a cualquiera de dichos manuales o trabajos. Cualquier miembro del público es un licenciatario, y será denominado como "Usted".

Una "Versión Modificada" del Documento significa cualquier trabajo que contenga el Documento o una porción del mismo, ya sea una copia literal o con modificaciones y/o traducciones a otro idioma.

Una "Sección Secundaria" es un apéndice titulado o una sección preliminar al prólogo del Documento que tiene que ver exclusivamente con la relación de quien publica o, los autores del Documento o, el tema general del Documento (o asuntos relacionados) y cuyo contenido no entra directamente en este tema general. (Por ejemplo, si el Documento es en parte un texto de matemáticas, una Sección Secundaria puede no explicar matemáticas.) La relación puede ser un asunto de conexión histórica, o de posición legal, comercial, filosófica, ética o política con el tema o la materia del texto.

Las "Secciones Invariantes" son ciertas Secciones Secundarias cuyos títulos son denominados como Secciones Invariantes, en la nota que indica que el documento es liberado bajo esta licencia.

Los "Textos de Cubierta" son ciertos pasajes cortos de texto que se listan, como Textos de Portada o Textos de Contra Portada, en la nota que indica que el documento es liberado bajo esta Licencia.

Una copia "Transparente" del Documento, significa una copia para lectura en máquina, representada en un formato cuya especificación está disponible al público general, cuyos contenidos pueden ser vistos y editados directamente con editores de texto genéricos o (para imágenes compuestas por píxeles) de programas genéricos de dibujo o (para dibujos) algún editor gráfico ampliamente disponible, y que sea adecuado para exportar a formateadores de texto o para traducción automática a una variedad de formatos adecuados para ingresar a formateadores de texto. Una copia hecha en un formato de un archivo que no sea Transparente, cuyo formato ha sido diseñado para impedir o dificultar subsecuentes modificaciones posteriores por parte de los lectores no es Transparente. Una copia que no es "Transparente" es llamada "Opaca".

Como ejemplos de formatos adecuados para copias Transparentes están el ASCII plano sin formato, formato de Texinfo, formato de  $\text{\LaTeX}$ , SGML o XML usando un DTD disponible ampliamente, y HTML simple que sigue los estándares, diseñado para modificaciones humanas. Los formatos Opacos incluyen PostScript, PDF, formatos propietarios que pueden ser leídos y editados únicamente en procesadores de palabras propietarios, SGML o XML para los cuáles los DTD y/o herramientas de procesamiento no están disponibles generalmente, y el HTML generado por máquinas producto de algún procesador de palabras solo para propósitos de salida.

La "Portada" en un libro impreso significa, la portada misma, más las páginas siguientes necesarias para mantener la legibilidad del material, que esta Licencia requiere que aparezca en la portada. Para trabajos en formatos que no tienen Portada como tal, "Portada" significa el texto cerca a la aparición más prominente del título del trabajo, precediendo el comienzo del cuerpo del trabajo.

## 2. Copia literal

Puede copiar y distribuir el Documento en cualquier medio, sea en forma comercial o no, siempre y cuando esta Licencia, las notas de derecho de autor, y la nota de licencia que indica que esta Licencia se aplica al Documento se reproduzca en todas las copias, y que usted no adicione ninguna otra condición a las expuestas en esta Licencia. No puede usar medidas técnicas para obstruir o controlar la lectura o copia posterior de las copias que usted haga o distribuya. Sin embargo, usted puede aceptar compensación a cambio de las copias. Si distribuye un número suficientemente grande de copias también deberá seguir las condiciones de la sección 3.

También puede prestar copias, bajo las mismas condiciones establecidas anteriormente, y puede exhibir copias públicamente.

### 3. Copiado en cantidades

Si publica copias impresas del Documento que sobrepasen las 100, y la nota de Licencia del Documento exige Textos de Cubierta, debe incluir las copias con cubiertas que lleven en forma clara y legible, todos esos textos de Cubierta: Textos Frontales en la cubierta frontal, y Textos Posteriores de Cubierta en la Cubierta Posterior. Ambas cubiertas deben identificarlo a Usted clara y legiblemente como quien publica tales copias. La Cubierta Frontal debe mostrar el título completo con todas las palabras igualmente prominentes y visibles. Además puede adicionar otro material en la cubierta. Las copias con cambios limitados en las cubiertas, siempre que preserven el título del Documento y satisfagan estas condiciones, puede considerarse como copia literal.

Si los textos requeridos para la cubierta son muy voluminosos para que ajusten legiblemente, debe colocar los primeros (tantos como sea razonable colocar) en la cubierta real, y continuar el resto en páginas adyacentes.

Si publica o distribuye copias Opacas del Documento cuya cantidad exceda las 100, debe incluir una copia Transparente que pueda ser leída por una máquina con cada copia Opaca, o entregar en o con cada copia Opaca una dirección en red de computador publicamente-accesible conteniendo una copia completa Transparente del Documento, sin material adicional, a la cual el público en general de la red pueda acceder a bajar anónimamente sin cargo usando protocolos de standard público. Si usted hace uso de la última opción, deberá tomar medidas necesarias, cuando comience la distribución de las copias Opacas en cantidad, para asegurar que esta copia Transparente permanecerá accesible en el sitio por lo menos un año después de su última distribución de copias Opacas (directamente o a través de sus agentes o distribuidores) de esa edición al público.

Se solicita, aunque no es requisito, que contacte a los autores del Documento antes de redistribuir cualquier gran número de copias, para permitirle la oportunidad de que le provean una versión del Documento.

### 4. Modificaciones

Puede copiar y distribuir una Versión Modificada del Documento bajo las condiciones de las secciones 2 y 3 anteriores, siempre que usted libere la Versión Modificada bajo esta misma Licencia, con la Versión Modificada haciendo el rol del Documento, por lo tanto licenciando la distribución y modificación de la Versión Modificada a quienquiera que posea una copia de este. En adición, debe hacer lo siguiente en la Versión Modificada:

- A. Uso en la Portada (y en las cubiertas, si hay alguna) de un título distinto al del Documento, y de versiones anteriores (que deberían, si hay alguna, estar listados en la sección de Historia del Documento). Puede usar el mismo título que versiones anteriores al original siempre que quién publicó la primera versión lo permita.
- B. Listar en la Portada, como autores, una o más personas o entidades responsables por la autoría o las modificaciones en la Versión Modificada, junto con por lo menos cinco de los autores principales del Documento (Todos sus autores principales, si hay menos de cinco).
- C. Estado en la Portada del nombre de quién publica la Versión Modificada, como quien publica.
- D. Preservar todas las notas de derechos de autor del Documento.
- E. Adicionar una nota de derecho de autor apropiada a sus modificaciones adyacentes a las otras notas de derecho de autor.
- F. Incluir, inmediatamente después de la nota de derecho de autor, una nota de licencia dando el permiso público para usar la Versión Modificada bajo los términos de esta Licencia, de la forma mostrada en la Adición (LEGAL)abajo.
- G. Preservar en esa nota de licencia el listado completo de Secciones Invariantes y en los Textos de las Cubiertas que sean requeridos como se especifique en la nota de Licencia del Documento

- H. Incluir una copia sin modificación de esta Licencia.
- I. Preservar la sección llamada "Historia", y su título, y adicionar a esta una sección estableciendo al menos el título, el año, los nuevos autores, y quién publicó la Versión Modificada como reza en la Portada. Si no hay una sección titulada "Historia" en el Documento, crear una estableciendo el título, el año, los autores y quien publicó el Documento como reza en la Portada, añadiendo además un artículo describiendo la Versión Modificada como se estableció en el punto anterior.
- J. Preservar la localización en red, si hay, dada en la Documentación para acceder públicamente a una copia Transparente del Documento, tanto como las otras direcciones de red dadas en el Documento para versiones anteriores en las cuáles estuviese basado. Estas pueden ubicarse en la sección "Historia". Se puede omitir la ubicación en red para un trabajo que sea publicado por lo menos 4 años antes que el mismo Documento, o si quien publica originalmente la versión da permiso explícitamente.
- K. En cualquier sección titulada "Agradecimientos" o "Dedicatorias", preservar el título de la sección, y preservar en la sección toda la sustancia y el tono de los agradecimientos y/o dedicatorias de cada contribuyente que estén incluidas.
- L. Preservar todas las Secciones Invariantes del Documento, sin alterar su texto ni sus títulos. Números de sección o el equivalente no son considerados parte de los títulos de la sección. M. Borrar cualquier sección titulada "Aprobaciones". Tales secciones no pueden estar incluidas en las Versiones Modificadas.
- M. Borrar cualquier sección titulada "Aprobaciones". Tales secciones no pueden estar incluidas en las Versiones Modificadas.
- N. No retitular ninguna sección existente como "Aprobaciones" o conflictuar con título con alguna Sección Invariante.

Si la Versión Modificada incluye secciones o apéndices nuevos o preliminares al prólogo que califican como Secciones Secundarias y contienen material no copiado del Documento, puede opcionalmente designar algunas o todas esas secciones como invariantes. Para hacerlo, adicione sus títulos a la lista de Secciones Invariantes en la nota de licencia de la Versión Modificada. Tales títulos deben ser distintos de cualquier otro título de sección.

Puede adicionar una sección titulada "Aprobaciones", siempre que contenga únicamente aprobaciones de su Versión Modificada por varias fuentes—por ejemplo, observaciones de peritos o que el texto ha sido aprobado por una organización como un standard.

Puede adicionar un pasaje de hasta cinco palabras como un Texto de Cubierta Frontal, y un pasaje de hasta 25 palabras como un texto de Cubierta Posterior, al final de la lista de Textos de Cubierta en la Versión Modificada. Solamente un pasaje de Texto de Cubierta Frontal y un Texto de Cubierta Posterior puede ser adicionado por (o a manera de arreglos hechos por) una entidad. Si el Documento ya incluye un texto de cubierta para la misma cubierta, previamente adicionado por usted o por arreglo hecho por la misma entidad, a nombre de la cual está actuando, no puede adicionar otra; pero puede reemplazar la anterior, con permiso explícito de quien publicó anteriormente tal cubierta.

El(los) autor(es) y quien(es) publica(n) el Documento no dan con esta Licencia permiso para usar sus nombres para publicidad o para asegurar o implicar aprobación de cualquier Versión Modificada.

## 5. Combinando documentos

Puede combinar el Documento con otros documentos liberados bajo esta Licencia, bajo los términos definidos en la sección 4 anterior para versiones modificadas, siempre que incluya en la combinación todas las Secciones Invariantes de todos los documentos originales, sin modificar, y listadas todas como Secciones Invariantes del trabajo combinado en su nota de licencia.

El trabajo combinado necesita contener solamente una copia de esta Licencia, y múltiples Secciones Invariantes Idénticas pueden ser reemplazadas por una sola copia. Si hay múltiples Secciones Invariantes

con el mismo nombre pero con contenidos diferentes, haga el título de cada una de estas secciones único adicionándole al final de este, en paréntesis, el nombre del autor o de quien publicó originalmente esa sección, si es conocido, o si no, un número único. Haga el mismo ajuste a los títulos de sección en la lista de Secciones Invariantes en la nota de licencia del trabajo combinado.

En la combinación, debe combinar cualquier sección titulada "Historia" de los varios documentos originales, formando una sección titulada "Historia"; de la misma forma combine cualquier sección titulada "Agradecimientos", y cualquier sección titulada "Dedicatorias". Debe borrar todas las secciones tituladas "Aprobaciones."

## 6. Colecciones de documentos

Puede hacer una colección consistente del Documento y otros documentos liberados bajo esta Licencia, y reemplazar las copias individuales de esta Licencia en los varios documentos con una sola copia que esté incluida en la colección, siempre que siga las reglas de esta Licencia para una copia literal de cada uno de los documentos en cualquiera de todos los aspectos.

Puede extraer un solo documento de una de tales colecciones, y distribuirlo individualmente bajo esta Licencia, siempre que inserte una copia de esta Licencia en el documento extraído, y siga esta Licencia en todos los otros aspectos concernientes a la copia literal de tal documento.

## 7. Agregación con trabajos independientes

Una recopilación del Documento o de sus derivados con otros documentos o trabajos separados o independientes, en cualquier tipo de distribución o medio de almacenamiento, no como un todo, cuenta como una Versión Modificada del Documento, teniendo en cuenta que ninguna compilación de derechos de autor sea llamada por la recopilación. Tal recopilación es llamada un "agregado", y esta Licencia no aplica a los otros trabajos auto-contenidos y por lo tanto compilados con el Documento, o a cuenta de haber sido compilados, si no son ellos mismos trabajos derivados del Documento.

Si el requerimiento de la sección 3 del Texto de la Cubierta es aplicable a estas copias del Documento, entonces si el Documento es menor que un cuarto del agregado entero, Los Textos de la Cubierta del Documento pueden ser colocados en cubiertas que enmarquen solamente el Documento entre el agregado. De otra forma deben aparecer en cubiertas enmarcando todo el agregado.

## 8. Traducción

La Traducción es considerada como una clase de modificación, Así que puede distribuir traducciones del Documento bajo los términos de la sección 4. Reemplazar las Secciones Invariantes con traducciones requiere permiso especial de los dueños de derecho de autor, pero puede incluir traducciones de algunas o todas las Secciones Invariantes adicionalmente a las versiones originales de las Secciones Invariantes. Puede incluir una traducción de esta Licencia siempre que incluya también la versión Inglesa de esta Licencia. En caso de un desacuerdo entre la traducción y la versión original en Inglés de esta Licencia, la versión original en Inglés prevalecerá.

## 9. Terminación

No se puede copiar, modificar, sublicenciar, o distribuir el Documento excepto por lo permitido expresamente bajo esta Licencia. Cualquier otro intento de copia, modificación, sublicenciamiento o distribución del Documento es nulo, y serán automáticamente terminados sus derechos bajo esa licencia. De todas maneras, los terceros que hayan recibido copias, o derechos, de su parte bajo esta Licencia no tendrán por terminadas sus licencias siempre que tales personas o entidades se encuentren en total conformidad con la licencia original.

## 10 Futuras revisiones de esta licencia

La Free Software Foundation puede publicar nuevas, revisadas versiones de la Licencia de Documentación Libre GNU de tiempo en tiempo. Tales nuevas versiones serán similares en espíritu a la presente versión, pero pueden diferir en detalles para solucionar problemas o intereses. Vea <http://www.gnu.org/copyleft/>.

Cada versión de la Licencia tiene un número de versión que la distingue. Si el Documento especifica que una versión numerada particularmente de esta licencia o "cualquier versión posterior" se aplica a esta, tiene la opción de seguir los términos y condiciones de la versión especificada o cualquiera posterior que ha sido publicada(no como un borrador)por la Free Software Foundation. Si el Documento no especifica un número de versión de esta Licencia, puede escoger cualquier versión que haya sido publicada(no como un borrador) por la Free Software Foundation.