

SOFTWARE LIBRE Y EDUCACIÓN:
GUADALINEX (DEBIAN) Y APLICACIONES
DIDÁCTICAS

Primeros pasos con Debian



Juan Alonso - Fermín Rubio - Paco Villegas

26 de marzo de 2005

Índice general

1. Inicio del Sistema	3
1.1. Gestores de arranque	3
1.1.1. Grub	4
1.1.2. Por si se opta por LILO.	14
1.2. Fichero /etc/inittab	17
1.2.1. Niveles de arranque	20
1.2.2. Control de acceso a servicios	23
1.2.3. Gdm	24
2. Sistemas de ficheros en Linux	27
2.1. Introducción.	27
2.2. Tipos de Sistemas de Ficheros.	28
2.3. Organización de los directorios	29
2.3.1. Ficheros de configuración del sistema	30
2.3.2. Logs del sistema	31
2.4. Creación de un Sistema de Ficheros.	32
2.5. Montaje y Desmontaje.	33
2.5.1. El fichero /etc/fstab	34
2.5.2. El comando mount	36
2.5.3. El comando umount	36
2.5.4. Herramientas gráficas para montar dispositivos	37
2.6. Chequeo y recuperación: fsck.	38
2.7. Enlaces	39
3. Permisos. Gestión de Usuarios	42
3.1. Introducción.	42
3.2. Permisos de acceso a los distintos objetos.	42
3.2.1. chmod	44
3.2.2. Permisos en modo gráfico	45
3.2.3. Más sobre permisos	46
3.3. Gestión de usuarios en modo texto.	48
3.4. Gestión de usuarios en modo gráfico.	50
4. Instalación, desinstalación de paquetes y actualización del sistema	56
4.1. apt - Introducción	56
4.1.1. ¿Qué es apt?	56
4.1.2. El archivo /etc/apt/sources.list	58
4.1.3. Agregar un CD-ROM al archivo sources.list	59
4.1.4. Instalar paquetes	61
4.1.5. Eliminando paquetes	63
4.1.6. Actualizando paquetes	65
4.1.7. Actualizando a una nueva versión	65



4.1.8.	Eliminando archivos de paquete no utilizados	66
4.1.9.	Instalando un nuevo entorno de escritorio: KDE	66
4.2.	dpkg - Introducción	69
4.2.1.	Instalar paquetes	69
4.2.2.	Desinstalar paquetes	70
4.2.3.	Opciones útiles	70
4.3.	synaptic	72
4.4.	Otros	75

Capítulo 1

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.

En este capítulo vamos a estudiar algunas cuestiones que aconsejan una primera lectura sin entrar en “profundidad” sobre todos los temas tratados. A medida que el curso avance, sí que es deseable tener una visión general clara sobre ellos.

1.1. Gestores de arranque

Cuando se enciende el ordenador se inicia una secuencia de procesos que es similar en todos los ordenadores basados en un sistema x86¹ y que se puede resumir en:

1. Al pulsar en el botón de encendido, el BIOS (Sistema Básico de Entrada y Salida) del sistema realiza la autoprueba de encendido (POST: *Power On Self Test*). Si todo va bien (se debe oír un sólo pitido), el BIOS continúa su trabajo configurando el hardware del ordenador y por último determina el dispositivo desde el que iniciar la carga del sistema operativo.
2. El dispositivo desde el que va a arrancar el sistema operativo lo podemos seleccionar a nuestro gusto² (dentro de las posibilidades de nuestra máquina). En general, y una vez instalados los sistemas operativos, será un disco duro (por ejemplo el maestro del primer canal IDE). El BIOS inicia la carga del registro de arranque maestro (MBR *Master Boot Record*).



El MBR se encuentra en el primer sector del disco (cilindro 0, cabeza 0, sector 1) y además del código necesario para iniciar la carga del sistema, contiene la tabla de particiones del disco. Ocupa los primeros 512 bytes del disco duro que son la suma de:

- 446 bytes se reservan para el código de lectura de arranque de la partición activa o el gestor de arranque.

¹Y por tanto en los Pentium o AMD de nuestros equipos.

²A través del Setup de la BIOS. Ya vimos esto en la primera entrega.



- 64 bytes para definir la tabla de particiones de hasta 4 entradas.
- 2 bytes para almacenar la Cifra Mágica (AA55 en hexadecimal). Si no está almacenado ese valor el MBR será tratado como no válido y se llama a la interrupción 18h para que la BIOS muestre un mensaje de error indicando que ese dispositivo no está preparado para el arranque.

Código de lectura de arranque de la partición activa o GRUB	
Tabla particiones	1ª partición
	2ª partición
	3ª partición
	4ª partición
Firma	Valor AA55h

3. Llegado a este punto, pueden ocurrir dos cosas:
 - a) No hemos instalado ningún gestor de arranque: en este caso se inicia la carga del sistema operativo instalado en la partición activa. Es lo usual en sistemas Windows.
 - b) Usamos Guadalinex y hemos instalado el gestor de arranque GRUB (o LILO) en el MBR. En este caso es ese gestor el que toma el control de la carga de los sistemas instalados en nuestro ordenador.

1.1.1. Grub

Si hemos cargado el gestor de arranque GRUB (*GRand Unified Boot loader*) en el MBR, 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: GRUB es un *boot loader* (Gestor de arranque) que permite iniciar la máquina según unas directivas concretas, además permite pasar parámetros al kernel e iniciar otros sistemas que no sean Linux.



Para encontrar información más amplia sobre GRUB

- `$info grub`
- En castellano: <http://www.linux.cu/manual/grub/grub.es.html>
- La documentación instalada en: `/usr/share/doc/grub` y las páginas man del programa
- La web del programa: <http://www.gnu.org/software/grub/>

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/menu.lst`).



En realidad el proceso no es tan “simple” y realmente se divide en varias etapas: 1, 1.5 y 2. Si se desea saber más sobre él se puede consultar:

<http://www.gnu.org/software/grub/manual/grub.html#Bootstrap%20tricks>

Si todo el proceso es correcto veremos en nuestro monitor



```

GNU GRUB  version 0.95  (638K lower / 522176K upper memory)

Guadalinux-2004
Guadalinux-2004 (A prueba de errores)

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.

The highlighted entry will be booted automatically in 6 seconds.

```

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:

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

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:

```

GNU GRUB  version 0.95  (638K lower / 522176K upper memory)

root (hd0,0)
kernel /boot/vmlinuz-2.6.5 root=/dev/sda1 splash=silent vga=791
initrd /boot/initrd.splash

Use the ↑ and ↓ keys to select which entry is highlighted.
Press 'b' to boot, 'e' to edit the selected command in the
boot sequence, 'c' for a command-line, 'o' to open a new line
after ('O' for before) the selected line, 'd' to remove the
selected line, or escape to go back to the main menu.

```



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. En la captura gráfica aparece (hd0, 0) ¿qué partición es?. Notar además que aunque el disco duro sea SCSI (sda) el sistema de numeración no cambia.

Desde esa ventana 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 eliminar 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 la ventana de línea de comandos. Para ver qué comandos tenemos disponibles podemos usar **help**:

```

completions of a device/filename. ESC at any time exits. ]
grub> help
background RRGGBB          blocklist FILE
boot                       cat FILE
chainloader [--force] FILE  clear
color NORMAL (HIGHLIGHT)  configfile FILE
displaym                   displaymem
find FILENAME              foreground RRGGBB
geometry DRIVE [CYLINDER HEAD SECTOR [
help [--all] [PATTERN ...] hide PARTITION
initrd FILE [ARG ...]      kernel [--no-mem-option] [--type=TYPE]
makeactive                 map TO_DRIVE FROM_DRIVE
mScrypt                    module FILE [ARG ...]
modulenounzip FILE [ARG ...] pager [FLAG]
partnew PART TYPE START LEN parttype PART TYPE
reboot                     root [DEVICE [HDBIAS]]
rootnoverify [DEVICE [HDBIAS]] serial [--unit=UNIT] [--port=PORT] [--
setup [--prefix=DIR] [--stage2=STAGE2_
splashimage FILE           terminal [--dumb] [--no-echo] [--no-ed
testvbe MODE               testvbe MODE
unhide PARTITION          uppermem KBYTES
vbeprobe [MODE]
grub> _
GNU GRUB version 0.95 (638K lower / 522176K upper memory)
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time exits. ]
grub> help color
color: color NORMAL (HIGHLIGHT)
Change the menu colors. The color NORMAL is used for most lines
in the menu, and the color HIGHLIGHT is used to highlight the
line where the cursor points. If you omit HIGHLIGHT, then the
inverted color of NORMAL is used for the highlighted line. The
format of a color is "FG/BG". FG and BG are symbolic color names.
A symbolic color name must be one of these: black, blue, green,
cyan, red, magenta, brown, light-gray, dark-gray, light-blue,
light-green, light-cyan, light-red, light-magenta, yellow and
white. But only the first eight names can be used for BG. You can
prefix "blink-" to FG if you want a blinking foreground color.
grub> _

```

Para obtener la ayuda de un comando escribiremos:

> help comando

Soporta la autocompletación de comandos (igual que las shell de Linux) mediante la tecla [Tab]³

➔ 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”.

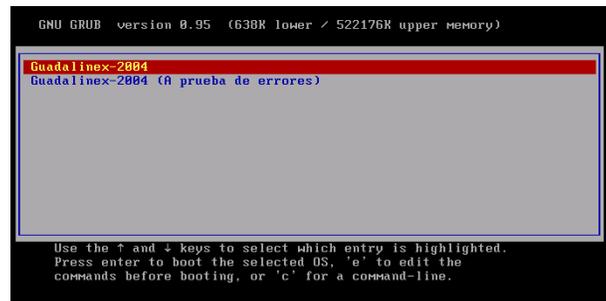
Ejecutar desde la ventana de línea de comandos de GRUB⁴:

³Es decir, no tenemos que conocer el nombre completo de un comando, si escribimos las letras iniciales y después pulsamos [TAB] se autocompleta el comando (si solo hay uno con esas letras iniciales, si hay varios que comiencen igual se muestran todos en pantalla y podremos elegir el adecuado).

⁴Recordar de nuevo que es posible usar la autocompletación de comandos.

```
grub> help
grub> displaymem
grub> cat (hd0,0)/autoexec.bat
grub> cat (hd0,1)/boot/grub/menu.lst
grub> geometry (hd0)
grub> color blue/light-gray yellow/red
```

Al pulsar sobre [Esc] el resultado final debe ser



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.



Fichero /boot/grub/menu.lst

La 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.

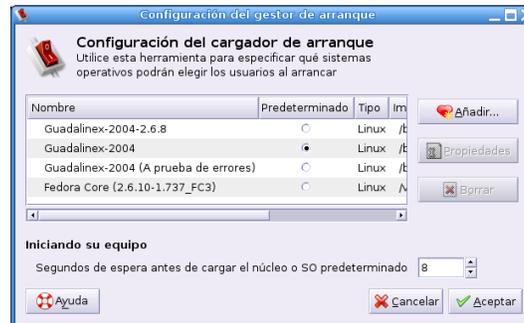
Disponemos de una herramienta gráfica de configuración de GRUB que podemos ejecutar bien desde una xterm

#boot-admin

o bien desde los menús



→Aplicaciones→Configuración→Sistema→Arranque



¿Y cuál es el fichero de configuración de Grub? pues el fichero /boot/grub/menu.lst



Analicemos el de un sistema modelo (es distinto del que tenéis en vuestros equipos). En primer lugar una aclaración: las líneas que comienzan con # son comentarios (algo que ya veremos es estándar en Linux) que se han añadido para explicar el fichero⁵.

```
$ cat /boot/grub/menu.lst
# sistema operativo por defecto, en mi caso el de título Guadalinex-2004 ...
default=06

# tiempo de espera (en segundos) antes de iniciar
# la carga del sistema de forma automática
timeout=10

# Si la opción de arranque por defecto da error, en vez de esperar
# a que el usuario haga algo, empezar de nuevo pero con
# la opción 1 (0 significa la primera opción)
fallback 1

# gráfico de fondo por defecto
#(en una práctica posterior veremos cómo se pone)
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
# texto que veremos en el menú
title Guadalinex-2004

# Establece y monta la partición "root": partición de arranque
root (hd0,0)

# núcleo a cargar y opciones de arranque7
kernel /boot/vmlinuz-2.6.5 root=/dev/hda1 splash=silent vga=791

# carga un disco RAM inicial: imagen initrd necesaria para dispositi-
vos SCSI
```

⁵Con el comando `cat` mostramos por pantalla el contenido del fichero pasado como argumento.

⁶las secciones `title` se numeran comenzando desde 0, cada una corresponde a un sistema o núcleo a cargar

⁷

- `splash=silent` es para arrancar con la pantalla de fondo en modo gráfico y que no veamos la salida del sistema cuando se inicia
- `vga=791`
Podemos usar los valores de la tabla



```

# y cuando el núcleo necesita de módulos "especiales" en el arranque
initrd /boot/initrd.splash
#Guadalinux "a prueba de errores"
title Guadalinux-2004 (A prueba de errores)
root (hd0,0)

#Núcleo y parámetros que se le pasan8
kernel /boot/vmlinuz-2.6.5 root=/dev/hda1 noapic noscsi nom-
ce vga=normal nosplash screen=800x600 xmodule=vesa xdepth=16 xvrefresh=60
initrd /boot/initrd.splash
# submenú de arranque del Windoze (9x o XP), y texto que veremos en el me-
nú principal
title Win98

# establece la partición de arranque pero sin montarla
rootnoverify (hd0,1)

# carga el fichero del primer sector de la
# partición 1 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/menu.lst
default=1
timeout=10
password --md5 $1$6U7Tn/$Pr3691BPbD/9nD8DZMg5A/
splashimage=(hd0,2)/boot/grub/splash.xpm.gz
title Linux
root (hd0,2)
kernel /boot/vmlinuz-2.6.5 root=/dev/hda3
initrd /boot/initrd.splash
title DOS
rootnoverify (hd0,0)
chainloader +1
title XP
rootnoverify (hd0,1)
chainloader +1

```



Si por alguna razón queremos eliminar este gestor de arranque podemos optar por reiniciar el sistema con un disco de Win 9x/ME y ejecutar el comando `fdisk /MBR`

Contraseñas

Colores	640x480	800x600	1024x768	1280x1024	1600x1200
256	769	771	773	775	796
32.768	784	787	790	793	797
65.536	786	788	791	794	798
16.8 M	786	789	792	795	799

Si con 791 nuestro monitor se pone "negro" podemos probar con 771, lo que nos permite una resolución de 800x600 con 256 colores. Notar que lo único que tenemos que hacer es modificar `vga=771`

⁸Para saber qué significado tienen se puede consultar: <http://intercentres.cult.gva.es/cefire/12400551/asesorias/informat/manual-knoppix/a1562.html>

1. Tenemos la posibilidad de poner una contraseña al gestor de arranque. Para esto (como root) hay que editar el fichero `/boot/grub/menu.lst` y añadir una línea del tipo ⁹:

```
password --md5 $1$6U7Tn/$Pr3691BPbD/9nD8DZMg5A/
```

Si usamos esta opción, es necesario introducir la contraseña (con **p**) para poder acceder a las opciones extendidas de GRUB.

Para desactivar esta opción sólo tenemos que comentarla con **#** o borrar la línea. Podemos generar la cadena encriptada 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`.

2. Con el programa `boot-admin`



podemos introducir “otra contraseña” si pulsamos sobre **Propiedades** (con Linux marcado en vídeo inverso) y en la ventana que aparece marcamos la opción de introducir contraseña. Pero no se trata de una contraseña para acceder a los menús de GRUB (como hemos visto antes), se trata de la contraseña que tendremos que introducir para acceder al sistema Linux que deseemos proteger.

Instalar GRUB

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, instalaremos GRUB en el MBR con:

```
#grub-install /dev/hda
```

En el disco duro Si GRUB no se instaló bien, podemos instalar GRUB como sigue:

1. Iniciamos el ordenador desde el lector de CDs con el CD de la distribución.
2. Cuando se haya cargado el SO, hay que abrir una Xterm
3. En esa terminal debemos ejecutar los comandos

a) Vamos a trabajar como administrador¹⁰

⁹La contraseña utilizada es “linux”.

¹⁰No hay que escribir nada en la contraseña del administrador cuando trabajas con el cdrom, solo pulsar **Intro**. Pondremos `#` para indicar que el comando se ejecuta como root, aunque no haya que escribirlo.



```
$su
password
```

- b) Vamos a crear un directorio en el que “poner” el Guadalinex de nuestro disco duro y lo montamos¹¹:

```
#mkdir sysimage
#mount -t ext3 /dev/hda2 sysimage
```

Suponemos que el disco con el que estás trabajando es el maestro del primer canal IDE y que Guadalinex está instalado en la segunda partición primaria (hda2). Si no es así tendremos que ajustarlo a nuestro sistema.

```
#chroot sysimage
```

Nos permite trabajar como si el árbol de ficheros partiera de ese directorio, se trata del directorio en donde se monta nuestro sistema Linux.

4. Con el editor `gedit` podemos modificar el fichero `/boot/grub/menu.lst` hasta ajustarlo a nuestra máquina, una vez que se adecúe a nuestro sistema ejecutaremos:

```
#grub-install /dev/hda
```

5. Para terminar sólo tenemos que ejecutar el comando `exit` (un par de veces).

➔ Para practicar

Password del root: Veamos cómo cambiar la contraseña del root (por si se nos olvida). Para eso, hemos de repetir los pasos anteriores del 1 al 3 y ejecutar el comando:

```
#passwd
```

e introducimos la nueva contraseña. Contraseña cambiada. Para salir sólo tenemos que ejecutar el comando `exit` (un par de veces).

Moraleja: el único sistema medio seguro es aquel al que no tenga nadie acceso ■

Creación de un disco de inicio Para disponer de grub en un disco¹² hay que realizar (como root) una serie de pasos¹³ que pasamos a enumerar:

1. Formateemos el disco con sistema de fichero ext2

```
#mke2fs /dev/fd0
```

2. Montemos el disquete en el directorio pasado como párametro. Con `-t ext2` le decimos al comando `mount` que se trata de un sistema de ficheros ext2

```
#mount -t ext2 /dev/fd0 /floppy
```

3. Instalemos grub en el disquete¹⁴

```
#grub-install --root-directory=/floppy '(fd0)'
```

¹¹Podemos comprobar de qué partición se trata escribiendo:

```
$/sbin/fdisk -l /dev/hda
```

¹²Notar que si accedemos al sistema sin más podemos crear ese disco sin tener que iniciar el ordenador con el CD, no hay que iniciar un entorno chroot

¹³El comando clásico (en Debian) para hacer discos de inicio si usamos lilo es `mkboot` (con Guadalinex edu habría que usar `mkboot` sin más)

¹⁴Las comillas son las de la tecla ?



4. Copiamos el fichero necesario para que todo funcione como debe¹⁵ (el fichero `menu.lst` debe estar adecuado a nuestro ordenador) y desmontamos el disquete

```
#cp /boot/grub/stage* /floppy/boot/grub
#cp /boot/grub/menu.lst /floppy/boot/grub
#umount /floppy
```

Listo, con esos comandos disponemos de un disquete de arranque por si borramos “accidentalmente” el gestor de arranque del MBR. Antes de nada deberíamos comprobar (cambiando en la BIOS la secuencia de arranque) que el disco está bien realizado.

Crear un CD de inicio. Para hacerlo, necesitamos una forma especial de `stage2` llamada `stage2_eltorito` y adecuar el fichero `menu.lst` a nuestro sistema. Guíemos el proceso:

1. Necesitamos una serie de directorios en los que realizar el trabajo, el contenido de esos directorios será después la imagen ISO que usemos para crear el CD. Creamos los subdirectorios con¹⁶:

```
# mkdir -p /opt/iso/boot/grub
```

2. Copiamos el archivo `stage2_eltorito` en el directorio `grub`:

```
# cp /lib/grub/i386-pc/stage2_eltorito /opt/iso/boot/grub
```

3. Copiamos el núcleo (o núcleos) y la imagen `initrd` en el directorio `/opt/iso/boot/`:

```
# cp /boot/vmlinuz-2.6.5 /opt/iso/boot/
# cp /boot/initrd.splash /opt/iso/boot/
```

4. Copiemos ahora el fichero de configuración de `grub` (`menu.lst`) en el directorio `/opt/iso/boot/grub` y lo adecuamos para que lea los archivos anteriores desde el CD: tenemos que sustituir en donde aparezca `hd` por `cd` y eliminar las líneas con `root (xxx)`. Por ejemplo:

```
timeout_8
default_0
fallback_1

5 title_Guadalinux-2004
kernel_(cd)/boot/vmlinuz-2.6.5_root=/dev/hda1_splash=silent_
  vga=791
initrd_(cd)/boot/initrd.splash

title_Guadalinux-2004_(A_prueba_de_errores)
10 kernel_(cd)/boot/vmlinuz-2.6.5_root=/dev/hda1_noapic_noscsi_
  nomce_vga=normal_nosplash_screen=800x600_xmodule=vesa_
  xdepth=16_xvrefresh=60
initrd_(cd)/boot/initrd.splash
```

Listado 1.1: `menu.lst`

5. Creamos ahora la imagen ISO9660 con¹⁷:

¹⁵En una práctica posterior veremos otra forma de poder iniciar un sistema en el caso de que no tengamos ese fichero.

¹⁶Al usar el parámetro `-p` los creamos de una sola “tajada”

¹⁷Ojo que es una sola línea y cuidado con los espacios



```
# cd /opt
# mkisofs -R -b boot/grub/stage2_eltorito -no-emul-boot -boot-load-size 4 -boot-info-table -o grub.iso iso
```

Ya sólo tenemos que usar el K3B para quemar en un CD la imagen creada y reiniciar el sistema para ver que funciona.

➔ Para practicar

1. Preparar el fichero `menu.lst` para que el CD de la sección anterior pueda arrancar sistemas Windows: se deja como ejercicio (es que a estas alturas del curso confiamos en que ya no sea necesario porque ya no se trabaje con él -;))
2. Si tenemos dos sistemas operativos instalados, modificar el fichero `/boot/grub/menu.lst` para que cambie el sistema de arranque por defecto:
 - Bien cambiando la línea: `default`
 - Bien permutando los bloques existentes para los sistemas operativos.
3. Añadir una contraseña a GRUB que impida que se pueda acceder a las opciones de edición sin más.
4. Modificar el fichero `menu.lst` para que GRUB se inicie siempre con los colores de ejemplo del **Para Practicar** en la página 7.
5. **Creación de un disquete de inicio (2):** Con el comando `grub-floppy` podemos hacer un disquete de inicio pero de funcionalidad menor que el que hemos creado anteriormente¹⁸. En esta práctica veamos como se hace e indiquemos la forma de iniciar Linux con él.

a) Creemos el disco con¹⁹:

```
# grub-floppy /dev/fd0
```

y respondamos que y a la cuestión planteada

b) Reiniciemos el sistema

```
# reboot
```

y cambiemos los parámetros de la BIOS para que se inicie desde la unidad A. Ahora se inicia grub pero, sólo nos sale el interfaz de comandos. Podemos usar `help` y ver de qué comandos disponemos y, que no se nos olvide, usar la socorrida autocompletación de comandos²⁰. Vamos a iniciar un sistema Guadalinux que suponemos está instalado en la segunda partición de nuestro maestro del primer canal IDE, para eso

```
grub> root (hd0,1)
Filesystem type is ext2fs, partition type 0x83
grub> kernel /boot/vmlinuz-2.6.5 root=/dev/hda1
[Linux-bzImage, setup=0x1400, size=0x1dd77f]
grub> initrd /boot/initrd.splash
[Linux-initrd @ 0x7fda000, 0x15a1b bytes]
grub> boot
```

y comienza la carga del sistema.

c) Si tenemos Windows instalado en la primera partición, ¿que tendríamos que hacer para cargarlo de esta manera?

6. **Imagen de fondo en el menú de GRUB:** Vamos a ver cómo poner una imagen de fondo en el menú de inicio de GRUB. La imagen de la que vamos a partir es `/usr/share/pixmaps/guadalinux/guadalinux-background.png`
Manos a la obra
 - a) Situémonos en la ruta en donde está la imagen

¹⁸En cierto sentido equivaldría a tener un grub instalado sin el fichero `menu.lst`, es decir, a realizar los pasos anteriores sin copiar ese fichero el disquete

¹⁹Si no funciona así tendremos que editar el archivo `/sbin/grub-floppy` y modificar la línea `pkglibdir=/lib/grub/*` dejándola de la forma `pkglibdir=/lib/grub/i386-pc`

²⁰Máxime ahora que el teclado parece estar un poco loco.

- ```
#cd /usr/share/pixmaps/guadalinex
```
- b) Convertamos la imagen al formato deseado<sup>21</sup> (formato xpm de dimensiones 640x480 y con una profundidad de 14 colores)
- ```
# convert -resize 640x480 -colors 14 guadalinex-background.png fondo.xpm
```
- c) La imagen la comprimirémos en formato gz.
- ```
gzip fondo.xpm
```
- d) Movamos el fichero obtenido a donde debe estar
- ```
# mv fondo.xpm.gz /boot/grub
```
- e) Modifiquemos el fichero de configuración de grub de forma que cargue la imagen en el inicio. Pero cuidado que la partición del disco en que está la imagen habrá que adecuarla a nuestro sistema:

```
...
timeout=10
# gráfico de fondo por defecto
splashimage=(hd0,1)/boot/grub/fondo.xpm.gz
...
# texto que veremos en el menú
title Linux
...

```



1.1.2. Por si se opta por LILO.

Este apartado lo hemos mantenido como información añadida. Se justifica desde la perspectiva de que LILO es el gestor de arranque por defecto de Guadalinex EDU. Por tanto, salvo que deseemos ampliar sobre este tema o porque tengamos Guadalinex EDU instalado, lo normal es que “saltemos” hasta 1.2 en la página 17.

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`.

Para encontrar información más amplia sobre LILO, se puede consultar en:

- `/usr/share/doc/lilo`, 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:

²¹También se puede hacer con Gimp.



- 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 4²² 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 correcto²³ veremos en nuestro monitor un menú con el que poder cargar el sistema operativo que deseemos.



Si pulsamos *intro* entraremos en el Sistema que hayamos definido por defecto. 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`. Comentemos cómo modificar algunos parámetros de forma manual²⁴, supongamos que nuestro fichero `/etc/lilo.conf` es:

```
#sección Global
boot=/dev/hda
vga=normal
prompt
#message = /boot/message
timeout=50
linear
```

²²Sobre LILO

²³Si 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.

²⁴Como ya se ha dicho antes, siempre que queramos modificar la configuración del sistema tendremos que trabajar como root.



```
append = "hda=scsi hdb=scsi hdc=scsi hdd=scsi hde=scsi hdf=scsi
          hdg=scsi hdh=scsi apm=power-off nomce"
#sección del sistema Linux
image=/boot/vmlinuz-2.4.20-xfs
  label=Guadalinux-Edu
  root=/dev/hda2
  read-only
#sección del sistema Windows
# other=/dev/hda1
#   label=dos
#   optional
#   table=/dev/hda
```

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 (#)

Sección Global

- `boot=/dev/hda` indica que LILO se carga desde el MBR del maestro del primer canal ide
- `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.
- `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.
- `timeout=50` 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).
- `default=Guadalinux-EDU` etiqueta del sistema a cargar por defecto, en este ejemplo si queremos comenzar con el otro sistema tendríamos que poner `dos`
- `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.

Sección linux

- `image=/boot/vmlinuz-2.4.20-xfs` ruta completa del fichero que contiene el kernel de Linux.
- `label=Guadalinux-EDU` etiqueta de este sistema, podemos poner lo 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/hda2` 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 “puede²⁵” 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.
- `password=contraseña` contraseña que permite cargar LILO.

Si modificamos el fichero `/etc/lilo.conf` es necesario ejecutar:

```
# lilo26
```

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.

1.2. 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.²⁷

²⁵Tanto la BIOS como el disco han de admitir transferencias de bloque EDD.

²⁶`lilo` se puede ejecutar con las opciones `-t` (*test*, prueba) o `-q` (*query*, consulta) para mostrar lo que LILO haría durante un inicio real.

²⁷Si después queremos acceder a esa salida por pantalla podemos usar la orden

```
$ dmesg
```

Para poder verla mejor, es usual usar (para salir q)

```
$ dmesg | less
```



```
RAMDISK driver initialized: 16 RAM disks of 8192K size 1024 blocksize
loop: loaded (max 8 devices)
Uniform Multi-Platform E-IDE driver Revision: 7.00alpha2
ide: Assuming 33MHz system bus speed for PIO modes; override with idebus=xx
PIIX4: IDE controller at PCI slot 0000:00:07.1
PIIX4: chipset revision 1
PIIX4: not 100% native mode: will probe irqs later
   ide1: BM-DMA at 0x1478-0x147f, BIOS settings: hdc:DMA, hdd:pio
hdc: VMware Virtual IDE CDROM Drive, ATAPI CD/DVD-ROM drive
Using anticipatory io scheduler
ide1 at 0x170-0x177,0x376 on irq 15
hdc: ATAPI 1X CD-ROM drive, 32kB Cache, UDMA(33)
Uniform CD-ROM driver Revision: 3.20
scsi: **** BusLogic SCSI Driver Version 2.1.16 of 18 July 2002 ****
scsi: Copyright 1995-1998 by Leonard N. Zubkoff <lnz@dandelion.com>
scsi0: Configuring BusLogic Model BT-958 PCI Wide Ultra SCSI Host Adapter
scsi0: Firmware Version: 5.07B, L/O Address: 0x1440, IRQ Channel: 11/Level
scsi0: PCI Bus: 0, Device: 16, Address: 0xF4000000, Host Adapter SCSI ID: 7
scsi0: Parity Checking: Enabled, Extended Translation: Enabled
scsi0: Synchronous Negotiation: Ultra, Wide Negotiation: Enabled
scsi0: Disconnect/Reconnect: Enabled, Tagged Queuing: Enabled
scsi0: Scatter/Gather Limit: 4096, Tagged Queuing: Enabled
scsi0: Scatter/Gather Limit: 128 of 8192 segments, Mailboxes: 211
scsi0: Driver Queue Depth: 211, Host Adapter Queue Depth: 192
scsi0: Tagged Queue Depth: Automatic, Untagged Queue Depth: 3
scsi0: *** BusLogic BT-958 Initialized Successfully ***
scsi0 : BusLogic BT-958
   Vendor: VMware, Model: VMware Virtual S Rev: 1.0
   Type: Direct-Access ANSI SCSI revision: 02
Red Hat/Adaptec aacraid driver (1.1.2-1k1 May 18 2004)
```

➔ Para practicar:

Si bien ahora está de moda “ocultar” el proceso de carga de Linux con gráficos de fondo, en general, es más instructivo y práctico (permite detectar problemas en tiempo de arranque) ver esa salida por pantalla. Podemos hacerlo de forma permanente modificando el fichero `/boot/grub/menu.lst` de forma adecuada o si sólo es para una vez, cuando se inicia Grub, editar (e) la línea

```
kernel /boot/vmlinuz-2.6.5 root=/dev/hda1 splash=silent vga=791
```

eliminando la parte final como sigue²⁸:

```
kernel /boot/vmlinuz-2.6.5 root=/dev/hda1
```

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 configuración²⁹ `/etc/inittab`³⁰. Mediante este fichero podemos ajustar el inicio del sistema para que se ejecute según nuestros intereses.

Un ejemplo de fichero `/etc/inittab`³¹ es:

```
#_ /etc/inittab: _init (8) _configuration .
#_ $Id: _inittab ,v_1.91_2002/01/25_13:35:21_miquels_Exp_$

#_ The _default _runlevel .
5 id:2: initdefault :

#_ Boot-time _system _configuration / initialization _script .
#_ This _is _run _first _except _when _booting _in _emergency _(-b) _mode .
si::sysinit:/etc/init.d/rcS
```

²⁸Una vez modificada pulsar [Intro] y b (de boot)

²⁹Se 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, un *script* podemos entenderlo como un fichero BAT del “antiguo” MSDOS), es decir, el orden en el que se escriben las líneas no es significativo.

En primer lugar *init* ejecuta el *script* `/etc/init.d/rcS`. Este *script*, irá llamando a los ficheros necesarios para configurar las variables de entorno, iniciar los sistemas de ficheros, ...

³⁰Quien quiera profundizar sobre el tema puede consultar la página man del fichero `inittab`

³¹Como siempre los # indican líneas de comentarios



```
#_What_to_do_in_single-user_mode.
~:~:~:~:S:wait:/sbin/sulogin

#_/etc/init.d_executes_the_S_and_K_scripts_upon_change
15 #_of_runlevel.
#
#_Runlevel_0_is_halt.
#_Runlevel_1_is_single-user.
#_Runlevels_2-5_are_multi-user.
20 #_Runlevel_6_is_reboot.

10:0:wait:/etc/init.d/rc_0
11:1:wait:/etc/init.d/rc_1
12:2:wait:/etc/init.d/rc_2
25 13:3:wait:/etc/init.d/rc_3
14:4:wait:/etc/init.d/rc_4
15:5:wait:/etc/init.d/rc_5
16:6:wait:/etc/init.d/rc_6
#_Normally_not_reached,_but_fallthrough_in_case_of_emergency.
30 z6:6:respawn:/sbin/sulogin

#_What_to_do_when_CTRL-ALT-DEL_is_pressed.
ca:12345:ctrlaltdel:/sbin/shutdown-t1-a-r-now

35 #_Action_on_special_keypress_(ALT-UpArrow).
#kb::kbrequest:/bin/echo"Keyboard_Request—edit_/etc/inittab_to_let_this_
work."

#_What_to_do_when_the_power_fails/returns.
pf::powerwait:/etc/init.d/powerfail_start
40 pn::powerfailnow:/etc/init.d/powerfail_now
po::powerokwait:/etc/init.d/powerfail_stop

#_/sbin/getty_invocations_for_the_runlevels.
#
45 #_The" id "field_MUST_be_the_same_as_the_last
#_characters_of_the_device_(after"tty").
#
#_Format:
#_<id>:<runlevels>:<action>:<process>
50 #
#_Note_that_on_most_Debian_systems_tty7_is_used_by_the_X_Window_System,
#_so_if_you_want_to_add_more_getty's_go_ahead_but_skip_tty7_if_you_run_X.
#
1:2345:respawn:/sbin/getty_38400_tty1
55 2:23:respawn:/sbin/getty_38400_tty2
3:23:respawn:/sbin/getty_38400_tty3
4:23:respawn:/sbin/getty_38400_tty4
5:23:respawn:/sbin/getty_38400_tty5
6:23:respawn:/sbin/getty_38400_tty6
60

#_Example_how_to_put_a_getty_on_a_serial_line_(for_a_terminal)
#
#T0:23:respawn:/sbin/getty-L_ttyS0_9600_vt100
#T1:23:respawn:/sbin/getty-L_ttyS1_9600_vt100
65

#_Example_how_to_put_a_getty_on_a_modem_line.
#
```

```
#T3:23:respawn:/sbin/mgetty_x0_s_57600_ttyS3
```

Listado 1.2: /etc/inittab

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 los que se ejecuta el proceso

acción explicaremos sólo las más usuales, para el resto de acciones ver los documentos de ampliación ya comentados

ctrlaltdel comando a ejecutar si se pulsa la combinación de teclas³² [CTRL]+[ALT]+[DEL]

initdefault nivel de ejecución por defecto

once el comando se ejecuta sólo una vez después de entrar en el nivel de ejecución seleccionado

powerfail comando a ejecutar si se recibe una señal SIGPWR (indica algún problema con la alimentación eléctrica)

powerokwait se ejecuta el comando si el SAI manda una señal indicando que se ha restablecido el nivel de energía eléctrica.

respawn se inicia una nueva instancia del proceso cada vez que termine

sysinit el *script* se ejecuta en la fase de arranque del sistema, independientemente del nivel de ejecución³³.

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.

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.

1.2.1. 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 6³⁴. El objetivo de cada uno de estos niveles es:

N carga del sistema (NONE)

S modo Monousuario (para no ser cambiado directamente)

³²Merece la pena fijarse en la línea:

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

si la mantenemos permitimos que cualquier usuario pueda hacer un *reset* con nuestra máquina con sólo pulsar [CTRL]+[ALT]+[DEL].

³³/etc/init.d/rcS → A “grosso modo” podemos decir que se encarga de configurar/cargar todo el sistema (los OK del inicio).

³⁴Debian utiliza el sistema de scripts **sys-V**.



- 0 (*halt*) para apagar el sistema
- 1 modo monousuario (equivale a `s` \rightarrow *single mode*)
- 2..5 modo multiusuario
- 6 (*reboot*) para reiniciar el sistema.

Orden de ejecución de los scripts³⁵ en `/etc/rc?.d`

- Todos los scripts ejecutados (o no) por el sistema `init` se encuentran en `/etc/init.d`. Los 8 directorios `/etc/rc?.d` ($? = S, 0 \dots 6$) contienen enlaces simbólicos a estos scripts³⁶. Estos enlaces se denominan `Sxxservicio` o `Kxxservicio`. En ambos casos, `xx` es un número de dos cifras y `servicio` es el nombre del script asociado.

Inicialmente se ejecuta el script `/etc/init.d/rcS`, él se encarga de que se ejecuten todos los scripts `S*` de `/etc/rcS.d`. Los scripts que tengan los números más pequeños se ejecutarán primero³⁷.

- Después de que se han ejecutado los scripts `/etc/rcS.d`, `init` cambia al nivel de ejecución especificado en `/etc/inittab` (normalmente 2). Para eso, `init` ejecuta el script `/etc/init.d/rc` que se ocupa de arrancar los servicios en `/etc/rc?.d`, donde `?` representa el nivel de ejecución pasado como argumento.

↔ Por ejemplo, si el nivel de arranque es el 2, `init` ejecuta

```
/etc/rc.d/rc 2
```

Este script ejecuta entonces los ficheros que se encuentran en el subdirectorio `/etc/rc2.d`. Los ficheros de este subdirectorio³⁸ 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.

El número de dos dígitos a continuación de la `K` o de la `S` indica el orden en que se ejecutará el script. Los scripts que tengan los números más pequeños se ejecutarán primero.

↔ Como ejemplo se puede revisar el script de configuración añadido por los creadores de Guadalinex (`/etc/init.d/guadalinex`) y comprobar que se ejecuta en el nivel de ejecución 2.

- Si deseamos personalizar los servicios en el arranque, es preferible hacerlo usando los ficheros contenidos en el subdirectorio `/etc/default` y no con los script del directorio `/etc/init.d`.

Cómo cambiar de nivel de ejecución

- La forma más sencilla es usando el comando³⁹:

³⁵Aunque ya se ha aclarado antes su importancia obliga a esta nueva nota.

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. Esto se llama *ejecución interpretada* de programas ya que cada línea se interpreta antes de su ejecución.

³⁶En 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, volveremos sobre ellos en un capítulo posterior) que están en `/etc/init.d`.

³⁷Solo deberían arrancarse ahora las cosas que necesitan ejecutarse una vez para tener el sistema en un estado consistente. No deberíamos arrancar demonios en este nivel. El nivel de ejecución en este momento es "`N`" (*none*).

³⁸Lo mismo que todos los de la forma `/etc/rcn.d` donde `n` es un número comprendido entre 0 y 6.

³⁹

- Se consigue lo mismo con el comando `telinit`.



```
#init [n]
```

donde `n` es un número que indica el nivel de arranque que deseamos “activar”. Cambiando niveles de ejecución:

↔ Si, por ejemplo, cambiamos desde el nivel 2 al nivel 5, se ejecutarán primero para el nivel de ejecución los scripts `K` y después todos los scripts `S` (ordenados de menor a mayor). Para optimizar el proceso, primero se revisa si un determinado servicio del nivel 5 se está ejecutando en el nivel de ejecución 2. Si ya lo estaba, y no hay ningún script `K` (stop) para él en el nivel 5 no se inicia de nuevo.

➔ Para practicar

1. Niveles de arranque

- a) Accede a la cuenta del root
- b) Modificar el fichero `/etc/inittab` de vuestro sistema para que arranque para un nivel 3
- c) Reinicia el sistema y comprueba que es así.
- d) Cambia a nivel 1 ejecutando

```
#init 1
```

Para que podamos acceder así es necesario que introduzcamos la *password* del root cuando se nos pida. Deshagamos el entuerto escribiendo

```
#telinit 2
```

- e) Dejar otra vez el fichero `/etc/inittab` como estaba al principio (nivel de ejecución 2)
- f) Reiniciar o apagar el sistema escribiendo respectivamente

```
init 6
```

ó

```
init 0
```



▪ Sólo por si tenemos LILO.

Para modificar de un nivel de arranque a otro tenemos como primera opción ejecutar

```
linux número_nivel_arranque40
```

cuando aparece

```
LILO boot:
```

al inicio del sistema. Por ejemplo, si nosotros siempre entramos en modo gráfico y deseamos hacerlo en modo texto escribiríamos⁴¹:

```
LILO boot: linux 3
```

- Podemos cambiar el nivel de ejecución en el arranque usando GRUB. Para eso, cuando aparezca el menú principal de GRUB pulsa la tecla `[e]` y en la ventana de edición escribe el número del nivel de ejecución (`[n]`) deseado,

```
kernel /boot/vmlinuz-2.6.5 [n] root=...
```

se escribe sólo el número tras el fichero que contiene al núcleo.

⁴⁰En el supuesto de que la etiqueta de nuestro sistema Linux sea `linux`

⁴¹Merece la pena aclarar que podemos pasar de un nivel a otro ejecutando:

```
# init "nuevo_nivel"
```

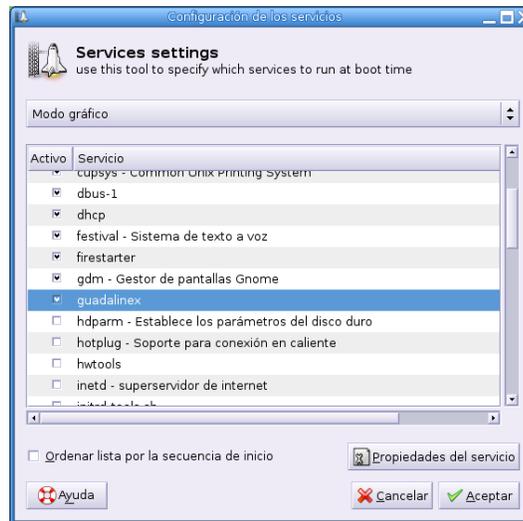
Así, si estamos en nivel 3 y deseamos pasar el 5 escribiríamos:

```
# init 5
```

1.2.2. Control de acceso a servicios

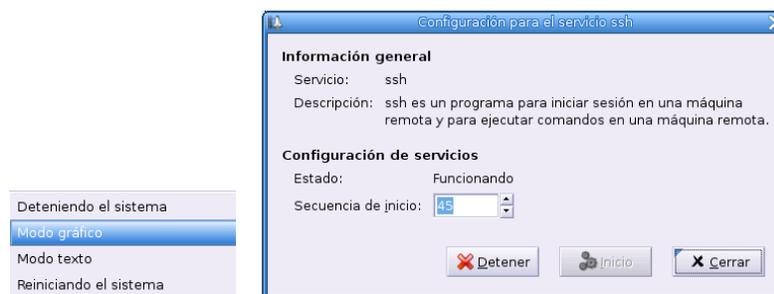
Vamos a enumerar las herramientas de que disponemos en Guadalinex para poder activar/desactivar determinados servicios según los distintos niveles de ejecución, se trata de⁴²:

services-admin (desde Gnome:  → **Configuración** → **Sistema** → **Servicios**), es una utilidad gráfica que permite seleccionar qué servicios están en activo.



Si optamos por marcar *Propiedades del servicio*, accederemos a funcionalidades añadidas:

- Seleccionar el nivel de arranque para el que realizamos los cambios
- Obtener más información sobre el servicio, establecer la prioridad y parámetros a pasarle.



/usr/sbin/update-rc.d utilidad en línea de comandos para activar/desactivar servicios. En general es más fácil trabajar con la anterior.

Usando este comando podemos configurar los enlaces simbólicos de los directorios **/etc/rc?.d** y el script situado en **/etc/init.d/**. Si por ejemplo deseamos que el servicio de nombre **servicio** se ejecute en el arranque

1. Se pone en el directorio **/etc/init.d/**. En general los programas que instalemos y que sean necesarios en el arranque sitúan sus scripts de forma automática aquí.

⁴²El comando a ejecutar desde una xterm es el que aparece en negrita



2. Después creamos los enlaces simbólicos⁴³ mediante el comando

```
update-rc.d servicio defaults
```

Al pasarle el parámetro `defaults` forzamos a que lo cree para los niveles de ejecución que van del 2 al 5.

↪ Por ejemplo, si ejecutamos

```
#update-rc.d ssh defaults 85
```

Creamos los enlaces simbólicos para el servicio de red `ssh` (*Secure Shell*) forzando a que los cree para los niveles de ejecución que van del 2 al 5. Con el 85 obligamos a que ese servicio se arranque antes de cualquier script que contenga un número mayor de 86.



Además de los programas anteriores, si tenemos un servicio en nuestra máquina, con:

```
$ /etc/init.d/service
```

podemos comprobar qué parámetros admite.

↪ Por ejemplo, con el servidor de impresión obtendríamos⁴⁴:

```
$ /etc/init.d/cupsys
```

```
Uso: /etc/rc.d/init.d/lpd {start|stop|restart|force-reload}
```

start arrancar	reload volver a cargar	force-reload forzar carga
stop detener	restart volver a arrancar	

O sea que si queremos pararlo sólo hay que ejecutar⁴⁵:

```
# /etc/init.d/cupsys stop
```

Parando cupsys: cupsd

1.2.3. Gdm

Si accedemos a nuestra máquina en modo gráfico, hemos estado utilizando el programa **gdm** (*GNOME Display Manager*):

⁴³Estos enlaces también se pueden crear a mano, pero eso mejor lo dejamos para cursos más avanzados.

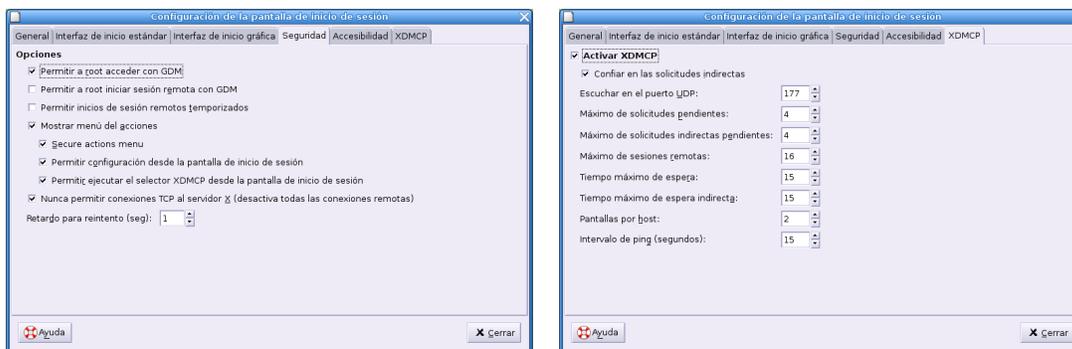
⁴⁴En general, estos parámetros funcionan para todos los scripts situados en `/etc/init.d/`.

⁴⁵También podemos usar:

```
#invoke-rc.d cupsys stop
```



La configuración de gdm se consigue⁴⁶ pulsando sobre  → **Configuración** → **Sistema** → **Configuración de de la pantalla de sesión**



Con esta utilidad podemos personalizar gdm tanto para conexiones locales como remotas:

- Con la pestaña **Seguridad** y marcando la casilla **Permitir a root acceder con GDM** permitimos que el root inicie una sesión gráfica en el sistema
- Por defecto las conexiones remotas están desactivadas, si deseamos permitir las hay que marcar la casilla **Activar XDMCP** en la pestaña **XDMCP** e iniciar el sistema en modo gráfico.
- Los cambios efectuados se almacenan en el fichero⁴⁷ /etc/X11/gdm/gdm.conf

➔ Para practicar:

1. Modificar la configuración de GDM para que permita el acceso como root⁴⁸.
2. Al hacer esta práctica se cerrarán todos los programas activos. Así que antes de hacerla debemos guardar todos los trabajos que tengamos a medias. Vamos a apagar el GDM escribiendo

```
#/etc/init.d/gdm stop
```

⁴⁶O con

```
#gdmsetup
```

⁴⁷Es un enlace al directorio /etc/gdm

⁴⁸Puede que ya esté hecho desde la 1ª entrega



Al hacerlo, cerramos el entorno gráfico y tendremos que entrar en modo consola.

3. Accedamos al sistema como root

```
login: root  
Password:
```

y ejecutemos el comando que inicia el servicio

```
#/etc/init.d/gdm start
```

y se nos mostrará de nuevo la pantalla gráfica de inicio del sistema. ■

Capítulo 2

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)

2.1. Introducción.

Revisemos los conceptos de formateo de un disco y de sistema de ficheros. Lo que denominamos formatear (por ejemplo, un floppy) comprende en realidad 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. 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 multivolumen¹.

En Linux trabajamos básicamente con cuatro tipos de ficheros:

1. Ficheros normales.
2. Directorios.
3. Enlaces.
4. Archivos especiales².

Los dos primeros son de sobra conocidos y del tercero daremos algunas pinceladas en esta sección.



Un par de anotaciones antes de seguir:

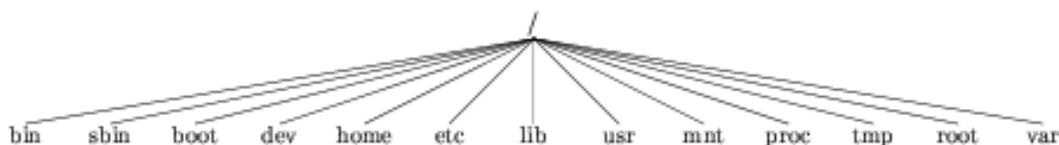
- Linux es “casesensitive”, o sea que el fichero `curso.txt` es distinto del fichero `Curso.txt` y lo mismo con los directorios.

¹Para ampliar sobre discos duros, particiones y sistemas de ficheros visita la página <http://www.saulo.net/pub/ddypart>.

²Aquí 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.

2.2. Tipos de Sistemas de Ficheros.

Linux soporta, además de los sistemas de ficheros nativos (minix, ext, xiafs, ext2 y ext3), varios tipos de sistemas de ficheros ajenos para facilitar el intercambio de datos con otros sistemas operativos.

Algunos de los sistemas de ficheros soportados por son:

ext2 Sistema de archivos estándar de Linux.

ext3 Sistema de ficheros que usa Guadalinex2004. Es una mejora del anterior con la ventaja de que es de tipo *journaling* (se reduce el tiempo de recuperación tras un apagado inesperado)³.

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.

ntfs Acrónimo de *new technology file system*, sistema de ficheros de Windows 2000/XP.

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.

³Características:

- 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 cómo hacerlo, además de la *manpage* del programa mirar la página <http://www.debianitas.net/docbook/ext3-v1.0/ext3-v1.0.html>.



minix Primer sistema de ficheros utilizado para Linux

ext Primer sucesor de minix (está en desuso).

xiafs Sucesor del ext (tampoco se usa)

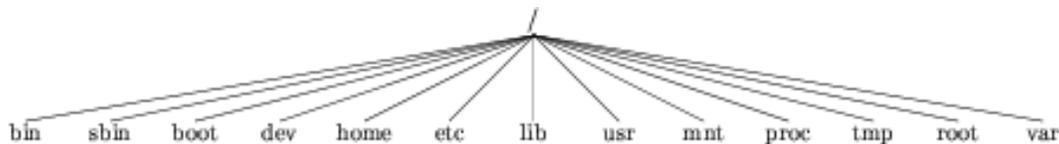
proc Sistema de archivos virtual de Linux.⁴

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 virtual⁵.

2.3. 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 ficheros⁶”, 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 directorios⁷:

/bin Contiene los comandos básicos del sistema operativo⁸

/dev Todos los ficheros de dispositivos.

/sbin Comandos esenciales para la administración del sistema⁹.

/etc Ficheros de configuración del sistema.

/home En él tendremos los directorios de trabajo de los usuarios.

/boot Aquí están los binarios de arranque del sistema.

/root Directorio de trabajo del “jefe”.

⁴No 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
```

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 memoria. 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.

⁵Algunas distribuciones (Guadalinex entre ellas), han optado por trabajar con un fichero como archivo de intercambio en lugar de dedicar una partición del disco a este menester.

⁶Incluso 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.

⁷Guadalinex y otras distribuciones crean además los directorios: **initrd** (si se borra no podremos acceder al sistema), **opt**, **auto** y **lost+found** (zona para poner los archivos “extraviados”) y otros para montar los dispositivos extraíbles.

⁸En **/usr/bin** también hay comandos del sistema pero son “menos básicos” que los que hay en **/bin**.

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.

⁹En **/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.



- /lib** Librerías básicas para trabajar en Linux. En él podemos encontrar información sobre el sistema.
- /mnt** De él cuelgan normalmente los sistemas de archivos de dispositivos extraíbles, floppy, CDROM, ... Esto no es obligatorio y Guadalinex se configura montándolos directamente en /floppy, /cdrom, /cdrom1,
- /tmp** En él se almacena información temporal.
- /usr** Aquí estarán los programas que no forman parte del sistema más básico.
- /proc** Ya se ha comentado anteriormente, este directorio no existe en realidad (físicamente).
- /var** ficheros de datos variables: logs del sistema, datos administrativos, etc.

2.3.1. Ficheros de configuración del sistema

Somos conscientes de que lo que sigue es un tanto árido, pero nos intriga saber dónde se configura “tal” comportamiento del sistema. Estos ficheros se almacenan en distintos lugares, fundamentalmente en:

\$HOME en el directorio de trabajo¹⁰ de cada usuario hay una serie de ficheros¹¹ que rigen el comportamiento de sus cuentas de registro.

/etc En este directorio está la mayoría de los ficheros de configuración del sistema¹²:

- aliases** puede contener listas de distribución utilizadas por el servicio de correo de Linux.
- bash.bashrc** valores por defecto en todo el sistema para la shell bash.
- csh.cshrc** en él se establecen los valores por defecto de la shell csh (shell C).
- exports** lista de los directorios locales para ser compartidos utilizando NFS (Network File System).
- fstab** contiene información que describe los diversos sistemas de ficheros y las ubicaciones del sistema Linux en las que se montan, se estudia en 2.5.1 en la página 34.
- group** en él se almacenan los nombres de grupo e ID de grupo (GID) definidos en el sistema.
- host.conf** 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).
- hosts** lista de direcciones IP y nombres de máquinas.
- hosts.allow** ordenadores que están autorizados para utilizar servicios TCP/IP desde el ordenador local.
- hosts.deny** ordenadores que no están autorizados a utilizar servicios TCP/IP desde el ordenador local.
- inittab** mediante este fichero podemos ajustar el inicio del sistema para que se ejecute según nuestros intereses
- inetd.conf** es el fichero de configuración para el demonio servidor **inetd**¹³
- ld.so.conf** lista de directorios que contienen librerías del sistema.
- mail.rc** parámetros del sistema asociados con el uso del correo.
- manpath.config** fichero de configuración del comando **man**.
- modules.conf** en él se establecen los alias y opciones para los módulos cargables de kernel.
- mtab** lista de sistemas de archivos montados en ese momento.
- passwd** cada línea de este fichero guarda información relativa a cada uno de los usuarios del sistema.
- printcap** archivo de configuración de las impresoras configuradas en nuestro equipo.
- profile** fichero que configura el entorno de trabajo, tiene validez para todo el sistema, programas de entorno e inicio de todos usuarios.
- protocols** en él se describen los distintos protocolos para Internet que están disponibles en el subsistema TCP/IP.

¹⁰**\$HOME** es una variable en la que se almacena la ruta del directorio de usuario que ejecuta la shell activa. Lo estudiaremos en temas posteriores. Podemos ver su valor ejecutando el comando **echo \$HOME**

¹¹El nombre de casi todos ellos empieza con un punto (.), son ficheros ocultos.

¹²Algunos puede que no estén (aún) en vuestro sistema y desde luego están otros que no enumeramos.

¹³Es un proceso especial que se queda a la escucha de conexiones TCP en unos puertos determinados. Cuando viene una solicitud de conexión, realiza una serie de comprobaciones y ejecuta el proceso servidor correspondiente.



resolv.conf contiene las direcciones IP de los servidores de nombres de nuestro proveedor de internet.

rpc define nombres y números para los procedimientos remotos de llamada.

services lista de servicios de red de Internet

shadow lista de contraseñas encriptadas para los usuarios que aparecen en el fichero **passwd**.

shells lista las rutas de los intérpretes de línea de comando con los que contamos en nuestro sistema.

sudoers fichero de configuración del comando **sudo**¹⁴.

syslog.conf en él se define qué mensajes de inicio de sesión recoge el demonio **syslogd** y qué ficheros se almacenan en él.

/etc/X11 directorio de configuración del sistema gráfico X y de los diferentes gestores de ventanas.

/etc/apt contiene, entre otros, el fichero **sources.list** con la lista de repositorios que utilizará APT para la instalación de paquetes.

/etc/cron.* distintos directorios con ficheros en los que se define la forma en que **crond** 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/init.d contiene las copias permanentes de los guiones de nivel de ejecución. Estos guiones están vinculados a ficheros en los directorios **/etc/rc?.d** 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).

2.3.2. 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 que monitoriza el registro general del sistema

klogd registro específico de la actividad del kernel

La configuración de los logs del sistema se determina en el fichero **/etc/syslog.conf**, y los ficheros en donde se almacenan los resultados están en el directorio: **/var/log**. Algunos de estos ficheros son:

¹⁴Con él podemos conseguir de una forma segura que determinados comandos restringidos al root puedan ser ejecutados por otros usuarios.



mail.* mensajes generados por el demonio `sendmail`¹⁵.

dmesg mensajes de arranque del kernel.

messages “cajón de sastre” de los mensajes del sistema.

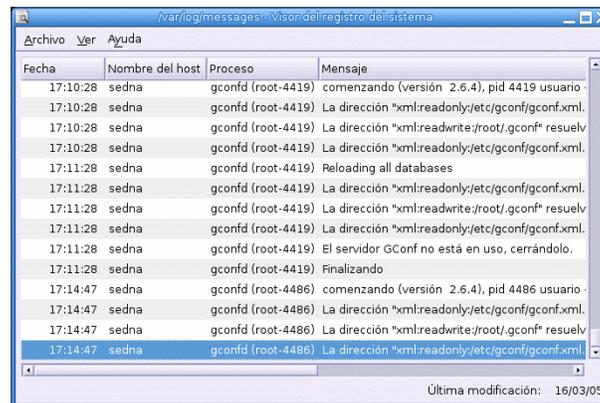
XFree86.* guarda información sobre lo que pasa con el entorno gráfico (servidor X).

...

Podemos acceder a los logs del sistema con el comando:

```
# gnome-system-log
```

o en modo gráfico con la siguiente secuencia de menús:  **Aplicaciones**→**Configuración**→**Sistema**→**Bitácora del sistema**. Pedirá la contraseña de root y mostrará el contenido del fichero `/var/log/messages`.



Podemos seleccionar **Archivo**→**Abrir** y elegir fichero de logs deseado.

➔ Para practicar

- Mostrar por pantalla las últimas 10 líneas del fichero `/var/log/messages`

```
#tail -f /var/log/messages
```
- Solicitar la página man de `dmesg` y luego visualizar el fichero `/var/log/dmesg`
 - Usando un editor de texto (`gedit` por ejemplo)
 - Usando una tubería (`|`) y algo de comandos:

```
$ dmesg | less
```

para salir hay que pulsar `q`
- Utilizar el bitácora del sistema para curiosear por los logs ■

2.4. Creación de un Sistema de Ficheros.

Como ya hemos comentado anteriormente, los sistemas de ficheros se crean sobre las particiones del disco duro. Una vez instalado el sistema, disponemos de varias utilidades para poder trabajar sobre las particiones. 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`. Pero si además de todo eso deseamos modificar el tamaño de una partición tendremos que usar `qtparted`. La sintaxis básica de uso es

¹⁵MTA: agente para el transporte de correo electrónico



similar en los dos y consiste en ejecutar el programa (como root) pasándole como argumento la unidad de disco con la que se va a trabajar (`/dev/hdx`)¹⁶.

Los sistemas de ficheros se crean con el comando `mkfs` (*Make Filesystem*). La sintaxis¹⁷ 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 tomará por defecto `ext2`.

`sistema_ficheros` es el único argumento obligatorio, y corresponde al dispositivo o a la partición del disco sobre la cual queremos crear el sistema de ficheros (`/dev/hda1`, `/dev/sda3`, `/dev/fd0`, etc).

Por ejemplo, la sentencia

```
# mkfs -t ext3 /dev/sda1
```

nos creará un sistema de ficheros tipo `ext3`, que ocupará todo el espacio de la partición `/dev/sda1`. Una vez creado el sistema de ficheros, podemos almacenar datos en él.

➔ Para practicar

1. Crear un sistema de ficheros de tipo `vfat` en un disquete (`/dev/fd0`), con la orden:

```
# mkfs -t vfat /dev/fd0
```

2. Comprobar que podemos conseguir lo mismo usando el comando¹⁸

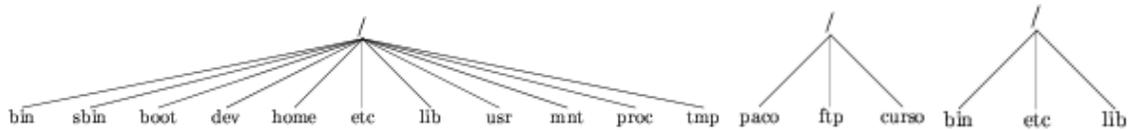
```
# mkfs.vfat /dev/fd0
```

■

2.5. Montaje y Desmontaje.

El concepto de montaje deriva del hecho de que en Unix/Linux todos los elementos son ficheros, incluidos los dispositivos. Para poder usar un sistema de ficheros éste tiene que estar *montado*. 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 se representa con el símbolo `/`. Este sistema de ficheros se monta al arrancar el sistema y ocupa el lugar más alto¹⁹.

Por ejemplo, supongamos que en tres particiones de nuestro disco, `/dev/hda1`, `/dev/hda2` y `/dev/hda3`, hemos creado tres sistemas de ficheros. Cada uno de ellos visto de forma independiente posee una raíz, que podemos esquematizar con el gráfico de abajo.



¹⁶`# fdisk -l /dev/hda` muestra la tabla de particiones del disco duro `hda`.
`# fdisk /dev/hda` solicita teclear `m` para mostrar ayuda, ¡cuidado con las opciones!
`# qtparted /dev/hdb` muestra la tabla de particiones con la posibilidad de redimensionarlas. Ya hicimos uso del programa en el proceso de instalación.

¹⁷No 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 ...

¹⁸De igual manera existen los scripts: `mkfs.ext2`, `mkfs.ext3`, ...

¹⁹En él se encuentra el “corazón” de nuestro sistema.

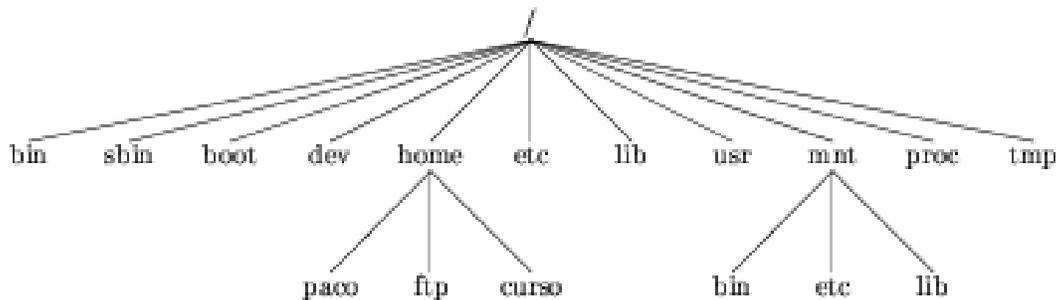


Cuando montamos un sistema de ficheros, indicamos un directorio del sistema de ficheros “padre”, del que va a colgar la estructura de directorios del sistema de ficheros “hijo”.

Por ejemplo, mediante las órdenes de montaje²⁰ 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`. Ahora ya podríamos acceder a cualquier dirección de este árbol de directorios.

2.5.1. El fichero `/etc/fstab`

El fichero `/etc/fstab` contiene información descriptiva de los distintos sistemas de ficheros del equipo. Este fichero es de solo lectura y debe ser mantenido por el administrador del sistema. La información de cada sistema de ficheros ocupa una línea de `/etc/fstab`, 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.

directorio Es el segundo campo y especifica el punto de montaje para este 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.²¹

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
users → Se permite a los usuarios desmontar la partición aunque no la hayan
montado
nouser → Sólo el root puede montar esta partición.
ro → Partición de sólo lectura.

```

²⁰La sintaxis la estudiaremos un poco más adelante.

²¹La orden para crear un subdirectorio es `mkdir`



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 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 2.6) 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.

El que sigue es un ejemplo de fichero `/etc/fstab`²² :

```

#_/_etc/fstab:_static_file_system_information.
#
#_The_following_is_an_example._Please_see_fstab(5)_for_further_details.
#_Please_refer_to_mount(1)_for_a_complete_description_of_mount_options.
#
5 #_Format:
#_<file_system>_<mount_point>_<type>_<options>_<dump>_<
    pass>
/dev/hda1_/_ext3_defaults , errors=remount-ro_0_1
proc_/_proc_/_proc_defaults_0_0
10 none_/_sys_/_sysfs_defaults_0_0
none_/_proc/bus/usb_/_usbdevfs_/_rw_0_0

/_swapfile_/_none_/_swap_sw_0_0

15 /dev/fd0_/_floppy_/_vfat_defaults , users , sync , noauto , showexec , umask=022_0_0

/dev/cdrom_/_cdrom_/_iso9660_defaults , ro , user , noexec , noauto_0_0
/tmp/image.iso_/_mnt/iso_/_iso9660_defaults , users , noauto , noexec , nosuid , loop_0_0
/dev/hda1_/_mnt/Linux1_/_auto_noauto , user , exec_0_0
20 /dev/sda_/_mnt/usb1_/_vfat_/_rw , users , sync , noauto_0_0
_/_0_0
/dev/sda1_/_mnt/usb2_/_vfat_/_rw , users , sync , noauto_0_0
_/_0_0
/dev/sdb_/_mnt/usb3_/_vfat_/_rw , users , sync , noauto_0_0
_/_0_0
/dev/sdb1_/_mnt/usb4_/_vfat_/_rw , users , sync , noauto_0_0
_/_0_0

```

Listado 2.1: `/etc/fstab`

²²El vuestro no tiene por qué coincidir, pero hemos dejado las opciones que el sistema pone por defecto.



2.5.2. El comando mount

El comando `mount` permite montar los sistemas de ficheros y anclarlos a puntos de montaje (directorios). A partir de este momento podremos acceder a ellos²³. La sintaxis más usual es:

```
mount [-t vfstype] [device] dir
```

Si montamos una unidad que está incluida en el fichero `/etc/fstab` y se omiten los argumentos `[-t vfstype]` 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 /cdrom24
```

y si lo que queremos es montar un disquete formateado bajo Windows basta con ejecutar

```
$ mount /floppy
```

El mismo comando hubiese montado un disquete de tipo `ext2`, si en la entrada correspondiente a tipo de sistema de ficheros de la línea asociada a `/dev/fd0` hubiésemos puesto **auto**.

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 su página man.

```
$ man mount
```

2.5.3. 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.

Por ejemplo, para desmontar la unidad de CD:

```
$ umount /cdrom
```

Cuando se monta una unidad de CD, el sistema no permite extraer el disco hasta que no se haya desmontado. Si lo que hemos montado es una unidad de disquete o “lápiz” usb, es fundamental desmontarlos antes de extraerlos. No olvidemos que el sistema es multitarea y puede haber operaciones sobre el dispositivo pendientes de realizarse aunque ya la hayamos ejecutado.

Sólo puede desmontar un dispositivo el usuario que lo montó o el root.



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 se utiliza la orden `fuser`. Por ejemplo, si en `/mnt/Windows9X1` tenemos montado nuestro windows y el sistema no nos deja desmontarlo, podemos ver de qué procesos está utilizando este sistema de ficheros con la orden:

```
$ /sbin/fuser -a /mnt/Windows9X1
```

²³También podemos montar en modo gráfico los sistemas de ficheros especificados en `/etc/fstab` haciendo doble clic sobre el icono [**E**quipo] y de nuevo doble clic sobre el dispositivo deseado.

²⁴Notad que la orden la puede ejecutar un usuario normal porque el dispositivo `/dev/cdrom` tiene activa la opción `user`.

Con la orden:

```
# umount -a
```

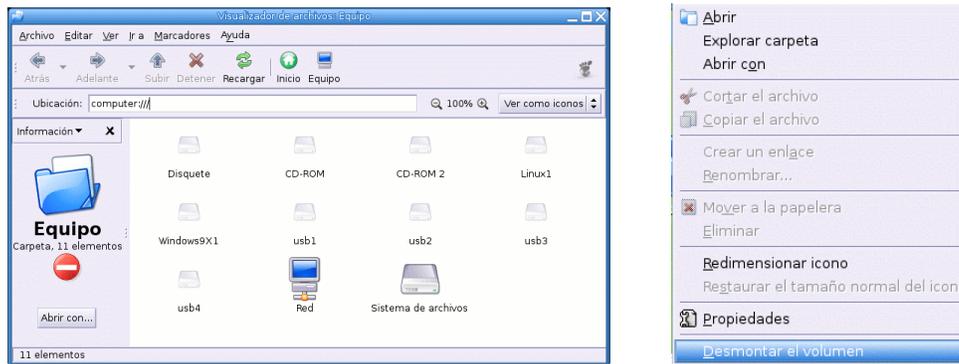
serán desmontados todos los sistemas de ficheros contenidos en `/etc/mtab`. Con la opción `-t` le podemos decir a `umount` que desmonte solo los sistemas de ficheros de un tipo en concreto (o varios, separando los argumentos por comas). Como siempre, no está de más echar un vistazo a

```
$ man umount
```

2.5.4. Herramientas gráficas para montar dispositivos

El montaje y desmontaje de dispositivos puede hacerse en modo gráfico. Para montar un dispositivo haga doble clic sobre el icono  del escritorio y en la ventana emergente, de nuevo doble clic sobre el dispositivo que desee.

Para desmontarlo, haga clic con el botón derecho del ratón sobre el icono del dispositivo que estará en el escritorio y en el menú emergente elija **Desmontar el volumen**.



➔ Para practicar

1. Probar a:
 - a) Crear en un disquete un sistema de ficheros de tipo `vfat`.
 - b) Montar el disquete y copiar en él el fichero `/etc/hosts`
 - c) Desmontar el disquete.
 - d) Comprobar que todo ha salido bien (habrá que montarlo de nuevo y visualizar su contenido)
2. Modificar el fichero `/etc/fstab` para que el sistema monte la partición Windows en el arranque. Se trata de practicar, después de probar que funciona, deja las cosas como estaban.
3. Con el método de instalación de Guadalinex2004, todo el sistema y nuestros documentos están en una misma partición (aunque en directorios bien diferenciados) ¿Qué tendríamos que hacer si necesitamos reinstalar el sistema?
Así que para los más atrevidos os proponemos crear una nueva partición del tipo `ext3`, que montaremos en el arranque sobre el directorio `/home` del directorio raíz. Vayamos por partes (en toda la práctica, como es lógico trabajaremos como `root`):
 - a) Redimensionar la partición windows (por ejemplo quitando 2 GB) y en el espacio liberado crear la nueva partición²⁵. Para ello ejecutar:


```
# qtparted /dev/hda
```

 Hemos supuesto que se va a hacer en este disco y que la partición creada es la `/dev/hda4`.

²⁵ de forma similar a como se hizo en el proceso de instalación



- b) Hacer una copia del contenido actual del directorio `/home`²⁶. Para ello situaros en este directorio,
- ```
cd /home
y
tar -cvzf copiahome.tar.gz *
```
- c) Crear el directorio `/copiahome` por ejemplo en `/mnt` y mover allí el fichero creado:
- ```
# mkdir /mnt/copiahome
# mv copiahome.tar.gz /mnt/copiahome
# cd /mnt/copiahome      y comprobar que allí está el fichero copia:
# ls -l
```
- d) Ahora montamos la nueva partición, copiamos en ella el fichero `copiahome.tar.gz`, lo descomprimos y lo desempaquetamos:
- ```
mount -t ext3 /dev/hda4 /home
cp copiahome.tar.gz /home
cd /home
tar -xvzf copiahome.tar.gz
```
- Ya hemos recuperado nuestro antiguo `/home` pero ahora físicamente está en una partición independiente. Ahora ya solo queda modificar el fichero `/etc/fstab` para que la monte en el arranque, así que editamos el fichero e incluimos una línea como ésta:
- e) `/dev/hda4 /home ext3 auto,user,exec,rw 0 0`
- La próxima vez que arranquemos el sistema todo parecerá igual que al principio pero el directorio `/home` está en una partición independiente y si tenemos que reinstalar el sistema ya no tendremos que preocuparnos de salvar antes nuestros documentos ■

## 2.6. Chequeo y recuperación: fsck.

El comando `fsck` nos permite chequear y, opcionalmente, reparar un sistema de ficheros (`ext3` o `ext2`). Repara inconsistencias en los sistemas de ficheros después de un apagado incorrecto de la máquina.

Si trabajamos con el sistema de ficheros `ext3`<sup>27</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>28</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 y el único parámetro obligatorio. Por ejemplo:

```
/sbin/fsck /dev/hda2
```

<sup>26</sup>lo haremos con un `tar.gz` para mantener los permisos.

<sup>27</sup>Para saber más sobre el sistema `ext3`: <http://linuxmobile.sourceforge.net/recursos/documentacion.html>

<sup>28</sup>Por ejemplo un apagón de luz.

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.



En los Guadalinux EDU de los centros no TIC se trabaja con el sistema de ficheros `ext2`<sup>29</sup>. Esto implica que cada vez que se apaga mal el sistema se queda “sucio” y tenemos que chequearlo de forma manual. El comando (más cómodo) a usar cuando ocurre esto es:

```
fsck -y /dev/hda1; reboot
```

En realidad se trata de dos comandos que se ejecutan de forma consecutiva, primero chequeamos el sistema diciendo que sí a todo (`-y`) y después reiniciamos el sistema (`reboot`). Si se trata de una máquina dual (con windows y linux) lo usual es que tengamos que sustituir `hda1` por `hda2`

Para más información sobre las opciones consultar el manual en línea.

### ➔ Para practicar

1. Chequear el disquete creado en el **Para practicar** anterior.
2. Si hemos hecho el apartado 3 de la práctica anterior, chequear la partición `/home`:
  - a) Si no sabemos cuál es podemos usar cualquiera de los comandos que siguen:
 

```
$mount
/sbin/fdisk -l /dev/hda
```
  - b) Ya es sencillo, usar `fsck` tal cual se ha explicado ■

## 2.7. Enlaces

El nombre de un fichero no es más que una etiqueta que referencia a un número, **inodo**<sup>30</sup>, que a su vez apunta al lugar físico donde se almacena la información que contiene el fichero.

Los enlaces permiten dar múltiples nombres a un fichero.

**Enlaces duros (*hard link*)** Un enlace duro es un nombre adicional para un fichero ya existente. Se crea con la orden `ln`. Por ejemplo, la orden :

```
$ ln notas.txt notashard.txt
```

crea un “nuevo fichero” de nombre `notashard.txt` cuyo contenido es el mismo que el del fichero `notas.txt`.

La salida de la orden

```
$ ls -li notas*
```

<sup>29</sup>En los centros TIC esto ya no es así debido a que se ha actualizado el sistema. Inicialmente también ocurría.

<sup>30</sup>Cada sistema de ficheros tiene su propia tabla de inodos.

es:

```
1289347 -rw-r--r-- 2 juan users 82 2004-02-29 11:33 notashard.txt
1289347 -rw-r--r-- 2 juan users 82 2004-02-29 11:33 notas.txt
```

Podemos observar que ambos ficheros tienen los mismos permisos y el mismo número de inodo, el que aparece delante de los permisos. Son en realidad el mismo fichero con dos nombres distintos. Si hacemos cambios en uno de ellos, se reflejará en el otro. Si borramos por ejemplo `notas.txt`, `notashard.txt` seguirá existiendo, en realidad lo que hemos borrado ha sido una de las referencias al inodo, la otra sigue existiendo. El 2 delante de `juan` indica que el fichero tiene dos enlaces.

No se pueden realizar enlaces hard entre ficheros de dos sistemas de ficheros distintos ni entre directorios.

**Enlaces simbólicos** Los enlaces simbólicos son ficheros que únicamente contienen el nombre de otro fichero<sup>31</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 /etc/fstab fstabsimb
```

Crea el fichero `fstabsimb` que apunta al fichero `/etc/fstab`

La salida de la orden

```
$ ls -li fstabsimb /etc/fstab
```

es:

```
310523 -rw-r--r-- 1 root root 1049 2004-03-08 17:55 /etc/fstab
1290242 lrwxrwxrwx 1 juan users 10 2004-03-08 18:16 fstabsimb -
> /etc/fstab
```

Nos muestra que:

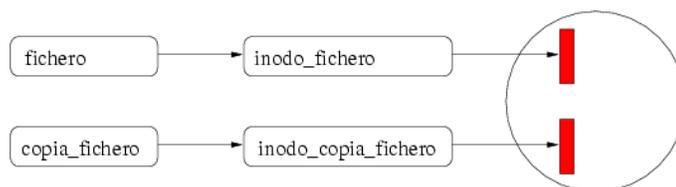
- Los ficheros `fstabsimb` y `/etc/fstab` tienen inodos distintos.
- Los permisos del enlace simbólico están todos activos, en realidad los permisos que imperan son los del fichero “apuntado”.
- El listado nos muestra también hacia quien apunta el fichero `fstabsimb`.

Como en los enlaces duros, si modificamos uno se modifica el otro<sup>32</sup> y si borrásemos el “apuntado” el fichero simbólico seguiría existiendo pero sin apuntar a nadie.

Los enlaces simbólicos son muy utilizados por el sistema, fundamentalmente para enlazar con las imágenes de las librerías compartidas en `/lib`, pues este tipo de ficheros facilita la actualización del sistema.

El siguiente gráfico puede clarificar la diferencia entre copiar ficheros y los dos tipos de enlaces.

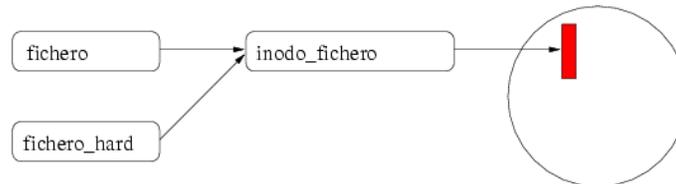
- Copiar fichero



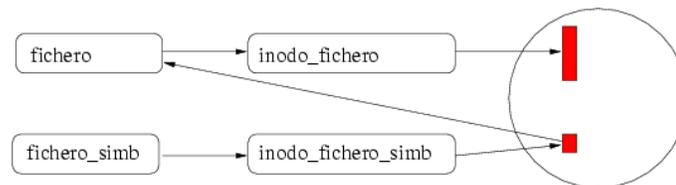
<sup>31</sup>En cierto sentido se parecen a los “accesos directos” de Windows.

<sup>32</sup>En el ejemplo, como usuario, no podremos modificar `fstabsimb` porque no tenemos permiso sobre `/etc/fstab`

- Enlace duro



- Enlace simbólico



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.

➔ **Para practicar**

1. Crear un enlace simbólico de nombre `logsimb` en nuestro directorio de usuario que apunte al directorio `/var/log/`:  
`$ ln -s /var/log/ logsimb`
  - a) Comprobar los efectos de este enlace (`logsimb` apunta a un directorio).
  - b) Borrar el enlace y crear uno para acceder al directorio `/tmp`
2. Crear otro enlace que acceda al fichero `/var/log/debug`. Edita con `gedit` el nuevo fichero.
3. Si tienes también windows en el equipo, monta la partición de windows y crea el enlace simbólico `misdocsimb` al directorio Mis documentos<sup>33</sup> de windows ■

<sup>33</sup>para escribir la cadena `/mnt/Windows9X1/Mis...` aprovecha la facilidad de “autocompletación” de nombres pulsando el tabulador. De paso observaremos como habría que introducir el carácter “espacio”.

# Capítulo 3

## Permisos. Gestión de Usuarios

Complots apocalípticos futurísticos aparte, sin los usuarios, los ordenadores no tienen objeto alguno. (*Red Hat Linux 7*, CHRISTOPHER NEGUS)

### 3.1. Introducción.

Linux es un sistema operativo multitarea y multiusuario. Esto posibilita que varios usuarios ejecuten distintas tareas a la vez, y en consecuencia, se hace necesario establecer algún mecanismo que proteja la información personal.

La forma que tiene el sistema de identificar a los usuarios es mediante la asignación de *cuentas de usuario*. Cada usuario dispone de un nombre de usuario que lo identifica y además puede pertenecer a uno o varios grupos<sup>1</sup>. En un sistema Linux, aunque lo utilice una sola persona, deberíamos tener al menos dos cuentas, una para el root, que debe utilizarse sólo cuando vayamos a efectuar labores de administración, y otra como usuario de “a pie” para la realización del trabajo cotidiano.

La identidad de cada usuario junto con el grupo al que pertenece determina los derechos de acceso a ficheros y otros recursos del sistema.

### 3.2. Permisos de acceso a los distintos objetos.

En UNIX cada fichero tiene un propietario (inicialmente el usuario que lo creó) y pertenece a un grupo en particular. Basado en esta estructura, el sistema asigna permisos sobre el fichero a tres categorías de usuarios:

1. para el propietario del fichero,
2. para el grupo al que pertenece el fichero, y
3. para todos los demás usuarios.

Para cada una de ellas, se asignan tres tipos de permisos básicos:

**r** de lectura

**w** de escritura, y

**x** de ejecución.

---

<sup>1</sup>al menos a un grupo con su mismo nombre de usuario que se crea al darle de alta.



Esta información está guardada en el inodo<sup>2</sup> del fichero. Se utilizan los primeros 9 bits (del 0 al 8) para los 9 permisos. Estos permisos, así como el tipo de objeto, pueden ser visualizados con la opción `-l` (formato largo) del comando `ls`. Supongamos que la salida del comando:

```
$ ls -l /home/mercedes/linux2004/entrega1.lyx
```

es

```
-rwxr-xr-- 1 mercedes gelinux 386 2003-9-1 22:37 /home/mercedes/linux2004/entrega1.lyx
```

Analicemos parte de esa salida: `-rwxr-xr--`

Para ello la dividimos en cuatro bloques: `- rwx r-x r--`

- El primer guión por la izquierda (`-`) nos indica que `entrega1.lyx` 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
- Los tres caracteres siguientes (`rwx`) indican que este fichero tiene permisos de lectura, escritura y ejecución para el propietario que en este caso es `mercedes`,
- Los que siguen (`r-x`) indican que el fichero tiene permisos de lectura y ejecución para el grupo `gelinux`, y
- El último grupo (`r- -`) indica que el resto de usuarios tan sólo podrán leer el fichero pero no modificarlo ni ejecutarlo.

El significado de los permisos `rwx`, cuando el objeto es un directorio, es el siguiente:

- r** Permite leer el contenido del directorio, es decir los nombres de los ficheros y sus inodos, pero no la información de estos (con ese permiso nada más, puede ejecutarse el comando `ls` pero no `ls -l`).
- w** Permite escribir en el directorio, es decir crear y suprimir ficheros, otros subdirectorios, etc.
- x** Permite recorrer el directorio y utilizar la información de los objetos del directorio, es decir, acceder a los inodos (se podría ejecutar `ls -l`).

Es importante resaltar que los permisos de un fichero están condicionados por los permisos del directorio donde reside. Por ejemplo, aunque un fichero tenga los permisos `-rwxrwxrwx`, otros usuarios no podrían acceder a él a menos que tengan permiso de ejecución para el directorio en el que se encuentra el fichero. Así mismo, si un directorio tuviera permiso `w` para “otros usuarios” estos podrían borrar ficheros de ese directorio aunque los ficheros tuvieran deshabilitado ese permiso. Habitualmente los usuarios suelen dar a sus ficheros los permisos `-rwxr--r--` y a los directorios `-rwxr-xr-x`.



El comando `chown` (*change owner*- cambiar propietario) se utiliza para cambiar el propietario y el grupo de un fichero<sup>3</sup>; con `chgrp` (*change group*- cambiar grupo) podemos modificar sólo el grupo. Los permisos pueden modificarse con el comando `chmod` (*change mode*- cambiar

<sup>2</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, etc.)

<sup>3</sup>ver la página man de `chown`.



modo). Naturalmente estas acciones sólo le están permitidas al root y al dueño del objeto. La sintaxis básica de las dos primeras es:

```
chown usuario fichero
```

```
chgrp grupo fichero
```

para cambiar el usuario o el grupo a un fichero. Después veremos cómo hacerlo en modo gráfico.

### 3.2.1. chmod

La sintaxis básica es<sup>4</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: notación octal y notación nemónica. En la notación octal el primer dígito determina los permisos del propietario, el segundo los del grupo, y el tercero los de 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 `entrega1.lyx` los permisos: `-rwxr-x--x` utilizando la notación de la tabla, introduciríamos:

```
$ chmod 751 entrega1.lyx
```

#### umask

Por último, comentar que los permisos por defecto de los objetos dependen del programa que los crea y del valor de la variable `user mask`. Su valor para todos los usuarios se define en `/etc/profile` y podemos visualizarlo editando el fichero o con la orden :

```
$ umask
```

No obstante cada usuario puede modificarla de forma transitoria con la orden

```
$ umask valorumask
```

y de forma permanente en su fichero `.bash_profile`.

Esta variable actúa como se explica en el siguiente ejemplo.

↪ **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.

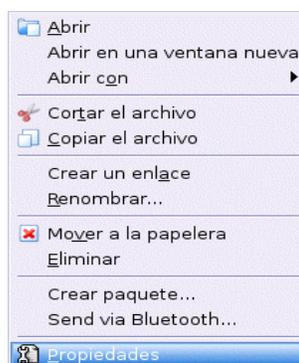
<sup>4</sup>El parámetro opcional `-R` (recursivamente) es interesante, permite cambiar los permisos de los ficheros de un directorio y de todos sus subdirectorios (se puede usar también con `chown` y `chgrp`). Si queréis usarlo mirad cómo se hace en la documentación en línea.

### 3.2.2. Permisos en modo gráfico

Además de los comandos ya comentados, podemos usar dos herramientas gráficas para modificar fácilmente los permisos de los ficheros:

- El gestor de ficheros **Nautilus**.

Por ejemplo, para cambiar los permisos del fichero `~/bashrc`<sup>5</sup>. Abrimos el navegador y pulsamos sobre el fichero con el botón derecho del ratón; aparecerá el menú de opciones:



la opción **Propiedades** de este menú nos permite acceder a la ventana

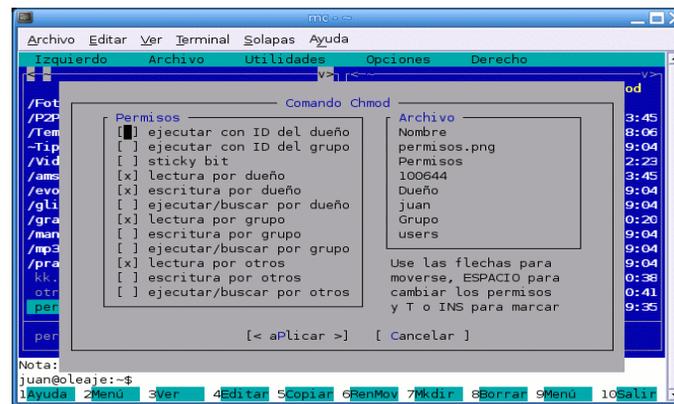


cambiar los permisos es ya un “juego de niños”. En este caso, el fichero en cuestión tendría de permisos: `-rw-r--r--`. Si modificamos algún permiso veremos como, automáticamente, cambia el número octal correspondiente al modo actual.

- El programa **mc**.

Lanzar el programa, `$ mc`, seleccionar el archivo cuyos permisos queramos modificar, pulsar en el menú **Archivo** y del menú emergente elegir cambiar **Permisos...**

<sup>5</sup>Ojo que el fichero es un fichero “punto” (oculto), o sea que hay que decirle al programa que nos muestre este tipo de ficheros: **Editar**→**Preferencias** y señalar la opción.



### 3.2.3. Más sobre permisos

En las capturas gráficas anteriores podemos 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 esté 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 (aunque sí pueda crearlos), 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. 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>6</sup>:

```
$ chmod 1750 nombre_objeto
```

- El *SUID*<sup>7</sup> le indica al kernel que el usuario que ejecute el fichero con este bit activo tome, durante la ejecución, la personalidad del usuario propietario del fichero. De esta forma se soluciona, por ejemplo, que cualquier usuario pueda modificar su clave de acceso al ejecutar el comando `passwd` (`/usr/bin/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. 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

<sup>6</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>7</sup>Los modos con valores octales 4000 y 2000 son para activar el *SUID* y el *SGID* respectivamente.



el valor del GID del propietario del programa durante el tiempo de ejecución.

Algunos sistemas (SVR4) utilizan este bit con los directorios: si un 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.

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 su significado con algunos 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 del grupo al que pertenece el fichero) como de sólo lectura. El equivalente octal sería un 4 en la posición de otros.

```
chmod u+x fichero
```

Con este comando añadimos (+) al propietario (u) el permiso de ejecución (x)

Podíamos haber hecho las dos modificaciones de una vez con la orden:

```
chmod o=r,u+x fichero
```

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.



### 3.3. Gestión de usuarios en modo texto.

El sistema, para cada usuario, mantiene cierta cantidad de información que resumimos a continuación:

**nombre de usuario** El nombre de usuario es el identificador único dado a cada usuario del sistema.

**user ID** El **UID** es un número único que el sistema asigna a cada usuario y por el que lo identifica.

**group ID** El **GID** es el número del grupo por defecto del usuario (su grupo principal).

**clave** La clave de usuario, que se almacena de forma encriptada. El comando **passwd** se utiliza para poner y cambiar la clave de los usuarios.

**nombre completo** El nombre real del usuario.

**directorio inicial** El directorio inicial de trabajo del usuario. Normalmente cuelga de **/home** con el mismo nombre que el usuario.

**intérprete de inicio** El intérprete de comandos que es arrancado para el usuario en tiempo de conexión, suele ser **/bin/bash**.

Toda esta información se almacena en el fichero **/etc/passwd**.

LINUX proporciona herramientas muy potentes para la administración de usuarios y de grupos, tanto en modo comando como en modo gráfico. Las órdenes<sup>8</sup> que usaremos para gestionar los usuarios del sistema son:

**adduser** para añadir un usuario<sup>9</sup>.

**passwd** para asignarle contraseña a un usuario o modificarla<sup>10</sup>.

**deluser** permite eliminar un usuario.

Supongamos que queremos añadir al 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:

```
adduser thales
```



Los valores por defecto se establecen en **/etc/adduser.conf**. Una nota sobre la configuración por defecto. Guadalinux, al dar de alta a un usuario, crea los **\$HOME** de usuario con permisos 755 que si bien pueden ser los adecuados para un ordenador en nuestra casa, no creemos sean los más adecuados para una máquina a la que accedan bastantes usuarios. En este caso, deberíamos cambiar la directiva **DIR\_MODE** de ese fichero para que las “casas” de los nuevos usuarios tengan unos permisos más restrictivos y seguros (por ejemplo 700)

```
DIR_MODE = 0700
```

Cuando cambiamos ese valor no afecta a los permisos de los **\$HOME** de usuario ya existentes, sólo tendrá validez para los que creamos de nuevo. Así que si deseamos cambiar los permisos del **\$HOME** de un usuario ya creado no tendremos más remedio que usar:

```
chmod 0700 /home/usuario
```

<sup>8</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>9</sup>También se puede usar **useradd** pero **adduser** es mejor como podéis comprobar.

<sup>10</sup>sólo el root o el propio usuario puede modificar su contraseña.



El sistema solicita la password (contraseña) del usuario. Como es obvio al escribir la contraseña de nuestro usuario no veremos los caracteres que introducimos y habrá que volverla a teclear para confirmarla.

A continuación nos solicita información adicional del usuario; esta información es opcional y podemos dejar campos en blanco

```
Nombre completo []: Thales Cica
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
```

y confirmamos que la información es correcta.<sup>11</sup>

El sistema habrá leído el fichero `/etc/login.defs`, en el que 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 subdirectorío `/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>12</sup>.

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:

```
thales:x:1003:1003: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 1003 (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? Ah!, ¡me salté un campo! el penúltimo campo es el subdirectorío inicial de este famoso usuario.

Pero después de un tiempo nuestro usuario ya no va a utilizar más nuestro sistema así que le daremos de baja, para ello basta con ejecutar:

```
deluser thales
```

Con esta orden se ha eliminado a nuestro usuario de los ficheros `/etc/passwd` y `/etc/shadow` pero no se ha borrado su directorio de trabajo ni algún otro asociado con sus aplicaciones. Para que también se hubiesen borrado deberíamos haber añadido la opción `-r` al comando `userdel`, así

```
userdel -r thales
```

no habría dejado rastro de nuestro usuario. Naturalmente hay otras formas de borrar esos directorios.

Si por alguna razón lo que quisiéramos fuera deshabilitar temporalmente la cuenta de un usuario, bastaría con anteponer un asterisco “\*” en el campo de la clave en `/etc/passwd`. Por ejemplo,

---

<sup>11</sup>Observa que el sistema ha incluido al usuario en los grupos `audio`, `dip`, `disk`, `floppy`, `cdrom` y otros. La razón es clara: para que el nuevo usuario pueda acceder a ellos (así podrá grabar o escuchar un CD de música por ejemplo).

Los grupos a los que se añadirán los nuevos usuarios los podemos establecer desde el fichero `/usr/local/sbin/adduser.local`

Para ampliar, consultar:

<http://www.debian.org/doc/manuals/securing-debian-howto/ch11.es.html>

[http://www.augcyl.org/glol/old/N\\_1/admbasica.html](http://www.augcyl.org/glol/old/N_1/admbasica.html)

<sup>12</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>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, 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.



```
thales:*x:1003:1003:Thales Cica, , ,:/home/thales:/bin/bash
```

impedirá que THALES entre en el sistema.

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`.<sup>17</sup>



Algunas órdenes para trabajar con grupos son:

**groups** ver los grupos a los que pertenece el usuario pasado como argumento.

**addgroup** crear un nuevo grupo

**delgroup** borra un grupo<sup>18</sup>

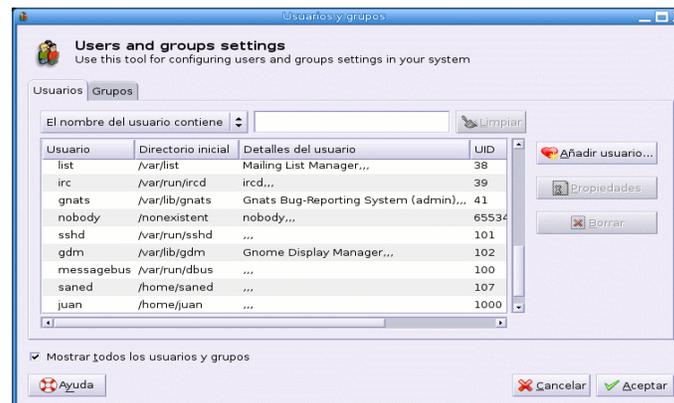
### 3.4. Gestión de usuarios en modo gráfico.

El sistema proporciona una herramienta gráfica para la gestión de usuarios y grupos que no puede ser más fácil de utilizar. Se accede a ella con:



Aplicaciones→Configuración→Sistema→Usuarios y grupos<sup>19</sup>

o desde un terminal con la orden `# users-admin`.



#### ➔ Para practicar: Varias sesiones abiertas

Hemos dicho muchas veces que Linux es multiusuario pero hasta ahora no hemos utilizado esta posibilidad para nada<sup>20</sup>. En esta práctica haremos ver que podemos estar trabajando varios usuarios a la vez.

Supongamos que hasta ahora sólo están registrados el root y el usuario de “a pie” *nano*, que ha iniciado el sistema en modo gráfico. Vamos a dar de alta a un nuevo usuario de nombre *thales*, así que desde una terminal gráfica ejecutamos:

```
$ su
```

```
$ Password:
```

<sup>17</sup>De nuevo si se desea ampliar, consultar las Web ya comentadas:

<http://www.debian.org/doc/manuals/securing-debian-howto/ch11.es.html>

[http://www.augcyl.org/glol/old/N\\_1/admbasica.html](http://www.augcyl.org/glol/old/N_1/admbasica.html)

<sup>18</sup>es una buena idea mirar las páginas man tanto de los comandos de grupo como de los de usuarios.

<sup>19</sup>En modo gráfico no se añade de forma automática los nuevos usuarios a los grupos `cdrom`, `audio`, `dialout`, `tec`, así que habría que haberlo hecho posteriormente.

<sup>20</sup>Hemos utilizado el comando `su` para cambiar momentáneamente de “personalidad” pero sigue habiendo solamente un usuario haciendo uso del sistema.



y tras introducir la clave de root,

```
adduser thales
```

Tras introducir la password del usuario thales y otros datos adicionales,

```
exit
```

Habíamos iniciado el sistema en modo gráfico como usuario *nano* pero *thales* necesita hacer uso del él con todos sus privilegios (no de “prestado” \$ su *thales*). Para ello pulsamos [CTRL]+[ALT]+[F1] y tras pulsar [Intro] aparece

```
login:
```

Tras introducir su nombre y contraseña, thales ha entrado en modo texto. Pero quiere trabajar en modo gráfico así que teclea:

```
$ startx -- :1
```

y ¡observa!, aparece un nuevo *display*<sup>21</sup> gráfico para él.

Para pasar de nuevo al display de *nano*, pulsamos [CTRL]+[ALT]+[F7] y para volver al de thales [CTRL]+[ALT]+[F8]. Ambos pueden estar ejecutando aplicaciones, mientras uno descarga un fichero de tamaño considerable el otro puede mirar su correo.

Pero ahora queremos trabajar como *root* para hacer ciertas labores de administración:

Pulsamos [CTRL]+[ALT]+[F2] y de nuevo, tras introducir el login y la password de root, hemos entrado en modo texto (pueden efectuarse tareas en modo texto), pero nos gusta más el modo gráfico, así que ahora:

```
startx -- :2
```

y tenemos un nuevo display para el root. Con [CTRL]+[ALT]+[F7] volvemos al de *nano* y con [CTRL]+[ALT]+[F9] al del *root*. Con [CTRL]+[ALT]+[F1] al modo texto de *thales* y con [CTRL]+[ALT]+[F2] al modo texto de *root*.

Tenemos a nuestra disposición 6 terminales de texto [CTRL]+[ALT]+[Fi], i=1 ... 6 y otros 6 gráficos [CTRL]+[ALT]+[Fj], j=7 ... 12 y ya hemos visto cómo movernos de uno al otro ■

## ➔ 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/alumnoi`. Dentro de él, deben crear el fichero `practica_individuali`. Ese fichero podrá ser visto por el profesor pero no podrá modificarlo. Cada alumno sólo tendrá acceso a sus ficheros.

<sup>21</sup>Se llama así al conjunto formado por un monitor, un teclado y un ratón. El display 0 pertenece al usuario *nano*, de ahí que la llamada `startx` deba llevar el modificador `-- :1` (los espacios son necesarios)



2. Para las prácticas en grupo cada grupo dispondrá de su 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 pero no pueden leer ni escribir en el directorio de otros grupos. El profesor podrá ver los ficheros de este directorio pero no modificarlos.
3. Habrá un directorio `/home/clase_linux` en el que todos los alumnos tendrán un fichero de nombre `alumnoi`, que sólo podrá modificar dicho alumno, pero podrá ser leído por todos los demás.

Tenemos que ver qué permisos deben tener los ficheros y directorios para conseguir las condiciones anteriores.

El problema se puede resumir en determinar:

1. Permisos del directorio `/home/alumnoi` y del fichero `/home/alumnoi/practica_individuali` 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/grupo1` y del fichero `/home/grupo1/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 `alumnoi` dentro de ese directorio.

## Solución

### Prácticas individuales.

#### 1. Creación de los usuarios:

Creamos el usuario profesor mediante el método que preferamos. Por ejemplo:

```
#adduser profesor22
```

Después creamos los distintos usuarios-alumnos<sup>23</sup>:

```
#adduser alumnoi (i:1..4)
```

Cada usuario que creamos pertenece a un grupo particular, de nombre el del propio 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 r-x r-x**

Hacemos que el profesor pertenezca al grupo `alumnoi`, editando y añadiéndolo en la entrada correspondiente al grupo `alumnoi` del fichero `/etc/group`<sup>24</sup>

```
alumno1:x:1001:profesor
```

```
...
```

```
alumno4:x:1004:profesor
```

#### 2. Modificación de los permisos:

Para ver los permisos de los distintos directorios de usuario lo haremos con la orden

```
#ls -l /home
```

Si los permisos del directorio cuando se crea son:

```
drwxr-xr-x 5 alumnoi alumnoi 4096 oct 9 20:33 alumnoi
```

Los modificaremos con

```
#chmod o-rx /home/alumnoi
```

quedando:

```
drwxr-x--- 5 alumnoi alumnoi 4096 oct 9 20:33 alumnoi
```

Los permisos del usuario no necesitan comentario. Para el grupo `alumnoi` (al cual el profesor

<sup>22</sup>La creación de usuarios y grupos que en la práctica hacemos en modo comando, puede hacerse en modo gráfico.

<sup>23</sup>Para cada usuario podemos asignar o cambiar las contraseñas de acceso con el comando

```
#passwd usuario
```

<sup>24</sup>Se puede en modo texto, como acabamos de decir o con la herramienta gráfica que hemos visto. Los UID de los usuarios no tienen por qué corresponder con los que aparecen a continuación.



pertenece) se permite **r** (ver qué ficheros hay en el directorio) y **x** (poderse meter dentro del directorio. Hacer `cd /home/alumnoi`). 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 de cada uno de los alumnos

Mas tarde entramos al sistema como alumno1 y, por ejemplo con gedit, creamos el fichero `practica_individual1`, cuyo contenido ahora es lo de menos.

```
$ gedit 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_individual1
```

### Práctica de grupo

Creamos los grupos

```
#addgroup grupo1
```

```
#addgroup grupo2
```

y añadimos en el fichero `/etc/group` a alumno1 y alumno2 como usuarios pertenecientes al grupo1 (lo mismo haremos con los alumnos 3 y 4 para el grupo 2)<sup>25</sup>.

```
grupo1:x:1006:alumno1,alumno2
```

Podemos comprobar cómo 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 (lo mismo para el grupo2) 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).

Pero los permisos por defecto son:

```
drwxr-xr-x
```

Para cambiarlos y conseguir el efecto deseado, ejecutaremos la orden:

```
#chmod 770 /home/grupo1
```

<sup>25</sup>El GID de ejemplo no tiene por qué ser el que aparezca en vuestro fichero.



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:

```
$ls -l /home
```

```
...
```

```
drwxrwxrwx 2 profesor profesor 4096 oct 9 23:12 clase_linux
```

```
...
```

Si dejamos así los permisos del directorio clase\_linux, los alumnos (mediante el **rw**x final del resto de usuarios) pueden crear sus ficheros alumnoi, pero otros alumnos podrán borrarlos, 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**x para todos) solamente el propietario del fichero podrá borrarlo.

```
$ls -l /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>26</sup>. Además, los permisos de este fichero impiden que pueda modificarlo otro alumno ■

---

<sup>26</sup>Es lo que pasa en el directorio `/tmp` de uso público.

## Capítulo 4

# Instalación, desinstalación de paquetes y actualización del sistema

¿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*

### 4.1. apt - Introducción

En este capítulo vamos a estudiar cómo instalar y desinstalar programas. En GuadaLinux existen diferentes métodos de instalación y desinstalación de paquetes

Estudiamos el comando `apt` aparte de los otros comandos debido a la “importancia” merecida que tiene. Este sistema es muy avanzado y de gran flexibilidad y potencia para entornos de red. 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 `apt`, su manejo es casi inmediato.

Se puede ampliar sobre el tema en:

- La página man del programa<sup>1</sup>.
- Y sobre todo la web: <http://www.debian.org/doc/>

#### 4.1.1. ¿Qué es apt?

APT son las siglas de *Advanced Packaging Tool*, es decir, *herramienta avanzada de empaquetamiento*. El sistema APT es un sistema abierto, basado en la licencia GNU y desarrollado por el ATP Team. Este sistema fue adaptado por la distribución Conectiva para poder usarse con `rpm`<sup>2</sup> y ha sido adoptado por otras distribuciones como:

- Conectiva <http://www.conectiva.com.br>
- Fedora <http://fedora.redhat.com/>
- Mandrake <http://www.mandrake.com>

---

<sup>1</sup>Recuerda: `$ man apt`

<sup>2</sup>RPM son las siglas de *RedHat Package Management*, es decir, sistema de paquetes de RedHat



- PLD <http://www.pld.org.pl>
- Vine <http://www.vinelinux.org>
- APT4RPM <http://apt4rpm.sf.net>
- Alt Linux <http://www.altlinux.ru>
- Red Hat <http://www.redhat.com>
- Sun Solaris <http://www.sun.com>
- SuSE <http://www.suse.de>
- Yellow Dog Linux <http://www.yellowdoglinux.com>

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 de los sistemas de paquetes son<sup>3</sup>:

- Mantienen 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>4</sup>
- Control sobre dependencias: controlando las dependencias antes de instalar o desinstalar un paquete sabemos si ese paquete es usado por otros paquetes o librerías para funcionar correctamente. Si no tenemos instalado ese programa/librería, antes de instalar se nos avisará. Lo mismo sucederá 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>5</sup>

El sistema de paquetes APT sigue una “nomenclatura” que permite identificarlo, se basa en dar de cada paquete los campos: nombre, versión, revisión, plataforma y extensión. Por ejemplo, consideremos el paquete<sup>6</sup>

```
acroread_4.05-3_i386.deb
```

veamos qué significa cada uno de esos campos:

**acroread** es el nombre del paquete.

**4.05** es el número de versión.

---

<sup>3</sup>RPM o deb

<sup>4</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>5</sup>Como en otros sistemas operativos, ¿verdad?

<sup>6</sup>Este paquete instala el visor Adobe Acrobat Reader.

-3 número de la revisión, en este caso indica que es la segunda modificación realizada.

**i386** nos indica la plataforma 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.

**deb** La extensión común a todos los paquetes *Debian*.

#### 4.1.2. El archivo `/etc/apt/sources.list`

Parte fundamental del funcionamiento de `apt`, es el archivo en que están localizadas las “fuentes”<sup>7</sup> en donde se encuentran los paquetes. Este archivo es:

`/etc/apt/sources.list`

Si miramos en nuestro GuadaLinux nos aparecerá:

```

#_Junta_de_íAndaluca_(Repositorio_raz)
#_éMtodo_HTTP

deb_http://http.guadalinex.org/debian_sarge_main_contrib_non-free
5 deb_http://http.guadalinex.org/debian-non-US_sarge/non-US_main_contrib_non-
 free
deb_http://http.guadalinex.org/debian-security_sarge/updates_main_contrib_
 non-free
deb_http://http.guadalinex.org/repositorio_muflon_guada

#_Fuentes
10 #_deb-src_http://http.guadalinex.org/debian_sarge_main_contrib_non-free
#_deb-src_http://http.guadalinex.org/repositorio_muflon_guada

#_éMtodo_FTP
#deb_ftp://ftp.guadalinex.org/repositorio_muflon_main_contrib_non-free_guada
15 #deb_ftp://ftp.guadalinex.org/repositorio_muflon/non-US_main_contrib_non-
 free

#_Mirror_Oficial_de_Guadalinex:_Centro_áInfortmico_íCientfico_de_íAndaluca_(
 CICA)
#deb_ftp://ftp.cica.es/debian_sarge_main_contrib_non-free
#deb_ftp://ftp.cica.es/guadalinex/repositorio_muflon_guada
20

#_Mirror_Oficial_de_Debian
#_Sarge
#deb_http://ftp.fi.debian.org/debian_sarge_main_contrib_non-free
#deb_http://ftp.fi.debian.org/debian-security_sarge/updates_main_contrib_non-
 free
25 #deb_http://non-us.debian.org/debian-non-US_sarge/non-US_main_contrib_non-
 free

```

Listado 4.1: `/etc/apt/sources.list`

Todas las líneas de este fichero están comentadas<sup>8</sup> (están encabezadas por el símbolo “#”) salvo cuatro. Estas líneas comienzan con **deb**, esto quiere decir que será en los “sitios” que figuran a continuación donde buscará los paquetes binarios (deb) a descargar. Estos paquetes ya están pre-compilados y son los que normalmente se usan.

<sup>7</sup>Máquinas de internet a las que nos conectaremos para bajar los programas.

<sup>8</sup>El que una línea de un fichero aparezca comentada, #, quiere decir que lo que figure a continuación será ignorado, no se leerá ni ejecutará.



Aparecen otras líneas comentadas encabezadas con `deb-src`, no debemos quitar el “#” salvo que deseemos descargar los paquetes fuente. Estos paquetes son los códigos originales con los que está hecho un programa<sup>9</sup>.

El fichero `/etc/apt/sources.list` puede contener varios tipos de líneas. APT sabe perfectamente cómo interpretar si son `http`, `ftp`, `ssh`, `file` (archivos locales).



Si realizamos algún cambio en este fichero, o queremos actualizar la lista de paquetes, siempre deberemos ejecutar, como root, el comando:

```
root@guadalinux:~# apt-get update
```

### 4.1.3. Agregar un CD-ROM al archivo `sources.list`

Puede ser que los paquetes que deseamos instalar los tengamos en un cd-rom y no necesitemos conectarnos a internet para descargarlos. Lo más sencillo es incorporar el cd-rom al archivo `sources.list`, para lo que ejecutaremos:

```
apt-cdrom [opciones] add
```

Nos solicitará un nombre para ese cd-rom en concreto, por lo que debemos indicárselo.

Las opciones que debemos tener en cuenta son:

**-d directory** punto de montaje del cdrom

**-r** renombrar a un cdrom ya identificado

**-m** no montarlo

**-f** modo rápido sin comprobar los paquetes que hay en él

**-a** modo de escaneo

↪ Veamos un ejemplo:

```
apt-cdrom add
Using CD-ROM mount point /cdrom/
Unmounting CD-ROM
Please insert a Disc in the drive and press enter
Mounting CD-ROM
Identifying.. [c1d7355687b57a98c0ab41f53e9602c2-2]
Scanning Disc for index files..
Found 2 package indexes and 0 source indexes.
Found label 'Debian GNU/Linux testing _Sarge_ - Official Snapshot i386 Binary-1 (20050320)'
This Disc is called:
'Debian GNU/Linux testing _Sarge_ - Official Snapshot i386 Binary-1 (20050320)
```

<sup>9</sup>Cuando decimos que Linux es “Código abierto” (en inglés, *Open Source*) nos referimos precisamente a esta característica. Cualquier persona es libre de ver, modificar y optimizar su código. De esta manera sabemos cómo funciona realmente una aplicación, sus virtudes y también sus defectos y fallos. Al tener acceso al código fuente podemos corregirlo en favor de toda la comunidad y perfeccionarlo. El código fuente es lo que Microsoft, tan celosamente, guarda y no libera. Solamente ellos pueden modificarlo.

El método de desarrollo de la comunidad Linux es mucho mejor que el de Microsoft. Cuando se detecta un fallo, sobre todo en lo referente a la seguridad, inmediatamente es corregido y puesto en conocimiento. No en vano hablamos de programadores y desarrolladores de todo el planeta. Mientras tanto, cuando se detecta un fallo en una aplicación de Microsoft, ésta se corrige o bien en la siguiente versión (previo pago, por supuesto) o bien mediante algún parche, que en algunos casos graves de seguridad se ha resuelto hasta 6 meses después de haberse conocido dicho “agujero”.



```

'Reading Package Indexes...
Hecho Wrote 181 records.
Writing new source list Source List entries for this Disc are:
deb cdrom:[Debian GNU/Linux testing _Sarge_ - Official Snapshot i386 Binary-1 (20050320)]/
unstable contrib main
Repeat this process for the rest of the CDs in your set.

```

La última línea nos indica que este proceso deberemos repetirlo para el resto de cd-rom's que queramos añadir.

Veamos cómo queda el fichero `/etc/sources.list`

```

#_Junta_de_íAndaluca_(Repositorio_raiz)
#_éMtodo_HTTP

deb_cdrom:[Debian_GNU/Linux_testing_-_Sarge_-_Official_Snapshot_i386_Binary
-1_(20050320)]/_unstable_contrib_main
5 deb_http://http.guadalinex.org/debian_sarge_main_contrib_non-free
deb_http://http.guadalinex.org/debian-non-US_sarge/non-US_main_contrib_non-
free
deb_http://http.guadalinex.org/debian-security_sarge/updates_main_contrib_
non-free
deb_http://http.guadalinex.org/repositorio_muflon_guada

10 #_Fuentes
#_deb-src_http://http.guadalinex.org/debian_sarge_main_contrib_non-free
#_deb-src_http://http.guadalinex.org/repositorio_muflon_guada

#_éMtodo_FTP
15 #deb_ftp://ftp.guadalinex.org/repositorio_muflon_main_contrib_non-free_guada
#deb_ftp://ftp.guadalinex.org/repositorio_muflon/non-US_main_contrib_non-
free

#_Mirror_Oficial_de_Guadalinex:_Centro_áInformtico_íCientfico_de_íAndaluca_(
CICA)
#deb_ftp://ftp.cica.es/debian_sarge_main_contrib_non-free
20 #deb_ftp://ftp.cica.es/guadalinex/repositorio_muflon_guada

#_Mirror_Oficial_de_Debian
#_Sarge
#deb_http://ftp.fi.debian.org/debian_sarge_main_contrib_non-free
25 #deb_http://ftp.fi.debian.org/debian-security_sarge/updates_main_contrib_non-
-free
#deb_http://non-us.debian.org/debian-non-US_sarge/non-US_main_contrib_non-
free

```

Listado 4.2: `/etc/apt/sources.list`



Debemos tener en cuenta que esto sólo funcionará si el cd-rom está debidamente configurado en el archivo `/etc/fstab` del sistema.

Si no fuese ésta la ruta de montaje del dispositivo, deberíamos configurarlo a mano, como por ejemplo:

```
apt-cdrom -d /home/usuario/midisco add
```



También tenemos la posibilidad de identificar un cd-rom sin agregarlo al archivo `sources.list` mediante la orden:

```
apt-cdrom ident
```

#### 4.1.4. Instalar paquetes

El sistema de paquetes utiliza una base de datos para monitorizar los paquetes instalados, los no instalados y los que están disponibles por si deseamos instalarlos.

A la hora de instalar paquetes utilizaremos el programa `apt-get` el cual utilizará la base de datos anteriormente comentada y averiguará cómo instalar los paquetes que le indicamos, así como aquellos otros paquetes adicionales, si son necesarios, que serán requeridos por el paquete que deseamos instalar para que funcione correctamente.

Una vez conectados a Internet, para actualizar la lista, se utiliza el comando:

```
apt-get update
```

Este comando busca el paquete solicitado en los archivos listados en `/etc/sources.list`

Si tenemos conexión a internet buscará las fuentes en los repositorios indicados en `/etc/sources.list`



No está de más realizar esta operación con cierta frecuencia para mantenerse informado sobre las posibilidades de actualización del sistema, sobre todo en lo referente a las actualizaciones de seguridad.

Una vez que tenemos el archivo `sources.list` preparado y la base de datos actualizada llega el momento esperado ¡vamos a instalar paquetes!

Nada mejor para explicarlo que con un ejemplo. Vamos a instalar un capturador de pantallas muy sencillo y potente llamado “*ksnapshot*”, así que vamos a indicar los pasos a seguir:

1. Actualizamos la base de datos ejecutando desde una `xterm` y como `root`. Nos debe aparecer algo similar a:

```
apt-get update
```

```
Obj http://http.guadalinex.org sarge/main Packages
Obj http://http.guadalinex.org sarge/main Release
Obj http://http.guadalinex.org sarge/contrib Packages
Obj http://http.guadalinex.org sarge/contrib Release
Obj http://http.guadalinex.org sarge/non-free Packages
Obj http://http.guadalinex.org sarge/non-free Release
Obj http://http.guadalinex.org sarge/non-US/main Packages
Obj http://http.guadalinex.org sarge/non-US/main Release
Obj http://http.guadalinex.org sarge/non-US/contrib Packages
Obj http://http.guadalinex.org sarge/non-US/contrib Release
Obj http://http.guadalinex.org sarge/non-US/non-free Packages
Obj http://http.guadalinex.org sarge/non-US/non-free Release
Obj http://http.guadalinex.org sarge/updates/main Packages
Obj http://http.guadalinex.org sarge/updates/main Release
Obj http://http.guadalinex.org sarge/updates/contrib Packages
Obj http://http.guadalinex.org sarge/updates/contrib Release
Obj http://http.guadalinex.org sarge/updates/non-free Packages
Obj http://http.guadalinex.org sarge/updates/non-free Release
Obj http://http.guadalinex.org muflon/guada Packages
Obj http://http.guadalinex.org muflon/guada Release Leyendo lista de paquetes... Hecho
```

1. Procedemos a instalar ejecutando:

```
apt-get install ksnapshot
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Paquetes sugeridos:
khelpcenter
Se instalarán los siguientes paquetes NUEVOS:
ksnapshot
0 actualizados, 1 se instalarán, 0 para eliminar y 125 no actualizados.
Necesito descargar 87,3kB de archivos.
Se utilizarán 291kB de espacio de disco adicional después de desempaquetar.
Des:1 http://http.guadalinex.org sarge/main ksnapshot 4:3.2.3-1.1 [87,3kB]
Descargados 87,3kB en 0s (257kB/s)
Seleccionando el paquete ksnapshot previamente no seleccionado.
(Leyendo la base de datos ...
98589 ficheros y directorios instalados actualmente.)
Desempaquetando ksnapshot (de ../ksnapshot_4%3a3.2.3-1.1_i386.deb) ...
Configurando ksnapshot (3.2.3-1.1) ...
```

2. Si todo ha funcionado correctamente, vamos a comprobarlo ejecutando la aplicación<sup>10</sup>:

```
$ ksnapshot
```



Vamos a detallar lo que ha hecho `apt-get`:

1. Ha comprobado que tenía el paquete **ksnapshot** en la base de datos.
2. A continuación ha detectado que no necesitaba paquetes extra (nos los ha detallado incluso, sólo uno sugerido).
3. Nos indica el tamaño de los archivos que necesita descargar y el espacio en disco duro que necesitará después de desempaquetarlos (por si acaso no tenemos espacio suficiente en nuestro disco duro, aunque en este caso se trata sólo de 291 kb, poca cosa)

<sup>10</sup>Ya no es necesario actuar como root (#) ahora lo podemos hacer como un usuario normal del sistema (\$)

4. En este caso no nos solicita conformidad para proceder a la descarga e instalación ya que el paquete que vamos a instalar no depende de otros. En caso contrario, si necesitase actualizar otros paquetes que tenemos instalados, además de instalar algunos nuevos, sí que nos pediría conformidad.
5. Finalmente tras haberle dado la conformidad, se conecta a los “*sites*” indicados en el `sources.list`, descarga el paquete, lo desempaqueta y configura. Ya sólo nos queda, en este caso, ejecutar la aplicación.

Los paquetes descargados son almacenados en el directorio `/var/cache/apt/archives` por si los necesitamos en algún otro momento o sencillamente deseamos instalarlo en otro ordenador; ya no tenemos por qué descargarlos de nuevo.

Puede ocurrir que se haya dañado el paquete instalado o sencillamente deseamos reinstalar una nueva versión disponible del mismo, entonces deberemos añadir la opción `--reinstall`

```
apt-get --reinstall install ksnapshot
```

#### 4.1.5. Eliminando paquetes

Si ya no necesitamos utilizar un determinado paquete, bien sea porque no lo utilizamos o porque no nos gusta, podemos eliminarlo del sistema utilizando `apt`. Para realizar esta tarea escribiremos, recuerda, en una `xterm` y como `root`:

```
#apt-get remove nombre_paquete
```

Vamos a ponerlo en práctica con el ejemplo del visor `xpdf`<sup>11</sup>. Si ejecutamos:

```
apt-get remove xpdf
```

Este sería el proceso normal para desinstalar un paquete, pero si probamos a ejecutar `$ xpdf` nos damos cuenta que la aplicación sigue estando instalada. ¿Qué ha sucedido? Sencillamente que no hemos desinstalado el paquete que contiene el binario que ejecuta la aplicación. Vamos a localizar el binario y eliminar la aplicación completamente.

Si escribimos

```
which xpdf
```

nos dirá exactamente dónde se encuentra el binario:

```
/usr/bin/xpdf
```

Ahora, para averiguar el nombre del paquete que realmente debemos indicar en `#apt-get remove nombre_paquete`, ejecutamos

```
dpkg -S /usr/bin/xpdf
xpdf-reader: /usr/bin/xpdf
```

Por tanto lo que debemos hacer para eliminar el `xpdf` de nuestro ordenador sería:

```
apt-get remove xpdf-reader
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Los siguientes paquetes se ELIMINARÁN:
xpdf xpdf-reader
```

---

<sup>11</sup>Visor de documentos PDF



```
0 actualizados, 0 se instalarán, 2 para eliminar y 125 no actualiza-
dos.
Necesito descargar 0B de archivos.
Se liberarán 1711kB después de desempaquetar.
¿Desea continuar? [S/n] s
(Leyendo la base de datos ...
98605 ficheros y directorios instalados actualmente.)
Desinstalando xpdf ...
Desinstalando xpdf-reader ...
```

Si ahora intentamos ejecutar:

```
$ xpdf
```

Veremos que ya no funciona, lo hemos eliminado completamente; aunque ojo, si lo hemos instalado con el comando apt desde internet no hemos borrado los paquetes de instalación. Estos siguen estando en `/var/cache/apt/archives` por si deseamos volver a instalar de nuevo.

↔ Podemos comprobarlo<sup>12</sup> con:

```
apt-get remove ksnapshot
$ ls /var/cache/apt/archives/ks*
ksnapshot-3.2.3-1.1_i386.deb
```

Así que si ahora ejecutamos

```
#apt-get install ksnapshot
```

se instalará el paquete de nuestro directorio local y no es necesario volver a bajarlo desde internet.

Ejecutando `apt-get` como en el ejemplo eliminaremos los paquetes, pero sus archivos de configuración, si es que existían, permanecerán intactos en el sistema. Para una eliminación completa del paquete deberíamos ejecutar:

```
apt-get --purge remove xpdf-reader
```

Para eliminar paquetes también lo podemos hacer añadiendo un “-” a continuación del nombre del paquete a eliminar `apt-get install nombre_paquete-`

Al añadir “-” le estamos indicando que deseamos eliminarla

```
apt-get install xpdf-reader-
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Los siguientes paquetes se ELIMINARÁN:
 xpdf xpdf-reader
0 actualizados, 0 se instalarán, 2 para eliminar y 125 no actualiza-
dos.
Necesito descargar 0B de archivos.
Se liberarán 1711kB después de desempaquetar.
¿Desea continuar? [S/n] s
(Leyendo la base de datos ...
98605 ficheros y directorios instalados actualmente.)
Desinstalando xpdf ...
Desinstalando xpdf-reader ...
```

Pero mejor si dejamos todo como estaba y, para finalizar, lo instalamos de nuevo con:

```
#apt-get install xpdf-reader
```

<sup>12</sup>Como los paquetes `xpdf` y `xpdf-reader` se instalan por defecto no estarán en este directorio. Al actualizarlos sí que estarán en él.

### 4.1.6. Actualizando paquetes

Imaginemos que el paquete `xpdf`, que ya tenemos instalado en nuestro ordenador, está dañado (no funciona) o simplemente deseamos instalar una nueva versión de este paquete; utilizaremos la opción `--reinstall`

```
apt-get --reinstall install xpdf
```

### 4.1.7. Actualizando a una nueva versión

Para actualizar todos los paquetes que tenemos instalados en nuestro sistema tenemos dos opciones:

```
apt-get upgrade
```

```
apt-get dist-upgrade
```

La diferencia entre ambos es que con `upgrade` se actualizará el sistema pero no se instalará un paquete nuevo, ni se eliminará uno ya instalado ni se actualizará un paquete que presente conflictos con otro ya instalado. Sin embargo, si usamos `dist-upgrade` realizamos una actualización completa, es decir, una vez determinado el mejor conjunto de paquetes para actualizar el sistema lo máximo posible, se instalan, actualizan y eliminan todos los que sean necesarios.

Resumiendo, si deseamos tener nuestro ordenador a la “última” deberemos optar por la segunda opción (`dist-upgrade`) ya que nos lo va actualizar todo y nos va a instalar lo último que haya disponible en este momento. Si deseamos actualizar el sistema de forma más “conservadora” optaremos por la 1ª.

Por ejemplo:

```
apt-get dist-upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Calculando la actualización... Listo

...

Necesito descargar 166MB de archivos.
Se utilizarán 40,2MB de espacio de disco adicional después de desempaquetar. ¿Desea continuar? [S/n]
```

Como podemos ver de la salida de ejemplo del comando, con esta opción realizamos una actualización completa ya que se consultan todas las posibles dependencias y conflictos que puedan surgir resolviéndolas de la manera más adecuada.

De todos modos, este tipo de actualización precisa de una buena conexión a internet ya que, siguiendo el ejemplo anterior, necesitamos descargar 166 MB de archivos ¡OJO!



Imaginemos que estamos actualizando nuestro sistema con un cd-rom. APT siempre buscará la versión más reciente de los paquetes y, por tanto, si en nuestro archivo `/etc/apt/sources.list` se encontrase con alguna otra fuente que tuviese una versión del paquete más reciente que la del cd-rom, se descargaría esta nueva versión.

### 4.1.8. Eliminando archivos de paquete no utilizados

Los paquetes que se instalan en nuestro sistema se bajan previamente a un repositorio<sup>13</sup> de paquetes desde el que son instalados automáticamente por APT, sin que nosotros debamos hacer nada especial para que esto ocurra. Sin embargo, con el paso del tiempo, este proceso hace que el repositorio empiece a crecer y vaya ocupando mucho espacio en nuestro disco duro.

Para borrar los paquetes después de haber actualizado por completo nuestro sistema podemos ejecutar:

```
apt-get clean
```

De esta forma se elimina la totalidad de paquetes de la caché.

Pero también podemos optar por:

```
apt-get autoclean
```

De este modo, sólo se eliminarían los paquetes “inútiles”, es decir, los que ya no sirven porque existe una nueva versión de los mismos.



Ojo, tanto la opción `clean` como `autoclean` no dan opción a elegir `[s/n]`, directamente se ejecutan y eliminan.

### 4.1.9. Instalando un nuevo entorno de escritorio: KDE

Ya habíamos visto en la primera entrega la existencia de diferentes entornos de escritorio (Gnome, KDE, Xfce4) y gestores de ventanas (AfterStep, Enlightenment, WindowMaker, IceWM, Sawfish, Blackbox, Fluxbox, Metacity, etc).

Nuestro Guadalinex incorpora sólo Gnome como entorno de escritorio y Fluxbox como gestor de ventanas pero vamos a instalar el “popular”<sup>14</sup> KDE.

#### ↔ Instalando KDE

Si disponemos de una buena conexión ADSL podemos instalar KDE , una vez realizado

```
apt-get update
```

con la orden:

```
apt-get install kde
```

El “problema” reside en que necesitamos descargar 125 MB de archivos, se instalarán 248 paquetes nuevos, etc... ya que también se descargan bastantes aplicaciones asociadas pero que no son necesarias para poder disfrutar de KDE.

Vamos a dejarlo más “light” y sólo nos vamos a bajar lo preciso. Los paquetes recomendados son:

- `kdebase`: La base del sistema de KDE (trae navegador, escritorio, barra de tareas, gestor de impresión...)
- `kmail`: Para leer el correo
- `kopete`: Para mensajería instantánea
- `kaffeine`: Para visualizar videos y reproducir audio.
- `juk`: Para escuchar música en el caso de que no nos agrade `kaffeine`

<sup>13</sup>Recuerda que los paquetes descargados son almacenados en el directorio `/var/cache/apt/archives`

<sup>14</sup>Es tan conocido como Gnome, posee gran potencia y versatilidad.

- kdm: Por si queremos apagar directamente el ordenador desde el escritorio

Una vez que conocemos lo que necesitamos, vamos a proceder a su instalación, ejecutando:

```
apt-get install kdebase kmail kopete kaffeine juk kdm
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Se instalarán los siguientes paquetes extras: enscript gv kappfinder kate kcontrol kdebase-data kdebase-
kio-plugins kdeprint kdesktop kfind khelpcenter kicker klipper kmenuedit konqueror konqueror-nsplugins
konsole kpager kpersonalizer ksmsserver ksplash ksysguard ksysguardd ktip ktnef kwin lesstif1 libgadu3
libkcal2 libkdegst0.6 libkdenetwork2 libkdepim1 libkonq4 libksieve0 libmimelib1 libmusicbrainz2 lib-
musicbrainz4 libsensors3 libtag1 libtunepimp-bin libtunepimp2 poster psutils xaw3dg Paquetes sugeri-
dos: kate-plugins konq-speaker efax hylafax-client mgetty-fax kicker-applets konq-plugins lm-sensors
lm-sensors-mod-2.7 kernel-image-2.6 libtunepimp2-dev Paquetes recomendados kregexpeditor xfonts-konsole
Se instalarán los siguientes paquetes NUEVOS: enscript gv juk kaffeine kappfinder kate kcontrol kde-
base kdebase-data kdebase-kio-plugins kdeprint kdesktop kdm kfind khelpcenter kicker klipper kmail kme-
nuedit konqueror konqueror-nsplugins konsole kopete kpager kpersonalizer ksmsserver ksplash ksysguard
ksysguardd ktip ktnef kwin lesstif1 libgadu3 libkcal2 libkdegst0.6 libkdenetwork2 libkdepim1 libkonq4
libksieve0 libmimelib1 libmusicbrainz2 libmusicbrainz4 libsensors3 libtag1 libtunepimp-bin libtunepimp2
poster psutils xaw3dg
0 actualizados, 50 se instalarán, 0 para eliminar y 125 no actualizados. Necesito descargar 31,5MB
de archivos. Se utilizarán 79,6MB de espacio de disco adicional después de desempaquetar. ¿Desea con-
tinuar? [S/n]
```

Y sólo necesitamos descargar 31,5 MB de archivos en comparación con los 125 MB anteriormente comentados.

Antes de finalizar la instalación se iniciará un asistente para indicar qué gestor de sesiones preferimos (gdm o kdm)<sup>15</sup>. Elegimos kdm<sup>16</sup>.



Una vez finalizada la instalación y desde la misma terminal trabajando como **root** ejecutamos:

```
/etc/init.d/gdm stop
```

Lo que hemos hecho es parar el gestor de sesiones de Gnome.

Ahora debemos abrir una sesión como **root** utilizando la combinación de teclas [**CTRL**]+[**ALT**]+[**F2**] (puede ser cualquier tecla de función desde F1 a F6). Nos identificamos como usuario **root**, ingresamos su contraseña y ejecutamos:

```
/etc/init.d/kdm start
```

Lo que hemos hecho es iniciar con el gestor de sesiones de KDE.

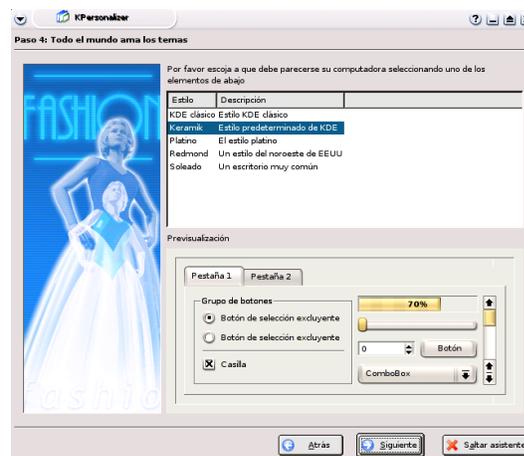
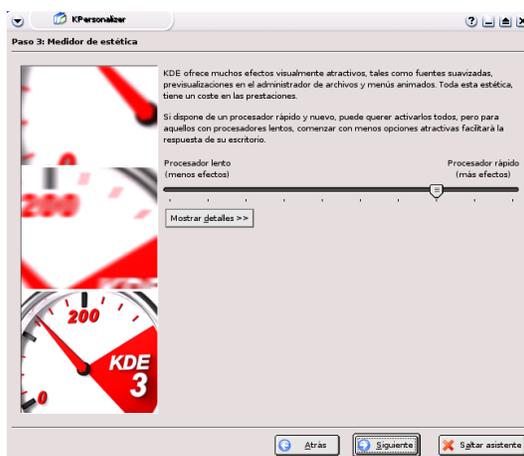
Nos aparecerá la siguiente pantalla en la que, además de identificarnos con nuestro nombre de usuario y contraseña, vamos a elegir en [**Menu**] → [**Sesion Type**] → [**KDE**]

<sup>15</sup>Del inglés gdm - *Gnome Display Manager* / kdm - *KDE Display Manager*

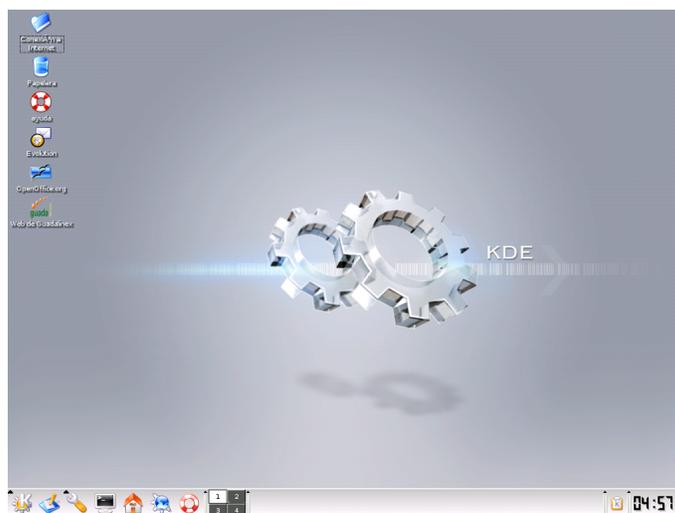
<sup>16</sup>Si preferimos el gestor de sesiones de Gnome siempre podremos volver a él.



A continuación nos aparecerán diferentes pantallas de configuración del entorno. Elegiremos las más adecuadas a nuestro gusto y equipo.



Y tras dar conformidad al asistente de configuración de KDE ya podemos acceder con este resultado:



Si aún, visto lo visto, os parece poco atractivo y queréis “adornarlo” más, lo único que tenemos que hacer es descargar el paquete `kdeartwork` utilizando:

```
apt-get install kdeartwork
```

Este paquete contiene temas de escritorio, salvapantallas, sonidos, fondos de escritorio, decoraciones de ventanas y estilos de controles (widgets) adicionales para KDE.

## 4.2. dpkg - Introducción

El programa `dpkg` es la base del sistema de gestión de paquetes de Debian. Fue creado por Ian Jackson en 1993; es similar a `rpm`<sup>17</sup>. Se utiliza para instalar y desinstalar paquetes “.deb” pero también resulta muy útil para obtener información sobre los paquetes instalados.

Utilizaremos `dpkg`, sobre todo, cuando trabajemos con archivos locales<sup>18</sup>. No es imprescindible ya que `apt` supe la gran mayoría de las tareas con gran facilidad, como hemos visto en el capítulo anterior. Indicaremos brevemente su uso más habitual, aunque si deseamos conocer más podemos ver las páginas man del programa<sup>19</sup>. Trabajaremos desde la carpeta donde tengamos los paquetes .deb previamente descargados, o copiados, o el cd-rom donde se encuentren los mismos.

### 4.2.1. Instalar paquetes

Para instalar paquetes sólo debemos ejecutar<sup>20</sup>:

```
dpkg -i paquete.deb
```

Siendo el “paquete.deb” el que previamente tenemos en nuestro ordenador (HD, cd-rom, USB disk...)

Vamos a ponerlo en práctica instalando el visor Adobe Acrobat Reader, que va incluido en el CDROM de instalación de Guadalinex pero no está instalado.

1. introducimos el CDROM y Guadalinex directamente lo monta

<sup>17</sup>*RPM Package Manager* (o RPM, originalmente llamado *Red Hat Package Manager*) es un sistema de administración de paquetes pensado para Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas. RPM es el formato de paquete de partida del Linux Standard Base.

Originalmente fue desarrollado por Red Hat para Red Hat Linux.

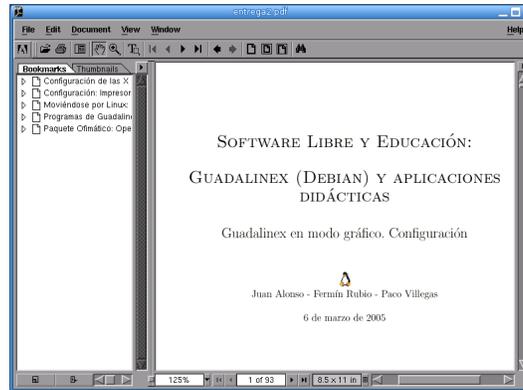
<sup>18</sup>Por ejemplo si no tenemos acceso a internet en la máquina en la que estamos trabajando.

<sup>19</sup>Recuerda: \$ `man dpkg`

<sup>20</sup>“-i” del inglés *install* (instalar)

2. abrimos una terminal y trabajamos como **root**

```
cd /cdrom/extras/paquetes
dpkg -i acroread_4.05-3_i386.deb
```
3. Una vez instalado deberemos aceptar los términos de uso de este software.
4. Ya lo podemos utilizar ejecutando desde un terminal: **\$ acroread**



### 4.2.2. Desinstalar paquetes

Para proceder a la desinstalación de paquetes ejecutamos<sup>21</sup>:

```
dpkg -r paquete.deb

dpkg -r acroread
(Leyendo la base de datos ...
103916 ficheros y directorios instalados actualmente.)
Desinstalando acroread ...
```

Si intentamos **\$ acroread** veremos que ya no funciona.

Si deseamos que, además de desinstalar, borre también los ficheros de configuración

```
dpkg -purge paquete.deb
```

### 4.2.3. Opciones útiles

Podemos utilizar a **dpkg** para:

**dpkg l** lista los paquetes instalados en nuestro sistema.

**dpkg -l nombre** lista la información que tiene de ese paquete.

```
dpkg -l acroread
Desired=Unknown/Install/Remove/Purge/Hold
| Estado=No/Instalado/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-
problems (Status,Err: mayúsc.=malo)
```

<sup>21</sup>“-r” del inglés: *remove* (suprimir, eliminar)



```
||/ Nombre Versión Descripción
+++-----
=====
rc acroread 4.05-
3 Adobe Acrobat Reader: Portable Document Form
```

**dpkg -L nombre** lista los directorios donde ha instalado archivos el paquete indicado.

```
dpkg -L acroread
/.
/usr
/usr/bin
/usr/lib
/usr/lib/netscape
/usr/lib/netscape/plugins-libc6
/usr/lib/Acrobat4
/usr/lib/Acrobat4/Reader
.
..
/usr/share/doc/acroread/ReadMe
/usr/share/doc/acroread/Acrobat.pdf
/usr/share/doc/acroread/License.pdf
/usr/share/doc/acroread/MapTypes.pdf
```

**dpkg -s nombre** lista información de los paquetes instalados, estado y dependencias.

```
dpkg -s acroread
Package: acroread
Status: install ok installed
Priority: optional
Section: non-free/text
Installed-Size: 13888
Maintainer: Anthony Fok <foka@debian.org>
Architecture: i386
Version: 4.05-3
Provides: pdf-viewer, postscript-preview
Depends: libc6 (>= 2.1.2), xlibs (>= 4.3.0-7)
Recommends: mime-support
Suggests: netscape-base-4 (>= 1:4.76-1) | netscape | mozilla
Description: Adobe Acrobat Reader: Portable Document Format file viewer.
Adobe Acrobat Reader for viewing and printing Adobe Portable Document
Format (PDF) files. It also works well as a plugin for the glibc2
(libc6) version of Netscape 4.7x.
.
Home Page: http://www.adobe.com/products/acrobat/readermain.html
```

**dpkg -S nombre** busca un nombre de fichero, correspondiente al patrón de búsqueda que le indicamos, en los paquetes instalados.

```
dpkg -S acroread
acroread: /usr/lib/menu/acroread
acroread: /usr/share/doc/acroread/MapTypes.pdf
acroread: /usr/lib/ Acrobat4/bin/acroread
acroread: /usr/share/doc/acroread/ReadMe
acroread: /usr/share/doc/acroread/copyright
acroread: /usr/share/doc/acroread/License.pdf
kdebase-data: /usr/share/icons/crystalsvg/64x64/apps/acroread.png
acroread: /usr/share/doc/acroread/INSTGUID.TXT.gz
kdebase-data: /usr/share/icons/crystalsvg/32x32/apps/acroread.png
kdebase-data: /usr/share/icons/crystalsvg/48x48/apps/acroread.png
acroread: /usr/share/doc/acroread/README.Debian
kdebase-data: /usr/share/icons/crystalsvg/16x16/apps/acroread.png
acroread: /usr/bin/acroread
kappfinder: /usr/share/apps/kappfinder/apps/Office/acroread.desktop
acroread: /usr/share/doc/acroread/ Acrobat.pdf
acroread: /usr/lib/mime/packages/acroread
kdebase-data: /usr/share/icons/crystalsvg/128x128/apps/acroread.png
acroread: /usr/share/doc/acroread/changelog.Debian.gz
acroread: /usr/lib/ Acrobat4/Reader/intellinux/bin/acroread
acroread: /usr/share/doc/acroread/LICREAD.TXT.gz
acroread: /usr/share/doc/acroread
```

### 4.3. synaptic

¡Muy bonito! ¿verdad? Dejamos para el final el entorno gráfico ¡eh!

Cierto, pero es preferible saber cómo funciona algo en profundidad para que, más tarde, cuando aparecen las “*ventanitas*” ya todo nos suene y sólo entonces sí que es realmente sencillo.

Aunque cuando finalice el capítulo os daréis cuenta que las herramientas gráficas no tienen la potencia de los comandos.

Empecemos pues, **synaptic** es una herramienta en modo gráfico para la instalación, desinstalación de paquetes y mantenimiento en general de nuestra distribución.

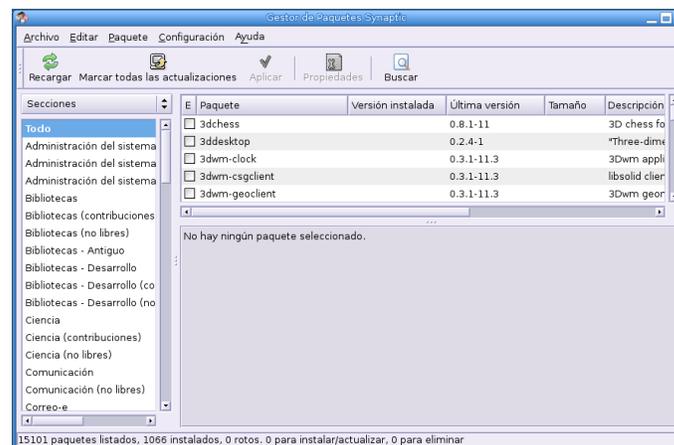
Para trabajar con ella podemos dirigirnos directamente a:



→ **Aplicaciones** → **Configuración** → **Sistema** → **Synaptic (gestor de paquetes)**

O bien desde una **xterm** y como **root** ejecutando:

```
synaptic &
```



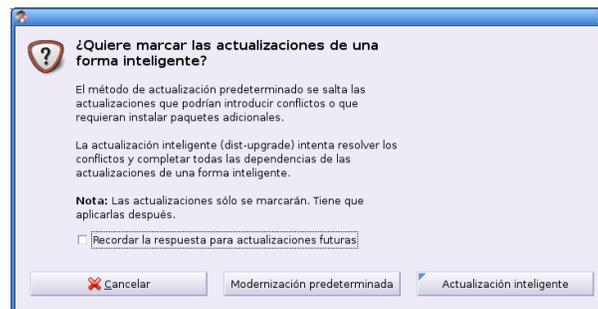
Todo lo que se ha explicado en los apartados anteriores de este capítulo tiene su reflejo en este entorno gráfico. Demos un repaso a las principales equivalencias:



**Recargar:** # apt-get update

**Marcar todas las actualizaciones:**

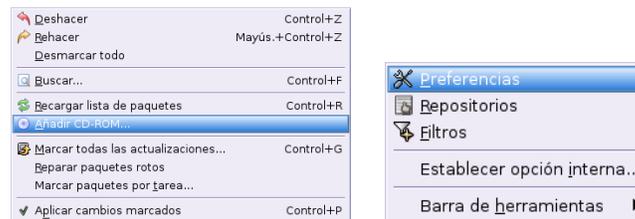
Al marcar todas las actualizaciones podemos elegir en el siguiente menú:



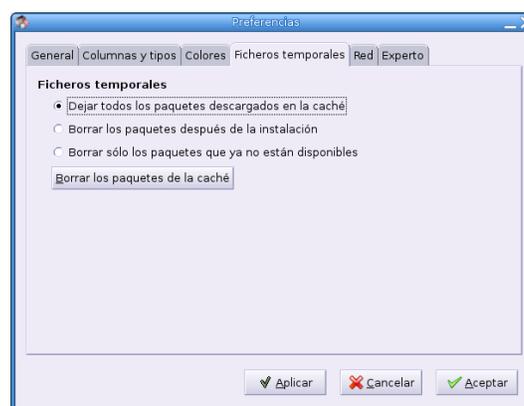
**Actualización inteligente:** # apt-get dist-upgrade

**Modernización predeterminada:** # apt-get upgrade

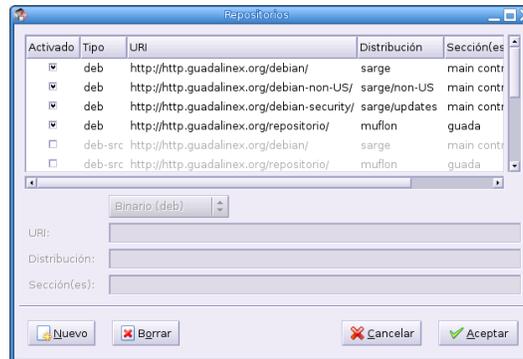
El menú [**A**cciones] no presenta dificultad. Desde [**P**referencias] podemos:



Ajustar las preferencias de **synaptic** tanto en aspectos visuales (colores, organización gráfica ...) como organizativos (proxy, acciones con los paquetes una vez descargados ...)

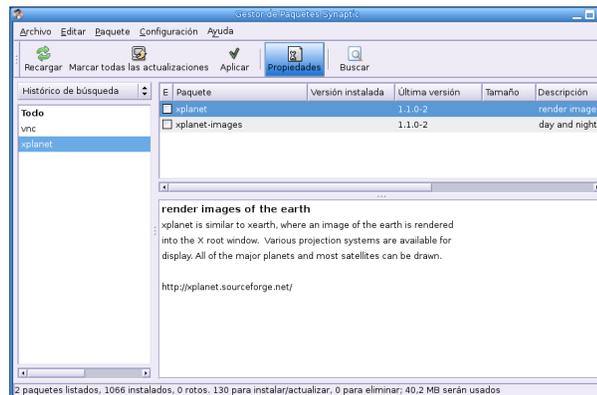


Desde el menú [**Acciones**] podemos configurar los repositorios, es decir, configurar el archivo: `/etc/sources.list`



➔ **Para practicar:** Vamos a ver cómo se instalaría un programa con synaptic:

1. En la casilla **Buscar** le indicaremos el nombre de la aplicación que deseamos instalar. En este caso vamos a elegir un programa de astronomía, **xplanet**.
2. Una vez que lo ha encontrado,



observamos haciendo clic con el botón derecho, en la opción [**Propiedades**], que nos indica en diferentes pestañas que el paquete no está instalado; nos muestra una descripción del mismo; las dependencias y los ficheros que va a instalar y dónde.

3. Para proceder a la instalación del paquete, antes debemos seleccionarlo haciendo clic con el botón derecho y eligiendo [**Marcar para instalación**].

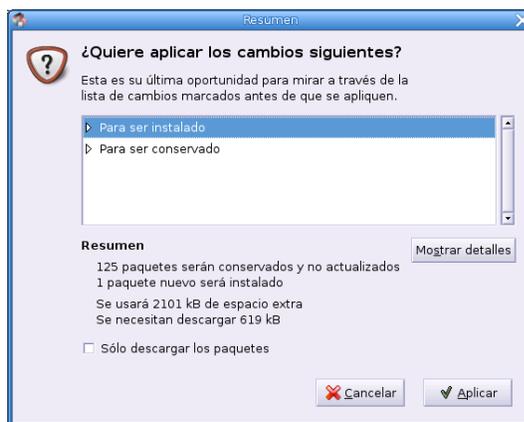


No es necesario marcar un paquete e instalarlo. Se pueden marcar diferentes y más tarde instalarlos todos juntos.

4. Una vez que hayamos decidido que ya no vamos a instalar más paquetes haremos “clic” en :



5. Y nos aparecerá una ventana informándonos de los cambios que va a realizar:



Podemos marcar una casilla de verificación para que sólo descargue el paquete y que no lo instale.

Si se trata de desinstalar paquetes se realizaría el mismo proceso pero, en lugar de señalar [**Marcar para la instalación**], procederíamos a elegir [**Marcar para la eliminación**] ya que el resto de pasos a seguir serían idénticos. ■

## 4.4. Otros

Las herramientas más comunes para el mantenimiento de nuestro GuadaLinux son las ya comentadas con anterioridad: `apt-get`, `dpkg` y `synaptic`.

Pero además de éstas, existen otras como son:

- `dselect`
- `aptitude`

Ambas se ejecutan en modo comando desde una `xterm` y, como es lógico, actuando como `root`.

Para ampliar información sobre estos comandos podéis ver las páginas man de ambos:

```
$ man dselect
```

```
$ man aptitude
```