

El sistema de ficheros ext3

Carlos Mancha

Índice de contenido

1.Introducción.....	1
2.Conceptos básicos de sistemas de ficheros.....	2
3.Sistemas de ficheros con journaling en Linux	3
4.Principales características del sistema de ficheros ext3.....	5
5.Ventajas de utilizar ext3.....	6
6.Conclusiones.....	10
7.Fuentes de información.....	10

1.Introducción.

Los sistemas de ficheros son uno de los principales componentes de un sistema operativo y de ellos se espera que sean rápidos y extremadamente fiables. Sin embargo a veces ocurren fallos imprevistos y las máquinas se caen inesperadamente bien por fallos hardware, por fallos software o por fallos eléctricos.

Después de un apagado incorrecto dejar de nuevo el sistema de ficheros en un estado consistente puede llevar mucho tiempo. Las capacidades de los discos duros crece y este tiempo se va convirtiendo en un serio problema, el sistema se queda "offline" muchos minutos mientras el disco es escaneado, chequeado y reparado. Aunque los discos duros cada vez son más rápidos, el crecimiento de su velocidad es muy pequeño en comparación con el enorme crecimiento de su capacidad (desafortunadamente el doble de capacidad de un disco supone emplear el doble de tiempo en su recuperación utilizando las técnicas tradicionales de chequeo).

Cuando la disponibilidad del sistema es muy importante este tiempo no se puede desperdiciar, así que es necesario un mecanismo para evitar realizar un chequeo completo del disco cada vez que se apague incorrectamente la máquina. Este nuevo mecanismo debe permitir que el sistema de ficheros sea fiable y tenga compatibilidad con las aplicaciones actuales. Para ello se crearon los sistemas de ficheros con journaling¹ o sistemas de ficheros transaccionales², los cuales permiten que la consistencia de los datos del sistema de ficheros se mantenga después de un apagado incorrecto de la máquina. En este documento se describen las ventajas de tener un sistema de ficheros con journaling ext3.

¹ *Journal o journaling es traducido por algunos autores como diario o bitácora.*

² *En el campo de los sistemas de ficheros una transacción se puede considerar como el conjunto de pasos necesarios para completar una operación sobre un fichero. La transacción garantiza que o todas o ninguna de las modificaciones que se realizan sobre el sistema de ficheros serán llevadas a cabo.*

2. Conceptos básicos de sistemas de ficheros.

Un fichero es una abstracción muy importante en informática. Los ficheros sirven para almacenar datos de forma permanente y ofrecen un pequeño conjunto de primitivas muy potentes (abrir, leer, avanzar puntero, cerrar, etc.). Los ficheros se organizan normalmente en estructuras de árbol, donde los nodos intermedios son directorios capaces de agrupar otros ficheros.

El sistema de ficheros es la forma en que el sistema operativo organiza, gestiona y mantiene la jerarquía de ficheros en los dispositivos de almacenamiento, normalmente discos duros. Cada sistema operativo soporta diferentes sistemas de ficheros. Para mantener la modularización del sistema operativo y proveer a las aplicaciones con una interfaz de programación (API) uniforme, los diferentes sistemas operativos implementan una capa superior de abstracción denominada Sistema de Ficheros Virtual (*VFS: Virtual File System*). Esta capa de software implementa las funcionalidades comunes de los diversos sistemas de ficheros implementados en la capa inferior.

Los sistemas de ficheros soportados por Linux se clasifican en tres categorías:

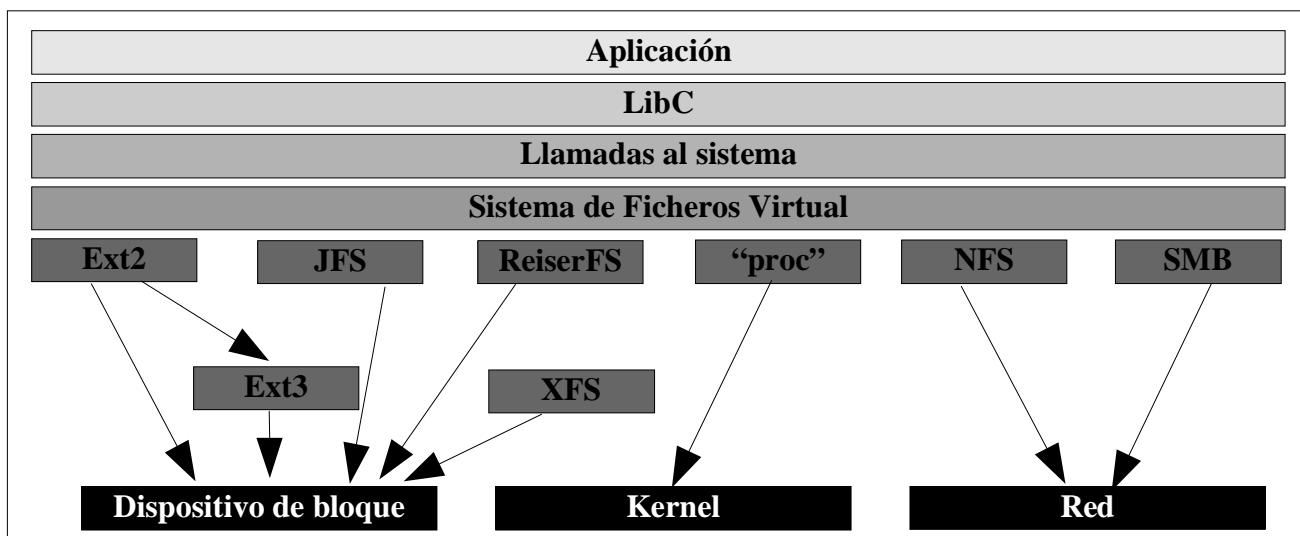
1. **Basados en disco:** discos duros, disquetes, CD-ROM, etc. (Estos sistemas son ext2, ext3, ReiserFS, XFS, JFS, ISO9660, etc.)
2. **Sistemas remotos (de red):** NFS, Coda, Samba, etc.
3. **Sistemas especiales:** procfs, ramfs y devfs.

El modelo general de ficheros puede ser interpretado como orientado a objetos, donde los objetos son construcciones de software (estructura de datos y funciones y métodos asociados) de los siguientes tipos:

- **Super bloque:** mantiene información relacionada a los sistemas de ficheros montados. Está representado por un bloque de control de sistema almacenado en el disco (para sistemas basados en disco).
- **i-nodo:** mantiene información relacionada a un fichero individual. Cada i-nodo contiene la meta-información del fichero: propietario, grupo, fecha y hora de creación, modificación y último acceso, más un conjunto de punteros a los bloques del disco que almacenan los datos del fichero. Almacena toda la información acerca del fichero excepto el fichero en sí.
- **Fichero:** mantiene la información relacionada a la interacción de un fichero abierto y un proceso. Este objeto existe sólo cuando un proceso interactúa con el fichero.
- **Dentry:** enlaza una entrada de directorio (*pathname*) con su fichero correspondiente. Los objetos "dentry" recientemente usados son almacenados en una caché (*dentry cache*) para acelerar la translación desde un nombre de fichero al i-nodo correspondiente.

Desde hace mucho tiempo, el sistema de ficheros estándar en Linux es el ext2. Éste fue diseñado por *Wayne Davidson* con la colaboración de *Stephen Tweedie* y *Theodore Ts'o*. Es una mejora al sistema anterior, ext, diseñado por *Rémy Card*. El ext2 está basado en i-nodos (asignación indexada). Cada i-nodo mantiene la meta-información del fichero y los punteros a los bloques con los datos "reales".

Donde encajan los Sistemas de Ficheros dentro del Sistema Operativo



3. Sistemas de ficheros con journaling en Linux .

Al trabajar con un ordenador, para mejorar el rendimiento de las operaciones de E/S, los datos del disco son temporalmente almacenados en la memoria RAM (Linux utiliza para ello dos mecanismos el *page-cache* y el *buffer-cache*). Los problemas surgen si hay un corte de suministro eléctrico antes que los datos modificados en la memoria (*dirty buffers*) sean grabados nuevamente al disco. Se generaría una inconsistencia en el estado global del sistema de ficheros. Por ejemplo, un nuevo fichero que todavía no fue "creado" en el disco u otros que hayan sido borrados pero sus i-nodos y bloques de datos todavía permanecen como "activos" en el disco.

El "fsck" (*file system check*) fue la herramienta que resolvía dichas inconsistencias, pero el "fsck" tiene que analizar la partición completa y verificar las interdependencias entre i-nodos, bloques de datos y contenidos de directorios. Con la ampliación de la capacidad de los discos, la recuperación de la consistencia del sistema de fichero se ha convertido en una tarea que requiere mucho tiempo, por lo que crea problemas serios de disponibilidad de las máquinas afectadas. Esta es la razón principal de que los sistemas de ficheros hayan importado de las bases de datos las técnicas de transacciones y recuperación, y así hayan aparecido los sistemas de ficheros con journaling.

Un sistema con journaling es un sistema de ficheros tolerante a fallos en el cual la integridad de los datos está asegurada porque las modificaciones de la meta-información de los ficheros son primero grabadas en un registro cronológico (log o journal, que simplemente es una lista de transacciones) antes que los bloques originales sean modificados. En el caso de un fallo del sistema,

un sistema con journaling asegura que la consistencia del sistema de ficheros es recuperada. El método más común es el de grabar previamente cualquier modificación de la meta-información en un área especial del disco, el sistema realmente grabará los datos una vez que la actualización de los registros haya sido completada.

A la hora de recuperar la consistencia después de un fallo, el módulo de recuperación analizará el registro y sólo repetirá las operaciones incompletas en aquellos ficheros inconsistentes, es decir que la operación registrada no se haya llevado a cabo finalmente, con lo que se recuperará la consistencia del sistema de ficheros casi al instante, ya que en vez de examinar todos los meta-datos (como hace el "fsck"), sólo se inspeccionan aquellas porciones de los meta-datos que han sido cambiadas recientemente.

La demanda de sistemas de ficheros que soporten terabytes de datos, miles de ficheros por directorios y compatibilidad con arquitecturas de 64 bits ha hecho que en los últimos años haya crecido el interés de la disponibilidad de sistemas con journaling en Linux, ya que utilizando estos sistema de ficheros se simplifican los reinicios de la máquina, se reduce la fragmentación y se aceleran las operaciones de entrada/salida.

Los primeros sistemas de ficheros con journaling fueron creados a mediados de los ochenta e incluyen a Veritas (VxFS), Tolerant y JFS de *IBM*. Linux tiene ahora disponibles cuatro sistemas de ficheros transaccionales: ReiserFS de *Namesys*, XFS de *Silicon Graphics (SGI)*, JFS de *IBM* y el ext3 que fue desarrollado por *Stephen Tweedie*, co-desarrollador del ext2. Cada uno de ellos tiene unas características específicas que le diferencian del resto, alguno se comportan mejor que otros en algunos casos (pero no en todos), por ejemplo ReiserFS es bueno leyendo ficheros pequeños o medianos, XFS tiene mejor rendimiento para ficheros grandes y con JFS se facilita mucho la migración de sistemas con OS/2 Warp y AIX a Linux.

Comparativa de los Sistemas de Ficheros con journaling en Linux

<i>Soporte del kernel</i>	<i>Ext3</i>	<i>ReiserFS</i>	<i>XFS</i>	<i>JFS</i>
<i>Requisitos del kernel</i>	No	No	Si	No
<i>Árbol de fuentes kernel 2.4.x</i>	2.4.15	2.4.1	No	No
<i>Árbol de fuentes kernel 2.5.x</i>	2.5.0	2.5.0	No	No
<i>Licencia</i>	GPL	GPL	GPL	GPL
<i>Características</i>	-	-	-	-
<i>Máximo tamaño de bloque</i>	4 Kb	4 Kb	4 Kb	4 Kb
<i>Máx. tamaño sistema de ficheros</i>	16.384 Gb	17.592 Gb	18.000 Pb ³	32 Pb
<i>Máximo tamaño de fichero</i>	2.048 Gb	1 Eb ⁴	9.000 Pb	4 Pb
<i>Login de datos</i>	Si	No	No	No
<i>Uso con NFS</i>	Si	No	Si	Si
<i>Inclusión en distribuciones</i>	-	-	-	-
<i>Red Hat 7.2</i>	Si	Si	No	Si
<i>Suse 8.0</i>	Si	Si	Si	Si
<i>Mandrake 8.2</i>	Si	Si	Si	Si
<i>Slackware 8.1</i>	Si	Si	Si	Si

4.Principales características del sistema de ficheros ext3.

El sistema de ficheros ext3 es una extensión con journaling del sistema de ficheros ext2. Como ya hemos visto con el journaling se obtiene una enorme reducción en el tiempo necesario para recuperar un sistema de ficheros después de una caída, y es por tanto muy recomendable en entornos donde la alta disponibilidad es muy importante, no sólo para reducir el tiempo de recuperación de máquinas independientes sino también para permitir que un sistema de ficheros de una máquina caída sea recuperado en otra máquina cuando tenemos un cluster con algún disco compartido. Además se posibilita que el sistema de ficheros caído de una máquina (por ejemplo un servidor) esté disponible cuanto antes para el resto de máquinas a través de la red (nfs, samba, ftp, http, etc.).

El principal objetivo del ext3 es por tanto la disponibilidad, es decir, cuando se apague incorrectamente la máquina tener el sistema totalmente disponible al momento después de volver a arrancar sin necesidad de que se tenga que esperar a pasar un "fsck", el cual tarda mucho tiempo.

³ Petabyte o 10^{15} bytes.

⁴ Exabyte o 10^{18} bytes.

Además con ext3 se ha añadido el journaling de manera que sea totalmente compatible con los sistemas de ficheros ext2 (es posible migrar sistemas de ficheros ext2 existentes a ext3 y viceversa muy fácilmente).

La distribución *Red Hat Linux 7.2* ya incluye ext3 como opción, y es actualmente el sistema de ficheros oficialmente soportado por dicha empresa (en las nuevas distribuciones es opcional formatear las particiones del disco con el antiguo ext2). El resto de distribuciones también lo están incluyendo (incluso se puede usar ext3 en *Debian GNU/Linux 3.0 Woody*), por lo que en poco tiempo se espera que el ext3 sustituya al ext2 como sistema de ficheros estándar en Linux.

Ext3 en realidad es ext2 con un fichero adicional de registro, es decir, es una capa adicional sobre ext2 que mantiene un fichero de registro log de transacciones. Debido a que está integrado en el ext2 puede que no explote todas las posibilidades de los sistemas de journaling puros, pero se está trabajando en esta área para mejorarlo.

5. Ventajas de utilizar ext3.

Podemos decir que hay cuatro razones principales para migrar de un sistema de ficheros ext2 a ext3: disponibilidad, integridad de los datos, velocidad y fácil migración.

→ Disponibilidad:

Después de un apagado incorrecto de la máquina los sistemas de ficheros ext2 no pueden ser montados de nuevo hasta que su consistencia haya sido chequeada por el programa "fsck". El tiempo que tarda el programa "fsck" está determinado por el tamaño del sistema de ficheros, hoy en día muy grandes (decenas de gigabytes), por lo que se tarda mucho tiempo en recuperar el sistema de ficheros. Además cuantos más ficheros tengamos en el sistema de ficheros más se tardará en chequear su consistencia. Chequear sistemas de ficheros de decenas de gigabytes puede llevar varios minutos, esto limita seriamente la disponibilidad.

En contraste el ext3 no requiere un chequeo del disco, incluso después de un apagado incorrecto del sistema. Esto es debido a que los datos son escritos al disco de tal manera que el sistema de ficheros siempre está consistente. Sólo se realizará un "fsck" en el caso de fallos hardware raramente dados (por ejemplo fallos físicos del disco duro), y en el caso de que el sistema de ficheros esté configurado para que se chequee completamente de forma automática cada cierto periodo de tiempo o cada cierto número de montajes para prevenir posibles fallos. Además con ext3 se utiliza (si fuese necesario) exactamente el mismo "fsck" que se utiliza con ext2.

El tiempo necesario para recuperar un sistema de ficheros ext3 después de un apagado incorrecto no depende del tamaño del sistema de ficheros ni del número de archivos que tenga, sólo depende del tamaño del "journal" (espacio usado para almacenar la información transaccional)

utilizado para mantener la consistencia. Con el tamaño que se utiliza por defecto para el "journal" (tamaño fijado automáticamente por la utilidad de creación del sistema de ficheros "mkfs") se tarda alrededor de un segundo en restaurar un sistema de ficheros inconsistente (dependiendo de la velocidad del hardware).

→ Integridad de los datos:

Usando ext3 el sistema de ficheros puede proporcionar garantías más fuertes respecto a la integridad de los datos en el caso de un apagado incorrecto del sistema. Pudiendo escoger el tipo y nivel de protección que se le da a los datos. Se puede escoger mantener la consistencia de los datos pero permitir daños en los datos dentro del sistema de ficheros en el caso de un apagado incorrecto, esto puede dar un pequeño aumento de la velocidad bajo algunas pero no todas las circunstancias. Alternativamente, se puede escoger asegurar que los datos son consistentes con el estado del sistema de ficheros, esto significa que nunca habrá "datos basura" de un fichero recientemente escrito después de una caída del sistema. Esta última opción es la utilizada por defecto.

EL ext3 escribe tres tipos de bloques de datos en el registro:

1. *Meta-información*: Contiene el bloque de meta-información que está siendo actualizado por la transacción. Cada cambio en el sistema de ficheros, por pequeño que sea, es escrito en el registro. Sin embargo es relativamente barato ya que varias operaciones de E/S pueden ser agrupadas en conjuntos más grandes y pueden ser escritas directamente desde el sistema page-cache.
2. *Bloques descriptores*: Estos bloques describen a otros bloques del registro para que luego puedan ser copiados al sistema principal. Los cambios en estos bloques son siempre escritos antes que los de meta-información.
3. *Bloques cabeceras*: Describen la cabecera y cola del registro más un número de secuencia para garantizar el orden de escritura durante la recuperación del sistema de ficheros.

Con ext3 se mantiene la consistencia tanto en la meta-información (i-nodos o metadatos) como en los datos de los ficheros (datos propiamente dichos). A diferencia de los demás sistemas de journaling mencionados anteriormente, la consistencia de los datos también está asegurada.

→ Velocidad:

A pesar de escribir a veces algún dato más de una vez, ext3 es en algunos casos incluso más rápido que el ext2 por que el journaling del ext3 optimiza el movimiento de cabeza del disco duro. Con ext3 se puede escoger entre tres modos de journaling diferentes para optimizar la velocidad, equilibrando esta con una mayor o menor integridad de los datos dependiendo de las necesidades.

Los diferentes modos son:

- *data=writeback*: limita la garantía de integridad de los datos, permitiendo a los antiguos datos aparecer en ficheros después de una caída, para un posible pequeño incremento de la velocidad en algunas circunstancias. Este es el modo journaling por defecto en muchos otros sistemas de ficheros journaling, esencialmente proporciona las garantías más

limitadas de integridad en los datos y simplemente evita el chequeo en el reinicio del sistema.

- *data=ordered* (modo por defecto): garantiza que los datos son consistentes con el sistema de ficheros. Los ficheros escritos recientemente nunca aparecerán con contenidos basura después de una caída.
- *data=journal*: requiere un "journal" grande para una velocidad razonable en la mayoría de los casos y por lo tanto tarda más tiempo recuperar el sistema en el caso de un apagado incorrecto, pero es algunas veces es más rápido para algunas operaciones ya que funciona muy bien si se escriben muchos datos al mismo tiempo (por ejemplo en los spools de correo o servidores NFS sincronizados). No obstante, utilizar el modo "journal" para un uso normal resulta con frecuencia un poco más lento.

El modo por defecto (*ordered*) es el recomendable, pudiendo cambiar el modo en el montaje del sistema de ficheros.

→ Fácil migración:

Las particiones ext3 no tienen una estructura de ficheros diferentes a los de ext2, por lo que no sólo se puede pasar de ext2 a ext3, sino que lo opuesto también funciona, esto es útil sobre todo si en algún caso el registro se corrompe accidentalmente, por ejemplo debido a sectores malos del disco. Es decir, existe total compatibilidad entre ext2 y ext3, se puede convertir un sistema de ficheros ext2 a ext3 y viceversa fácilmente, además de poder montar un sistema de ficheros ext3 como ext2⁵ (ya que la estructura de formateo del disco es la misma).

Es posible por tanto cambiar fácilmente de ext2 a ext3 y beneficiarse de las ventajas de un sistema de ficheros journaling robusto sin necesidad de reformatear el disco. Podemos pasar de un sistema a otro sin necesidad de tener que realizar un tedioso proceso de backup, formateo y restauración de los datos, con la posibilidad de que se produzca algún error (con los otros sistemas de ficheros con journaling es necesario formatear la partición con su propia utilidad de formateo). El programa "tune2fs" puede añadir el journal a un sistema de ficheros ext2 ya existente. Si el sistema de ficheros estaba ya montado cuando se migraba, el journal será visible como un fichero "journal" en el directorio raíz del sistema de ficheros. Si no estaba montado el journal estará oculto y no aparecerá en el sistema de ficheros (así ocurre si se crea durante la instalación del sistema).

El sistema de ficheros ext3 se ha beneficiado de la prolongada historia de mejoras y corrección de errores que tiene el ext2 (del cual a parte su código fuente) y continuará siendo así. Esto significa que ext3 comparte la robustez del ext2, pero también las nuevas características que se han añadido al ext2.

Resumiendo, ext3 es totalmente compatible en ambos sentidos con ext2. Se puede migrar un sistema ex2 a ext3 muy fácilmente, se puede montar un sistema ext3 como ext2 sin modificar nada del sistema de ficheros journal y también eliminar el journal para volver al sistema ext2 anterior.

⁵ Sólo cuando la partición ext3 ha sido anteriormente desmontada correctamente, sino se puede perder la información de journal necesaria para recuperar el sistema de un apagado incorrecto.

Otras ventajas importantes de utilizar ext3 son:

1. El ext3 como el ext2 tiene múltiples desarrolladores y organizaciones involucradas en su desarrollo, por lo que su evolución no depende de una sola persona o empresa.
2. Ext3 proporciona y hace uso de una capa genérica de journaling (Journaling Block Device, JBD) la cual puede ser usada en otros contextos. El ext3 no sólo puede hacer "journal" un sistema de ficheros estándar, también otros dispositivos soportados por Linux (NVRAM, disk-on-chip, USB flash memory drives, etc.) pueden ser utilizados con ext3.
3. Ext3 tiene una amplia compatibilidad con todas las plataformas, trabaja tanto en arquitecturas de 32 como de 64 bits, y tanto en sistemas little-endian como big-endian. Algunos sistemas operativos (por ejemplo algunos clones y variantes de UNIX y BeOS) pueden acceder a ficheros en un sistema de ficheros ext2, estos sistemas también lo pueden hacer en un sistema de ficheros ext3.
4. Ext3 no requiere profundos cambios en el corazón del núcleo y no requiere tampoco nuevas llamadas al sistema. Ext3 está integrado actualmente en los kernels responsabilidad de *Alan Cox* y *Linus Torvalds* lo incluirá muy pronto en su kernel oficial. Seguramente el ext3 será el sistema de ficheros estándar de Linux en un futuro próximo.
5. Ext3 reserva uno de los i-nodos especiales de ext2 para el registro de journal, pero los datos del mismo pueden estar en cualquier conjunto de bloques, y en cualquier sistema de ficheros. Inclusive se puede compartir el registro de journal entre sistemas distintos.
6. El programa de recuperación de sistemas de ficheros "e2fsck" tiene un muy reconocido éxito en la recuperación de datos cuando el software o el hardware falla y corrompe un sistema de ficheros. Ext3 usa el mismo código que el "e2fsck" para salvar el sistema de ficheros después de una posible corrupción, y por consiguiente tiene la misma robustez que el ext2 contra posibles pérdidas catastróficas de datos cuando haya fallos de corrupción en los mismos.

Todas estas peculiaridades del ext3 son totalmente transparentes al usuario el cual trabajará igual que lo hacía con ext2, incluido el montaje y utilización de otros sistemas de ficheros (NFS, dispositivos de almacenamiento externos, etc.).

6. Conclusiones.

Por todas las razones que hemos visto no sólo se puede confiar en el sistema de ficheros ext3, sino que además es recomendable utilizarlo ya que *Red Hat* lo ha incluido como sistema de ficheros estándar de su distribución habiendo realizado todas las pruebas necesarias en múltiples configuraciones diferentes para ver su robustez, con un resultado positivo en todas ellas. Incluso desde hace más de dos años el ext3 esta funcionando en grandes servidores (por ejemplo los servidores de *rpmfind.net*).

Actualmente Linux es conocido como un sistema operativo muy estable, la problemática se genera cuando el hardware no es tan fiables como se desearía o la persona que lo utiliza no lo hace correctamente, ya que en la mayoría de los casos, cuando un sistema falla normalmente es debido a un fallo hardware o a un fallo humano. A veces se producen apagados incorrectos de las máquinas, y por tanto es necesario esperar a que se realice un chequeo y recuperación del disco durante varios (a veces muchos) minutos para poder volver a utilizarla, y además se corre el peligro de una pérdida importante de información.

Ya se han realizado las primeras pruebas de una instalación de *Linux Mobile System* en un USB Flash Memory Drive con sistema de ficheros ext3 con un resultado totalmente satisfactorio. Siendo totalmente transparente el cambio de sistema de ficheros a ext3 para el usuario. Por lo que en próximas versiones de *Linux Mobile System* se incluirá el sistema de ficheros transaccional ext3 como sistema de ficheros por defecto.

7. Fuentes de información.

Este documento esta basado en la información recopilada de las siguientes fuentes:

- *"Journal File System in Linux"*. Ricardo Galli. 2001.
- *"Journaling the Linux ext2 Filesystem"*. Sthephen C. Tweedie. 1998.
- *"Ext3, Journaling Filesystem"*. Stephen C. Tweedie. 2000.
- *"Whitepaper: Red Hat's New Journaling File System: ext3"*. Michael K. Johnson. Red Hat Inc. 2001.
- *"Red Hat Linux/x86 7.2 Release Notes"*. Red Hat Inc. 2001.
- *"Advanced filesystem implementor's guide"*. (IBM DeveloperWorks). Daniel Robbins. 2001.
- *"Journaling File Systems"*. (Linux Magazine). Steve Best. 2002.
- *"Linux Filesystems"*. William von Hagen. 2002.

Carlos Alberto Mancha Compañy <carlosmancha@users.sourceforge.net>

Este documento se distribuye bajo licencia GFDL. Copyright © 2002 Linux Mobile System (LMS) - <http://linuxmobile.sf.net>