

Curso Linux Admin

Manejo y Creación de
Filesystems

Temario

| | |
|--|----|
| Particiones y sistemas de archivos | 3 |
| Información general | 3 |
| Tipos de particiones | 4 |
| Particiones primarias | 4 |
| Particiones extendidas y lógicas | 5 |
| Razones para el uso de particiones | 5 |
| Las ventajas del uso de particiones extendidas | 6 |
| Creación de particiones | 6 |
| fdisk..... | 6 |
| Otras herramientas | 8 |
| Sistema de archivos | 8 |
| Rutas y nombre de archivos | 8 |
| Tipos de sistemas de archivos | 9 |
| EXT2 | 9 |
| EXT3 | 9 |
| EXT4..... | 10 |
| JFS | 10 |
| RaiserFS..... | 11 |
| XFS..... | 11 |
| ZFS..... | 11 |
| Creación de los sistemas de archivos | 12 |
| Comandos relacionados | 12 |
| Partición de intercambio | 13 |
| Montado de los sistemas de archivos | 13 |
| Comando mount | 13 |
| Comando umount | 14 |
| Archivo /etc/fstab | 14 |
| Campos dump y pass | 15 |
| Archivo /etc/mtab | 15 |
| Autofs | 15 |
| Archivo de configuración | 15 |
| Chequeo del sistema de archivos | 16 |

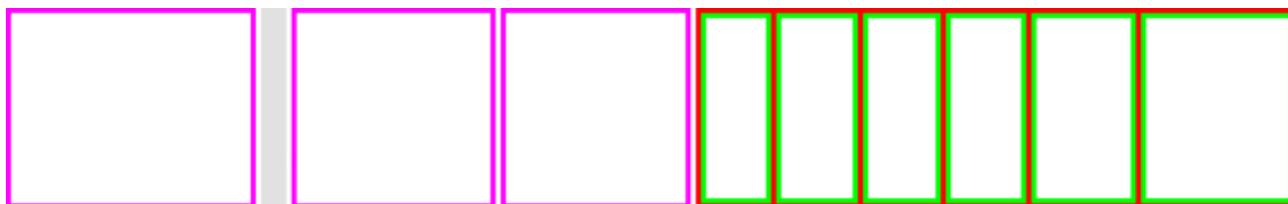
Particiones y sistemas de archivos

Una partición de disco, en informática, es el nombre genérico que recibe cada división presente en una sola unidad física de almacenamiento de datos. Toda partición tiene su propio sistema de archivos (formato); generalmente, casi cualquier sistema operativo interpreta, utiliza y manipula cada partición como un disco físico independiente, a pesar de que dichas particiones estén en un solo disco físico.

Información general

A toda partición se le da formato mediante un sistema de archivos como FAT, NTFS, ext4 ,ext3, ext2, FAT32, ReiserFS, Reiser4 u otro. En Windows, las particiones reconocidas son identificadas con una letra seguida por un signo de doble punto (p.ej. C:\) hasta cuatro particiones primarias; prácticamente todo tipo de discos magnéticos y memorias flash (como pendrives) pueden particionarse. Sin embargo, para tener la posibilidad de más particiones en un solo disco, se utilizan las particiones extendidas, las cuales pueden contener un número ilimitado de particiones lógicas en su interior. Para este último tipo de particiones, no es recomendado su uso para instalar ciertos sistemas operativos, sino que son más útiles para guardar documentos o ejecutables no indispensables para el sistema. Los discos ópticos (DVD, CD) no soportan particiones.

Es necesario tener en cuenta que solo las particiones primarias y lógicas pueden contener un sistema de archivos propio.



Representación gráfica de un disco particionado. Cada recuadro blanco representa algún sistema de archivos vacío. Los espacios en gris representan los espacios sin particionar del disco. Las particiones rodeadas por líneas moradas o violetas representan las particiones primarias. Las particiones rodeadas por bordes rojos representan la partición extendida (que es un tipo de partición primaria); y en su interior, se encuentran las particiones lógicas, rodeadas por los bordes de color verde.

Es común que los sistemas basados o similares a UNIX generalmente se usan hasta con 3 particiones: la principal, montada en el directorio raíz (/); a veces hay también una segunda que se usa para montar el directorio /home, el cual contiene las configuraciones de los usuarios, y finalmente, una tercera llamada swap, que se usa para la memoria virtual temporal. Sin embargo, 2 particiones (/ y swap); es el mínimo suficiente en estos sistemas operativos. Cabe decir además que las particiones de intercambio (swap) pueden instalarse sin problemas dentro de una partición lógica. Las particiones de intercambio, al igual que a la memoria RAM, no se les asigna un directorio; este tipo de particiones se usa para guardar ciertas réplicas de la memoria RAM, para que de esta forma la RAM tenga más espacio para las tareas en primer plano, guardando las tareas en segundo plano dentro de la partición de intercambio. Algunos sistemas tipo UNIX están diseñados para funcionar con una sola partición, sin embargo, estos diseños no son muy comunes.

Tipos de particiones

El formato o sistema de archivos de las particiones (p. ej. NTFS) no debe ser confundido con el tipo de partición (p. ej. partición primaria), ya que en realidad no tienen directamente mucho que ver. Independientemente del sistema de archivos de una partición (FAT, ext3, NTFS, etc.), existen 3 tipos diferentes de particiones:

- **Partición primaria:** Son las divisiones crudas o primarias del disco, solo puede haber 4 de éstas o 3 primarias y una extendida. Depende de una tabla de particiones. Un disco físico completamente formateado consiste, en realidad, de una partición primaria que ocupa todo el espacio del disco y posee un sistema de archivos. A este tipo de particiones, prácticamente cualquier sistema operativo puede detectarlas y asignarles una unidad, siempre y cuando el sistema operativo reconozca su formato (sistema de archivos).
- **Partición extendida:** También conocida como partición secundaria es otro tipo de partición que actúa como una partición primaria; sirve para contener infinidad de unidades lógicas en su interior. Fue ideada para romper la limitación de 4 particiones primarias en un solo disco físico. Solo puede existir una partición de este tipo por disco, y solo sirve para contener particiones lógicas. Por lo tanto, es el único tipo de partición que no soporta un sistema de archivos directamente.
- **Partición lógica:** Ocupa una porción de la partición extendida o la totalidad de la misma, la cual se ha formateado con un tipo específico de sistema de archivos (FAT32, NTFS, ext2,...) y se le ha asignado una unidad, así el sistema operativo reconoce las particiones lógicas o su sistema de archivos. Puede haber un máximo de 23 particiones lógicas en una partición extendida. Linux impone un máximo de 15, incluyendo las 4 primarias, en discos SCSI y en discos IDE 8963.

Particiones primarias

En los equipos PC, originales de IBM, estas particiones tradicionalmente usan una estructura llamada Tabla de particiones, ubicada al final del registro de arranque maestro (MBR, Master Boot Record). Esta tabla, que no puede contener más de 4 registros de particiones (también llamados *partition descriptors*), especifica para cada una su principio, final y tamaño en los diferentes modos de direccionamiento, así también como un solo número, llamado *partition type*, y un marcador que indica si la partición está activa o no (sólo puede haber una partición activa a la vez). El marcador se usa durante el arranque; después de que el BIOS cargue el registro de arranque maestro en la memoria y lo ejecute, el MBR de DOS comprueba la tabla de partición a su final y localiza la partición activa. Entonces carga el sector de arranque de esta partición en memoria y la ejecuta. A diferencia del registro de arranque maestro, generalmente independiente del sistema operativo, el sector de arranque está instalado junto con el sistema operativo y sabe cómo cargar el sistema ubicado en ese disco en particular.

Notar que mientras la presencia de un marcador activo se estandariza, no se utiliza en todos los gestores de arranque. Por ejemplo, los gestores LILO, GRUB (muy comunes en el sistema Linux) y XOSL no buscan en la tabla de particiones del MBR la partición activa; simplemente cargan una segunda etapa (que puede ser contenida en el resto del cilindro 0 ó en el sistema de archivos). Después de cargar la segunda etapa se puede cargar el sector de arranque de cualquiera de las particiones del disco (permitiendo al usuario seleccionar la partición), o si el gestor conoce cómo

localizar el kernel (núcleo) del sistema operativo en una de las particiones (puede permitir al usuario especificar opciones de kernel adicionales para propósitos de recuperación)estrategicos.

Particiones extendidas y lógicas

Cualquier versión del DOS puede leer sólo una partición FAT primaria en el disco duro. Esto unido al deterioro de la FAT con el uso y al aumento de tamaño de los discos movió a Microsoft a crear un esquema mejorado relativamente simple: una de las entradas de la tabla de partición principal pasó a llamarse partición extendida y recibió un número de tipo de partición especial (0x05). El campo inicio de partición tiene la ubicación del primer descriptor de la partición extendida, que a su vez tiene un campo similar con la ubicación de la siguiente; así se crea una lista enlazada de descriptores de partición. Los demás campos de una partición extendida son indefinidos, no tienen espacio asignado y no pueden usarse para almacenar datos. Las particiones iniciales de los elementos de la lista enlazada son las llamadas unidades lógicas; son espacios asignados y pueden almacenar datos. Los sistemas operativos antiguos ignoraban las particiones extendidas con número de tipo 0x05, y la compatibilidad se mantenía. Este esquema reemplaza al antiguo ya que todas las particiones de un disco duro se pueden poner dentro de una sola partición extendida. Por alguna razón, Microsoft no actualizó su sistema operativo DOS para arrancar desde una partición extendida, debido a que la necesidad para particiones primarias se preservaron. Por encima de éstas todavía se habría permitido una partición FAT primaria por unidad, significando todas las otras particiones FAT primarias deben tener sus números de tipo de partición prior cambiando al arranque DOS, para que ésta sea capaz de proceder. Esta técnica, usada por varios administradores de arranque populares, se llama ocultación de la partición. Sin embargo hay que tener en cuenta una quinta partición que se puede comprimir pero no es muy recomendable.

Razones para el uso de particiones

Algunos sistemas de archivos (p.e. versiones antiguas de sistemas FAT de Microsoft) tienen tamaños máximos más pequeños que los que el tamaño que proporciona un disco, siendo necesaria una partición de tamaño pequeño, para que sea posible el adecuado funcionamiento de este antiguo sistema de archivos.

Se puede guardar una copia de seguridad de los datos del usuario en otra partición del mismo disco, para evitar la pérdida de información importante. Esto es similar a un RAID, excepto en que está en el mismo disco.

En algunos sistemas operativos aconsejan más de una partición para funcionar, como por ejemplo, la partición de intercambio (swap) en los sistemas operativos basados en Linux.

A menudo, dos sistemas operativos no pueden coexistir en la misma partición, o usar diferentes formatos de disco "nativo". La unidad se particiona para diferentes sistemas operativos.

Uno de los principales usos que se le suele dar a las particiones (principalmente a la extendida) es la de almacenar toda la información del usuario (entiéndase música, fotos, vídeos, documentos), para que al momento de reinstalar algún sistema operativo se formatee únicamente la unidad que

lo contiene sin perder el resto de la información del usuario.

A lo largo de los años han aparecido numerosos sistemas de particionamiento, para casi todas las arquitecturas de ordenadores existentes. Muchos son relativamente transparentes y permiten la manipulación conveniente de las particiones de disco; algunos, sin embargo, son obsoletos.

Las tablas de particiones (MBR) sólo admiten hasta 2,2 TB por partición. Dado que sólo soportan 4 particiones primarias, el tamaño máximo admisible para un disco duro sería de 8,8 TB (el resto de capacidad no se podría utilizar). Como la arquitectura IBM PC es muy común, las tablas de partición probablemente subsistirán cierto tiempo. Sin embargo, existe un proyecto de Intel llamado Extensible Firmware Initiative (EFI) con el sistema GPT, que soporta teóricamente hasta 9,4 9,4 ZB.

Las ventajas del uso de particiones extendidas

Las particiones extendidas se inventaron para superar el límite de 4 particiones primarias máximas por cada disco duro y poder crear un número ilimitado de unidades lógicas, cada una con un sistema de archivos diferente de la otra. Todos los sistemas modernos (Linux, cualquier Windows basado en NT e incluso OS/2) son capaces de arrancar desde una unidad lógica. Sin embargo, el MBR por defecto utilizado por Windows y DOS sólo es capaz de continuar el proceso de arranque con una partición primaria. Cuando se utiliza este MBR, es necesario que exista por lo menos una partición primaria que contenga un cargador de arranque (comúnmente el NTLDR de Windows). Otros cargadores de arranque que reemplazan el MBR, como por ejemplo GRUB, no sufren de esta limitación.

Creación de particiones

fdisk

El comando que vamos a utilizar para manipular las particiones en nuestro Linux es **fdisk** y su sintaxis es la siguiente:

```
fdisk [opciones] dispositivo
```

donde **dispositivo** es el dispositivo del disco (/dev/hdx en IDE, /dev/sdx para SCSI o SATA)

Para poder utilizarlo debemos tener permiso de administrador.

```
# fdisk /dev/hdb
The number of cylinders for this disk is set to 20805.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
Command (m for help): m
```

```

Command action
  a  toggle a bootable flag
  b  edit bsd disklabel
  c  toggle the dos compatibility flag
  d  delete a partition
  l  list known partition types
  m  print this menu
  n  add a new partition
  o  create a new empty DOS partition table
  p  print the partition table
  q  quit without saving changes
  s  create a new empty Sun disklabel
  t  change a partition's system id
  u  change display/entry units
  v  verify the partition table
  w  write table to disk and exit
  x  extra functionality (experts only)

```

Para crear una partición primaria de 5 GB usamos n (new):

```

Command (m for help): n
Command action
  e  extended
  p  primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-20805, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-20805, default 20805): +5G

Command (m for help): p

Disk /dev/hdb: 10.7 GB, 10737418240 bytes
16 heads, 63 sectors/track, 20805 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1            1         9689     4883224+   83  Linux

```

Por defecto, crea la partición tipo Linux (Id 83) con l (list) vemos el tipo de particiones soportadas para cambiar el tipo de partición se usa t (type)

```

Command (m for help): t 1
Selected partition 1
Hex code (type L to list codes): 82
Changed system type of partition 1 to 82 (Linux swap / Solaris)

Command (m for help): p

Disk /dev/hdb: 10.7 GB, 10737418240 bytes
16 heads, 63 sectors/track, 20805 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1            1         9689     4883224+   82  Linux swap /
Solaris

```

Para que se guarden los cambios debemos usar w (write)

Otras herramientas

Fdisk no es la única herramienta para manipular particiones. Por suerte el mundo GNU nos ofrece muchas herramientas distintas para realizar lo mismo.

- cfdisk : interfaz para el fdisk (también escribe la tabla de particiones)
- gparted : programa de GNU que permite crear, destruir, cambiar el tamaño, chequear y copiar particiones
- qtparted : clónico del Partition Magic para Linux (basado en libparted)

Sistema de archivos

Los sistemas de archivos o archivos (en inglés:filesystem), estructuran la información guardada en una unidad de almacenamiento (normalmente un disco duro de una computadora), que luego será representada ya sea textual o gráficamente utilizando un gestor de archivos. La mayoría de los sistemas operativos manejan su propio sistema de archivos.

Lo habitual es utilizar dispositivos de almacenamiento de datos que permiten el acceso a los datos como una cadena de bloques de un mismo tamaño, a veces llamados sectores, usualmente de 512 bytes de longitud. El software del sistema de archivos es responsable de la organización de estos sectores en archivos y directorios y mantiene un registro de qué sectores pertenecen a qué archivos y cuáles no han sido utilizados. En la práctica, un sistema de archivos también puede ser utilizado para acceder a datos generados dinámicamente, como los recibidos a través de una conexión de red (sin la intervención de un dispositivo de almacenamiento).

Los sistemas de archivos tradicionales proveen métodos para crear, mover, renombrar y eliminar tanto archivos como directorios, pero carecen de métodos para crear, por ejemplo, enlaces adicionales a un directorio o archivo (enlace duro en Unix) o renombrar enlaces padres (".." en Unix).

El acceso seguro a sistemas de archivos básicos puede estar basado en los esquemas de lista de control de acceso o capacidades. Las listas de control de acceso hace décadas que demostraron ser inseguras, por lo que los sistemas operativos experimentales utilizan el acceso por capacidades. Los sistemas operativos comerciales aún funcionan con listas de control de acceso.

Rutas y nombre de archivos

Normalmente los archivos y carpetas se organizan jerárquicamente.

La estructura de directorios suele ser jerárquica, ramificada o "en árbol", aunque en algún caso podría ser plana. En algunos sistemas de archivos los nombres de archivos son estructurados, con sintaxis especiales para extensiones de archivos y números de versión. En otros, los nombres de archivos son simplemente cadenas de texto y los metadatos de cada archivo son alojados separadamente.

En los sistemas de archivos jerárquicos, usualmente, se declara la ubicación precisa de un archivo con una cadena de texto llamada "ruta" —o path en inglés—. La nomenclatura para rutas varía ligeramente de sistema en sistema, pero mantienen por lo general una misma estructura. Una ruta

viene dada por una sucesión de nombres de directorios y subdirectorios, ordenados jerárquicamente de izquierda a derecha y separados por algún carácter especial que suele ser una diagonal ('/') o diagonal invertida ('\') y puede terminar en el nombre de un archivo presente en la última rama de directorios especificada.

En un sistema tipo Unix como GNU/Linux, la ruta para la canción llamada "canción.ogg" del usuario "álvaro" sería algo como:

```
/home/álvaro/música/canción.ogg
```

donde:

- '/' representa el directorio raíz donde está montado todo el sistema de archivos.
- '**home/álvaro/música/**' es la ruta del archivo.
- '**canción.ogg**' es el nombre del archivo.

Tipos de sistemas de archivos

EXT2

ext2 (second extended filesystem o "segundo sistema de archivos extendido") es un sistema de archivos para el kernel Linux. Fue diseñado originalmente por Rémy Card. La principal desventaja de ext2 es que no implementa el registro por diario (en inglés Journaling) que sí implementa su sucesor ext3, el cual es totalmente compatible.

ext2 fue el sistema de archivos por defecto de las distribuciones de Linux Red Hat Linux, Fedora Core y Debian hasta ser reemplazado por su sucesor ext3.

El sistema de archivos tiene un tipo de tabla FAT de tamaño fijo, donde se almacenan los i-nodos. Los i-nodos son una versión muy mejorada de FAT, donde un puntero i-nodo almacena información del archivo (ruta o path, tamaño, ubicación física). En cuanto a la ubicación, es una referencia a un sector del disco donde están todos y cada una de las referencias a los bloques del archivo fragmentado. Estos bloques son de tamaño especificable cuando se crea el sistema de archivos, desde los 512 bytes hasta los 4 kB, lo cual asegura un buen aprovechamiento del espacio libre con archivos pequeños.

EXT3

ext3 (third extended filesystem o "tercer sistema de archivos extendido") es un sistema de archivos con registro por diario (journaling). Es el sistema de archivo más usado en distribuciones Linux, aunque en la actualidad está siendo reemplazado por su sucesor, ext4.

La principal diferencia con ext2 es el registro por diario (journaling). Un sistema de archivos ext3 puede ser montado y usado como un sistema de archivos ext2. Otra diferencia importante es que ext3 utiliza un árbol binario balanceado (árbol AVL) e incorpora el asignador de bloques de disco Orlov.

EXT4

ext4 (fourth extended filesystem o «cuarto sistema de archivos extendido») es un sistema de archivos con registro por diario (en inglés Journaling), anunciado el 10 de octubre de 2006 por Andrew Morton, como una mejora compatible de ext3. El 25 de diciembre de 2008 se publicó el kernel Linux 2.6.28, que elimina ya la etiqueta de "experimental" de código de ext4.

Las principales mejoras son:

- Soporte de volúmenes de hasta 1024 PiB.
- Soporte añadido de extent.
- Menor uso del CPU.
- Mejoras en la velocidad de lectura y escritura.

JFS

Journaling File System (JFS) es un sistema de archivos de 64-bit con respaldo de transacciones creado por IBM. Está disponible bajo la licencia GNU GPL. Existen versiones para AIX, eComStation, OS/2, sistemas operativos Linux y HP-UX

Fue diseñado con la idea de conseguir "servidores de alto rendimiento y servidores de archivos de altas prestaciones, asociados a e-business". Según se lee en la documentación y el código fuente, va a pasar un tiempo antes de que la adaptación a Linux este finalizada e incluida en la distribución estándar del kernel.

JFS utiliza un método interesante para organizar los bloques vacíos, estructurándolos en un árbol y usa una técnica especial para agrupar bloques lógicos vacíos.

JFS fue desarrollado para AIX. La primera versión para Linux fue distribuida en el verano de 2000. La versión 1.0.0 salió a la luz en el año 2001. JFS está diseñado para cumplir las exigencias del entorno de un servidor de alto rendimiento en el que sólo cuenta el funcionamiento. Al ser un sistema de archivos de 64 bits, JFS soporta archivos grandes y particiones LFS (del inglés Large File Support), lo cual es una ventaja más para los entornos de servidor.

RaiserFS

ReiserFS es un sistema de archivos de propósito general, diseñado e implementado por un equipo de la empresa Namesys, liderado por Hans Reiser.

Actualmente es soportado por Linux y existen planes de futuro para incluirlo en otros sistemas operativos.

También es soportado por Windows (de forma no oficial), aunque por el momento de manera inestable y rudimentaria (ReiserFS bajo windows).

A partir de la versión 2.4.1 de Linux, ReiserFS se convirtió en el primer sistema de archivos con journal en ser incluido en el núcleo estándar. También es el sistema de archivos por defecto en varias distribuciones, como SuSE (excepto en openSuSE 10.2 cuyo formato por defecto es ext3), Xandros, Yoper, Linspire, Kurumin Linux, FTOSX, Libranet y Knoppix.

Con la excepción de actualizaciones de seguridad y parches críticos, Namesys ha cesado el desarrollo de ReiserFS (también llamado reiser3) para centrarse en Reiser4, el sucesor de este sistema de archivos.

XFS

XFS es un sistema de archivos de 64 bits con journaling de alto rendimiento creado por SGI (antiguamente Silicon Graphics Inc.) para su implementación de UNIX llamada IRIX. En mayo de 2000, SGI liberó XFS bajo una licencia de código abierto.

XFS se incorporó a Linux a partir de la versión 2.4.25, cuando Marcelo Tosatti (responsable de la rama 2.4) lo consideró lo suficientemente estable para incorporarlo en la rama principal de desarrollo del kernel. Los programas de instalación de las distribuciones de SuSE, Gentoo, Mandriva, Slackware, Fedora Core, Ubuntu y Debian ofrecen XFS como un sistema de archivos más. En FreeBSD el soporte para solo-lectura de XFS se añadió a partir de Diciembre de 2005 y en Junio de 2006 un soporte experimental de escritura fue incorporado a FreeBSD-7.0-CURRENT.

ZFS

ZFS es un sistema de archivos desarrollado por Sun Microsystems para su sistema operativo Solaris. El significado original era 'Zettabyte File System', pero ahora es un acrónimo recursivo.

El anuncio oficial de ZFS se produjo en Septiembre del 2004.1 El código fuente del producto final

se integró en la rama principal de desarrollo de Solaris el 31 de octubre del 2005 y fue lanzado el 16 de noviembre de 2005 como parte del build 27 de OpenSolaris.

ZFS fue diseñado e implementado por un equipo de Sun liderado por Jeff Bonwick.

ZFS destaca por su gran capacidad, integración de los conceptos anteriormente separados de sistema de archivos y administrador de volúmenes en un solo producto, nueva estructura sobre el disco, sistemas de archivos ligeros, y una administración de espacios de almacenamiento sencilla.

Creación de los sistemas de archivos

Sobre cada partición debemos crear sistemas de archivos con el comando `mkfs`

```
mkfs [-V] [-t filesystem] dispositivo [n_bloques]
```

Opciones:

- **-V** verbose
- **-t** filesystem tipo de sistema de archivos a crear (ext2, ext3, xfs, etc.) . Si no se especifica se crea el por defecto del sistema (en Linux ext2)
- **n_bloques** número de bloques usados para el sistema de archivos (si no se pone, se usa toda la partición)

`mkfs` es un front-end a distintos comandos, que permiten crear particiones de los tipos específicos:

- **mkfs.ext2** o **mke2fs** crea sistemas ext2
- **mkfs.ext3** crea sistemas ext3, equivalente a `mkfs.ext2 -j`
- **mkfs.jfs**, **mkfs.reiserfs**, **mkfs.xfs** crean sistemas JFS, ReiserFS y XFS, respectivamente
- **mkfs.msdos**, **mkfs.vfat** crea sistemas MS-DOS
- **mkswap** crea un sistema de archivos de tipo Linux swap

Cada uno de estos comandos pueden tener distintas opciones por lo cual es altamente recomendable que miremos las MAN de cada comando.

Comandos relacionados

- **dumpe2fs** muestra información de sistemas de archivos ext2/ext3 información sobre inodos, bloques y grupos
- **tune2fs** permite ajustar parámetros en sistemas ext2/ext3 . Por ejemplo define el intervalo entre chequeos automáticos, convierte ext2 en ext3, etc.
- **e2label** cambia la etiqueta de un sistema ext2/ext3

existen comandos similares para otros tipos de sistemas de archivos, p.e. reiserfstune, jfs_tune, etc.

Partición de intercambio

Si lo que creamos es una partición de intercambio, la debemos inicializar con mkswap

```
# fdisk -l /dev/hdb
Disk /dev/hdb: 10.7 GB, 10737418240 bytes
16 heads, 63 sectors/track, 20805 cylinders
Units = cylinders of 1008 * 512 = 516096 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1            1          9689    4883224+   83  Linux
/dev/hdb2           9690       20805    5602464   82  Linux swap /
Solaris
```

```
# mkswap /dev/hdb2
Setting up swapspace version 1, size = 5736919 kB
no label, UUID=a6c2849b-c33e-478e-8a8d-fecfe3f18f6d
```

Una vez creada debemos activarla con swapon

```
# swapon /dev/hdb2
# swapon -s
Filename                Type              Size      Used      Priority
/dev/hda7                partition        377488    0         -1
/dev/hdb2                partition        5602456  0         -2
```

Finalmente, para que se active en el arranque, debe incluirse la entrada correspondiente en el archivo /etc/fstab

```
/dev/hda2  none  swap  sw  0  0
```

Montado de los sistemas de archivos

Para poder acceder a los sistemas de archivos debemos montarlos los comandos para montar y desmontar son: **mount** y **umount**

Comando mount

Permite asociar (montar) directorios a sistemas de archivos

Ejemplo

```
$ mount -t ext3 /dev/hdb1 /home2
```

Formato

```
mount [opciones] [-t tipo] [disp.] [dir.]
```

Algunas opciones:

- **tipo** tipo de sistema de archivos (ext2, ext3, reiserfs, vfat, etc.); si se pone auto intenta determinar de forma automática
- **-a** monta los filesystems listados en /etc/fstab

- **-r/-w** monta los sistemas de sólo lectura/escritura
- **-f** simulación; usado con **-v** (verbose) para chequear
- **-n** monta sin añadir la entrada a `/etc/mstab`; se usa cuando `/etc` es sólo lectura
- **-o** opciones de montaje; formato igual al usado en el archivo `fstab`

Comando `umount`

Desmonta los sistemas de archivos

Formato

```
umount [opciones] directorio
```

Ejemplo:

```
$ umount /home2
```

Algunas opciones

- **-a** desmonta los filesystems listados en `/etc/mstab`
- **-r** si falla, intenta remontar sólo lectura
- **-f** fuerza el desmontado

Si hay algún proceso bloqueando el filesystem, este no se puede desmontar por lo tendremos que utilizar el comando **fuser -c directorio** para ver el **PID** del proceso

Archivo `/etc/fstab`

Al iniciar el sistema se montan los filesystems listados en `/etc/fstab` cada línea del archivo tiene las siguientes columnas:

```
(file system) (mount point) (tipo) (opciones) (dump) (pass)
```

Ejemplo:

```
/dev/hda9 /home ext3 defaults 0 2
```

Alguna de las opciones son:

- **rw** monta tipo lectura/escritura
- **ro** sólo lectura
- **auto/noauto** monta/no monta con `mount -a` (monta/no monta al inicio)
- **exec/noexec** Permite/no permite la ejecución de archivos binarios en la partición
- **suid/nosuid** permite/no permite que los bits `setuid` y `setgid` tengan efecto
- **dev/nodev** interpreta/no interpreta dispositivos de bloques o caracteres en el filesystem
- **async** toda la I/O se realiza de forma asíncrona
- **user** puede montarlo un usuario (y sólo el que lo monta puede desmontarlo); implica las opciones **noexec**, **nosuid** y **nodev**, a menos que se fuercen (p.e. `user,exec,suid,dev`)
- **users** puede montarlo/desmontarlo un usuario y el que desmonta no tiene que ser el que lo montó; implica las mismas opciones que `user`
- **defaults** selecciona opciones por defecto (`rw, suid, dev, exec, auto, nouser y async`)

Si un directorio aparece listado en el fstab puede montarse sin especificar el dispositivo:

```
$ mount /home
```

Campos dump y pass

- **dump** lo usa el comando dump para determinar de que filesystems hacer copias de seguridad valor 1 o 0 según si la partición va a tener un backup controlado por dump o no (normalmente no se usa)
- **pass**: lo usa el comando fsck para determinar el orden en que se chequean los filesystems al iniciar el sistema si 0, el filesystem no se chequea. Si es > 0, los filesystems se chequean en el orden indicado por los números. Si varios tienen el mismo número, se chequean en paralelo (si es posible) normalmente / tendrá 1 y el resto 2

Archivo /etc/mtab

Contiene una lista de los filesystem que están montados en el sistema

Ejemplo de archivo /etc/mtab

```
$ cat /etc/mtab
/dev/hda1 / ext3 rw,errors=remount-ro 0 0
proc /proc proc rw 0 0
devpts /dev/pts devpts rw,gid=5,mode=620 0 0
tmpfs /dev/shm tmpfs rw 0 0
/dev/hda9 /home ext3 rw 0 0
/dev/hda8 /tmp ext3 rw 0 0
/dev/hda5 /usr ext3 rw 0 0
/dev/hda6 /var ext3 rw 0 0
usbfs /proc/bus/usb usbfs rw 0 0
/dev/hdb1 /home2 ext2 rw,nodev 0 0
```

Autofs

Sistema que permite montar los filesystems “bajo demanda” cuando se accede al directorio, este se monta se desmonta automáticamente después de un tiempo de inactividad (por defecto, 5 minutos) y suele usarse para montar sistemas remotos con NFS

Archivo de configuración

- /etc/auto.master define los puntos de montaje por cada uno de los puntos definidos, se inicia un proceso automount usando los parámetros indicados

Ejemplo de auto.master:

```
/home /etc/auto.home
/misc /etc/auto.misc --timeout 60
```

Los archivos le indican al automount los filesystems a montar

Ejemplo de auto.misc

```
cdrom      -fstype=iso9660,ro  :/dev/cdrom
floppy     -fstype=auto        :/dev/fd0
windoz     -fstype=vfat       :/dev/hda1
```

esto monta el cdrom, el floppy y la partición /dev/hda1 en los directorios /misc/cdrom, /misc/floppy y /misc/windoz respectivamente

Chequeo del sistema de archivos

Periódicamente es necesario chequear los sistemas de archivos el comando básico para chequeo y reparación es fsck.

Al igual que mkfs, fsck es un front-end a comandos específicos para cada filesystem:

- e2fsck, fsck.ext2 o fsck.ext3 chequean sistemas ext2/ext3
- fsck.jfs, fsck.reiserfs, fsck.xfs para JFS, ReiserFS y XFS
- fsck.msdos, fsck.vfat para sistemas MS-DOS

Alguno de los errores que pueden aparecer se deben a:

- Varios archivos que usan el mismo bloque
- Bloques marcados libres y ocupados simultáneamente
- Número de enlaces erróneo
- Nodos-i conteniendo información pero que no están en la entrada del directorio (la información se recupera en el directorio lost+found con el número de nodo-i)
- Entradas del directorio que apuntan a nodos-i ilegales o vacíos

Algunas de las opciones de fsck son:

- -t filesystem tipo de filesystem a chequear
- -A chequea los filesystems listados en /etc/fstab
- -N no ejecuta; simplemente indica lo que haría
- -R usado con -A no chequea el filesystem raíz