

# MOUNT

## NOMBRE

mount - monta un sistema de ficheros

## SINOPSIS

mount [-hV]

mount -a [-fFnrsvw] [-t tipo-sfv]

mount [-fnrsvw] [-o opciones [...]] dispositivo | dir

mount [-fnrsvw] [-t tipo-sfv] [-o opciones] dispositivo dir

## DESCRIPCIÓN

Todos los ficheros accesibles en un sistema Unix están dispuestos en un gran árbol, la jerarquía de ficheros, con la raíz en /. Estos ficheros pueden estar distribuidos sobre varios dispositivos. La orden mount sirve para pegar el sistema de ficheros encontrado en algún dispositivo al gran árbol de ficheros. De modo análogo pero al revés, la orden umount(8) lo despegará de él de nuevo.

La forma más normal de la orden mount es

mount -t tipo dispositivo dir

Esto le dice al núcleo que anexe el sistema de ficheros que encuentre en dispositivo (que es del tipo tipo) al directorio dir. Los contenidos anteriores (si había), así como el propietario y permisos de dir se vuelven invisibles (están ocultos, tapados), y mientras este sistema de ficheros permanezca montado, el nombre de camino dir, también llamado punto de montaje, se refiere a la raíz del sistema de ficheros en dispositivo.

Tres formas de llamada no montan realmente nada:

`mount -h`

muestra un mensaje de ayuda;

`mount -V`

muestra una cadena de caracteres relativa a la versión; y un solo

`mount [-t tipo]`

lista todos los sistemas de ficheros montados (del tipo tipo) --vea más adelante--.

El sistema de ficheros `proc` no está asociado a ningún dispositivo o fichero especial, y cuando se monte, se puede emplear una palabra arbitraria, como `proc`, en vez de una especificación de dispositivo. (La elección personal `none` es menos afortunada: el mensaje de error ‘`none busy`’ (nadie [está] ocupado) de `umount` puede confundir.)

La mayoría de dispositivos se indican mediante un nombre de fichero (de un dispositivo especial de bloques), como `/dev/sda1`, pero hay otras posibilidades. Por ejemplo, en el caso de un montaje por NFS, dispositivo puede ser algo como `knuth.cwi.nl:/dir`.

El fichero `/etc/fstab` (vea `fstab(5)`), puede contener renglones que describan qué dispositivos se montan usualmente dónde, empleando cuáles opciones. Este fichero se emplea de tres formas:

(i) La orden

`mount -a [-t tipo]`

(usualmente dado en un guión de arranque) hace que todos los sistemas de ficheros mencionados en `fstab` (del tipo adecuado) se monten como se indique, excepto aquéllos cuya línea contenga la palabra clave `noauto`. Añadir la opción `-F` hará que `mount` se bifurque, de forma que los sistemas de ficheros se monten simultáneamente, en paralelo.

(ii) Cuando se monte un sistema de ficheros mencionado en `fstab`, basta con dar sólo el dispositivo o el punto de montaje.

(iii) Normalmente, sólo el superusuario puede montar sistemas de ficheros. Sin embargo, cuando `fstab` contiene la opción `user` en una línea, entonces cualquiera puede montar el sistema de ficheros corre-

spondiente.

Así, dada la línea

```
/dev/cdrom /cdrom iso9660 ro,user,noauto,unhide
```

cualquier usuario puede montar el sistema de ficheros de tipo iso9660 encontrado en su CD-ROM mediante la orden

```
mount /dev/cdrom
```

o

```
mount /cdrom
```

Para más detalles, vea fstab(5).

Los programas `mount` y `umount` mantienen una lista de los sistemas de ficheros montados actualmente en el fichero `/etc/mtab`. Si no se dan argumentos a `mount`, se muestra esta lista. Cuando el sistema de ficheros `proc` esté montado (digamos en `/proc`), los ficheros `/etc/mtab` y `/proc/mounts` tienen contenidos muy similares. El primero tiene algo más de información, como las opciones de montaje empleadas, pero no está necesariamente al día (cf. la opción `-n` más abajo). Es posible reemplazar `/etc/mtab` por un enlace simbólico a `/proc/mounts`, pero de esta forma se perderá algo de información, y en particular trabajar con el dispositivo de bucle será menos conveniente.

## OPCIONES

El conjunto completo de opciones empleadas por una llamada de `mount` se determina primeramente extrayendo las opciones para el sistema de ficheros desde la tabla de `fstab`, luego aplicando las opciones especificadas al argumento `-o`, y finalmente aplicando una opción `-r` o `-w` cuando esté presente.

Opciones disponibles para la orden `mount`:

`-V` Saca la versión.

`-h` Muestra un mensaje de ayuda.

- v Modo prolijo.
  
- a Monta todos los sistemas de ficheros (de los tipos dados) mencionados en fstab.
  
- F (Empleado en conjunción con -a.) Bifurca una nueva encarnación de mount para cada dispositivo. Esto efectuará los montajes en diferentes dispositivos o servidores de NFS en paralelo. Esto tiene la ventaja de que es más rápido; también las esperas en NFS van en paralelo. Una desventaja es que los montajes se hacen en un orden no definido. Así, Ud. no puede emplear esta opción si quiere montar tanto /usr como /usr/spool.
  
- f Causa que todo se haga excepto la llamada al sistema real; en otras palabras, esto no monta el sistema de ficheros, es un “engaño” (fake). Esta opción es útil en conjunción con la otra opción -v para determinar lo que la orden mount está intentando hacer. También se puede emplear para añadir entradas para dispositivos que fueron montados anteriormente con la opción -n.
  
- n Monta sin escribir en /etc/mtab. Esto es necesario por ejemplo cuando /etc está en un sistema de ficheros de lectura exclusiva.
  
- s Tolera opciones de montaje medio malas en vez de fallar. Esto hará que se pasen por alto opciones de montaje no admitidas por un tipo de sistema de ficheros. No todos los sistemas de ficheros admiten esta opción. Esta opción existe para dar soporte al automontador de Linux basado en autofs.
  
- r Monta el sistema de ficheros de lectura exclusiva. Un sinónimo es -o ro.
  
- w Monta el sistema de ficheros de lectura y escritura. Esto es lo predeterminado. Un sinónimo es -o rw.
  
- t tipo-sfv

El argumento que sigue a -t se emplea para indicar el tipo del

sistema de ficheros. Los tipos de sistemas de ficheros que se soportan actualmente se listan en `/usr/src/linux/fs/filesystems.c`: minix, ext, ext2, xiafs, hpfs, msdos, umsdos, vfat, proc, nfs, iso9660, smbfs, ncpfs, affs, ufs, romfs, sysv, xenix, coherent. Note que los tres últimos son equivalentes y que xenix y coherent se quitarán en un futuro no muy lejano: emplee sysv siempre en su lugar. Desde la versión 2.1.21 del núcleo, los tipos ext y xiafs ya no existen.

El tipo iso9660 es el predeterminado. Si no se da la opción `-t` o si se especifica el tipo auto, se mira el súper-bloque para ver el tipo del sistema de ficheros (se soportan minix, ext, ext2, xiafs, iso9660, romfs). Si esta prueba falla y `/proc/filesystems` existe, entonces se intentará con todos los sistemas de ficheros listados allí, excepto aquéllos etiquetados "nodev" (como por ejemplo proc y nfs).

Observe que el tipo auto puede ser útil para disquetes montados por usuarios. Advertencia: las pruebas emplean una heurística (la presencia de 'magia' apropiada) y podrían reconocer un tipo de sistemas de ficheros equivocado.

Se puede especificar más de un tipo en una lista de tipos separados por comas. La lista de los tipos de sistemas de ficheros puede prefijarse por `no` para especificar los tipos de sistemas de ficheros sobre los que no se deberá tomar ninguna opción. (Esto puede tener sentido con la opción `-a`.)

Por ejemplo, la orden:

```
mount -a -t nomsdos,ext
```

monta todos los sistemas de ficheros listados en `/etc/fstab` excepto los de tipo msdos o ext.

- o Las opciones se especifican mediante la bandera `-o` seguida por una lista de opciones separadas por comas. Algunas de estas opciones sólo son útiles cuando aparecen en el fichero `/etc/fstab`. Las siguientes opciones se aplican a cualquier sis-

tema de ficheros que se esté montando:

**async** Toda la E/S al sistema de ficheros debería hacerse asíncronamente.

**atime** Actualizar el tiempo de acceso al nodo-í para cada acceso. Esto es lo predeterminado.

**auto** Puede montarse con la opción -a.

**defaults**

Emplear las opciones predeterminadas: rw, suid, dev, exec, auto, nouser y async.

**dev** Interpretar dispositivos especiales de caracteres o bloques en el sistema de ficheros.

**exec** Permitir la ejecución de binarios.

**noatime**

No actualizar los tiempos de acceso a nodo-í en este sistema de ficheros (p.ej., para un acceso más rápido en el directorio de 'spool' de las noticias o grupos de discusión ('news') para hacer más rápido un servidor de noticias).

**noauto** Sólo puede montarse explícitamente (esto es, la opción -a no hará que el sistema de ficheros se monte).

**nodev** No interpretar o reconocer dispositivos o ficheros especiales de bloques o caracteres en este sistema de ficheros.

**noexec** No permitir la ejecución de ningún binario en el sistema de ficheros montado. Esta opción puede ser útil para un servidor que tiene sistemas de ficheros que contienen binarios para otras arquitecturas distintas de la suya.

**nosuid** No permitir el efecto de los bits SUID ni SGID.

**nouser** Prohibir a un usuario ordinario (esto es, distinto de root) montar el sistema de ficheros. Esto es lo predeterminado.

**remount**

Intentar re-montar un sistema de ficheros ya montado. Esto se emplea comúnmente para cambiar las opciones de montaje en un sistema de ficheros, especialmente para que se pueda escribir en un sistema de ficheros que estaba de lectura exclusiva.

**ro** Montar el sistema de ficheros de lectura exclusiva.

**rw** Montar el sistema de ficheros de lectura y escritura.

**suid** Permitir el efecto de los bits SUID y SGID.

**sync** Toda la E/S al sistema de ficheros debería hacerse síncronamente.

**user** Permitir a un usuario ordinario montar el sistema de ficheros. Esta opción implica las opciones noexec, nosuid y nodev (a menos que se sustituyan por otras subsiguientes, como en la línea de opciones user,exec,dev,suid).

Las siguientes opciones se aplican solamente a ciertos sistemas de ficheros. Las hemos clasificado por sistemas de ficheros. Todas siguen a la bandera -o.

**uid=valor y gid=valor**

Establecen el propietario y grupo del raíz del sistema de ficheros (predeterminados: uid = gid = 0, pero con la opción uid o gid sin especificar el valor, se toma el uid o el gid del proceso en curso).

**setuid=valor y setgid=valor**

Establecen el propietario y el grupo de todos los ficheros.

mode=valor

Establece el modo de todos los ficheros a valor & 0777 descartando los permisos originales. Añade permiso de paso a los directorios que lo tengan de lectura. El valor se da en octal.

protect

No permitir ningún cambio en los bits de protección del sistema de ficheros.

usemp Poner el uid y gid de la raíz del sistema de ficheros al uid y gid del punto de montaje a partir del primer sync o umount, y luego borrar esta opción. Extraño...

verbose

Mostrar un mensaje informativo para cada montaje con éxito.

prefix=cadena

Prefijo empleado antes del nombre del volumen, cuando se sigue un enlace.

volume=cadena

Prefijo (de como mucho 30 de longitud) empleado antes de '/' cuando se sigue un enlace simbólico.

reserved=valor

(Valor predeterminado: 2.) Número de bloques no usados al principio del dispositivo.

root=valor

Para dar explícitamente la localización del bloque raíz.

bs=valor

Para dar el tamaño de bloque. Los valores permitidos son 512, 1024, 2048 y 4096.



grpquota / noquota / quota / usrquota

Estas opciones se aceptan pero no tienen efecto.

Ninguna.

Ninguna. Advierta que el sistema de ficheros 'ext' está anticuado. No lo use. Desde la versión 2.1.21 de Linux, extfs ya no forma parte de los fuentes del núcleo.

El sistema de ficheros 'ext2' es el sistema de ficheros estándar.

Debido a un fallo del núcleo, puede montarse con cualquier opción de montaje. Esto se corrigió en Linux 2.0.4.

bsddf / minixdf

Establece el comportamiento para la llamada al sistema statfs.

El comportamiento con minixdf es devolver en el campo f\_blocks el número total de bloques del sistema de ficheros, mientras que el comportamiento con bsddf (que es el predeterminado) es sustraer los bloques de arriba empleados por el sistema de ficheros ext2 y no disponibles para almacenamiento de ficheros. Así:

```
% mount /k -o minixdf; df /k; umount /k
```

```
Filesystem 1024-blocks Used Available Capacity Mounted on
/dev/sda6 2630655 86954 2412169 3% /k
```

```
% mount /k -o bsddf; df /k; umount /k
```

```
Filesystem 1024-blocks Used Available Capacity Mounted on
/dev/sda6 2543714 13 2412169 0% /k
```

(Advierta que este ejemplo muestra que uno puede añadir opciones en la línea de órdenes a las opciones dadas en /etc/fstab.)

check / check=normal / check=strict

Establece el nivel de comprobaciones. Cuando al menos una de estas opciones se da (y check=normal está puesta de forma predeterminada), se comprueban los mapas de bits de los nodos-í y de los bloques cuando se realiza el montaje (lo cual puede llevar medio minuto o algo así en un disco grande). Con comprobación estricta (strict), la desasignación de bloques comprueba que el bloque a liberar esté en la zona de datos.

check=none / nocheck

No se realiza ninguna comprobación.

debug Muestra información de depuración en cada (re-)montaje.

errors=continue / errors=remount-ro / errors=panic

Define el comportamiento cuando se encuentra un error. (O bien no hace caso de errores y simplemente marca el sistema de ficheros como erróneo y continúa, o bien re-monta el sistema de ficheros como de lectura exclusiva, o bien se produce un pánico y se para el sistema.) Lo predeterminado está establecido en el súper-bloque del sistema de ficheros, y puede cambiarse empleando tune2fs(8).

grpuid o bsdgroups / nogrpuid o sysvgroups

Estas opciones definen qué identificador de grupo recibe un fichero recién creado. Cuando se establece grpuid, toma el GID del directorio en el que se crea; de otro modo (el predeterminado), toma el FSGID del proceso en curso, a menos que el directorio tenga activado el bit setgid, en cuyo caso toma el GID del directorio padre, y también toma el bit setgid si es él mismo un directorio.

resgid=n y resuid=n

El sistema de ficheros ext2 reserva un cierto porcentaje del espacio disponible (por omisión el 5 %, consulte mke2fs(8) y tune2fs(8)). Estas opciones determinan quién puede usar los bloques reservados. (En pocas palabras: cualquiera que tenga el UID especificado, o pertenezca al grupo especificado.)

sb=n En vez del bloque número 1, usa el bloque n como súper-bloque.

Esto podría ser de utilidad cuando el sistema de ficheros resultara dañado. Usualmente, las copias del súper-bloque se encuentran cada 8192 bloques: en los bloques 1, 8193, 16385, ... (De esta forma, uno llega a tener cientos o incluso miles de copias del súper-bloque en un sistema de ficheros grande. No conozco ninguna opción de mke2fs que haga que se escriban menos copias.)

grpquota / noquota / quota / usrquota

Estas opciones se aceptan pero no se tienen en cuenta.

(Nota: fat no es un sistema de ficheros separado, sino una parte común de los sistemas de ficheros msdos, umsdos y vfat.)

blocksize=512 / blocksize=1024

Establece el tamaño de bloque (el predeterminado es 512).

uid=valor y gid=valor

Establece el propietario y grupo de todos los ficheros. (Por omisión: el UID y GID del proceso en curso.)

umask=valor

Establece la umask (la máscara de bits de los permisos que no están presentes). El valor predeterminado es el de la umask del proceso en curso. El valor se da en octal.

check=valor

Se puede escoger tres niveles diferentes de elección (en cuanto al comportamiento con nombres de ficheros):

r[elaxed]

Se aceptan y tratan por igual las letras mayúsculas y las minúsculas, las partes con nombres largos se truncan (p.ej.: nombremuy largo.fuubar se convierte en nombremu.fuu), se aceptan en cada parte del nombre (nombre y extensión) espacios iniciales y por medio.

n[ormal]

Como "relaxed", pero se rechazan muchos caracteres especiales (\*, ?, <, espacios, etc.). Esto es el comportamiento predeterminado.

s[trict]

Como "normal", pero los nombres no pueden contener partes largas y los caracteres especiales que de vez en cuando

se usan en Linux pero no son aceptados por MS-DOS, se rechazan (+, =, espacios, etc.).

conv=b[inary] / conv=t[ext] / conv=a[uto]

El sistema de ficheros fat puede efectuar conversiones CR/LF <--> NL (formato de texto de MS-DOS a/desde formato de texto UNIX) en el núcleo. Se dispone de los siguientes modos de conversión.

binary Ninguna traducción. Esto es lo predeterminado.

text Traducción CR/LF <--> NL en todos los ficheros.

auto Traducción CR/LF <--> NL en todos los ficheros que no tengan una extensión "binaria bien conocida". La lista de extensiones conocidas se puede encontrar al principio de /usr/src/linux/fs/fat/misc.c (en 2.0, la lista es: exe, com, bin, app, sys, drv, ovl, ovr, obj, lib, dll, pif (código de programas); arc, zip, lha, lzh, zoo, tar, z, arj (archivadores comunes); tz, taz, tzp, tpz (abreviaturas de tar.Z y tar.zip), gz, tgz, deb (archivos comprimidos con gzip, paquetes tar.gz y Debian); gif, bmp, tif, gl, jpg, pcx (formatos gráficos); tfm, vf, gf, pk, pxl y dvi (relativos a TeX)).

Los programas que hagan lseek's computados no querrán que el núcleo efectúe una conversión de texto como éstas. Mucha gente ha arruinado sus datos por esta traducción. ¡Tenga cuidado!

Para sistemas de ficheros montados en modo binario, se dispone de una herramienta de conversión (fromdos/todos).

debug Activa la bandera debug. Se mostrarán una cadena de versión y una lista de parámetros del sistema de ficheros (estos datos también se muestran si los parámetros resultan ser inconsistentes).

fat=12 / fat=16

Especifica una FAT de 12 ó de 16 bits. Esto sustituye a la rutina de detección automática de tipo de FAT. ¡Utilice esta opción con cuidado!

quiet Activa la bandera quiet. Intentos de hacer chown o chmod en ficheros no devuelven errores, aunque fallen. ¡Utilice esta opción con cuidado!

sys\_immutable, showexec, dots, nodots, dotsOK=[yes|no]

Varios intentos equivocados de forzar convenciones Unix o DOS sobre un sistema de ficheros FAT.

uid=valor y gid=valor

Establece el propietario y grupo de todos los ficheros. (Por omisión: el UID y GID del proceso en curso.)

umask=valor

Establece la umask (la máscara de bits de los permisos que no están presentes). El valor predeterminado es el de la umask del proceso en curso. El valor se da en octal.

case=lower / case=asis

Convierte todos los nombres de ficheros a minúsculas (lower) o los deja tal cual (asis). (Por omisión: case=lower.)

conv=binary / conv=text / conv=auto

Para conv=text, elimina algunos CRs aleatorios (en particular, todos los seguidos de NL) cuando lee un fichero. Para conv=auto, escoge más o menos aleatoriamente entre conv=binary y conv=text. Para conv=binary, simplemente lee lo que está en el fichero. Esto es el comportamiento predeterminado.

nocheck

No para abruptamente el montaje cuando ciertas comprobaciones de consistencia fallen.

Los nombres de fichero normales iso9660 aparecen en un formato 8.3 (esto es, con restricciones en la longitud de los nombres de fichero

como en DOS), y además todos los caracteres están en mayúsculas. Asimismo no hay nada previsto para dueños de ficheros, protecciones, número de enlaces, provisión para dispositivos de caracteres o bloques, etc.

Rock Ridge es una extensión a iso9660 que proporciona todas estas características típicas de Unix. Básicamente hay extensiones en cada registro de directorio que proporcionan toda la información adicional, y cuando Rock Ridge está en uso, el sistema de ficheros es indistinguible de un sistema de ficheros UNIX normal (de lectura exclusiva, por supuesto).

`norock` Desactiva el uso de las extensiones Rock Ridge, incluso si están disponibles. Cf. `map`.

`check=r[elaxed] / check=s[trict]`

Con `check=relaxed`, un nombre de fichero se convierte primero a minúsculas antes de que se efectúe la búsqueda. Esto probablemente sólo tiene sentido en conjunción con `norock` y `map=normal`. (Por omisión: `check=strict`.)

`uid=valor` y `gid=valor`

Da a todos los ficheros en el sistema de ficheros los IDs indicados de usuario o grupo, posiblemente sustituyendo la información encontrada en las extensiones Rock Ridge. (Por omisión: `uid=0,gid=0`.)

`map=n[ormal] / map=o[ff]`

Para volúmenes que no son Rock Ridge, la traducción de nombres "normal" convierte las letras mayúsculas a minúsculas ASCII, quita un `‘;1’` del final, y convierte `‘;’` a `‘.’`. Con `map=off` no se realiza ninguna traducción de nombre. Vea `norock`. (Por omisión: `map=normal`.)

`mode=valor`

Para volúmenes que no son Rock Ridge, da a todos los ficheros el modo (permisos) indicado. (El predeterminado: permiso de lectura

para todo el mundo.) Desde Linux 2.1.37 uno ya no necesita más especificar el modo en base diez. (La base ocho se indica mediante un 0 inicial.)

`unhide` Muestra también ficheros ocultos y asociados.

`block=[512|1024|2048]`

Establece el tamaño de bloque al valor indicado. (Por omisión: `block=1024`.)

`conv=a[uto] / conv=b[inary] / conv=m[text] / conv=t[ext]`

(Por omisión: `conv=binary`.) Desde Linux 1.3.54 esta opción ya no tiene ningún efecto. (Y los valores distintos de `binary` solían ser muy peligrosos, llevando a menudo a corrupción silenciosa de datos.)

`cruft` Si el byte alto de la longitud de fichero contiene otra basura, active esta opción de montaje para no hacer caso de los bits de orden alto de la longitud de fichero. Esto implica que un fichero no puede ser mayor de 16 MB. La opción '`cruft`' se establece automáticamente si el CD-ROM entero tiene un tamaño raro (negativo, o de más de 800 MB). También se pone sola cuando se ven números de secuencia del volumen distintos de 0 ó 1.

Ninguna.

Vea las opciones de montaje para `fat`. Si el sistema de ficheros `msdos` detecta una inconsistencia, informa del error y pone el sistema de ficheros de lectura exclusiva. El sistema de ficheros se puede hacer de nuevo de lectura y escritura re-montándolo.

Justo como `nfs`, la implementación `ncp` espera un argumento binario (una `struct ncp_mount_data`) para la llamada al sistema `mount`. Este argumento se construye mediante `ncpmount(8)` y la versión actual de `mount` (2.6h) no conoce nada sobre `ncp`.

En vez de una cadena de opciones textual, analizada por el núcleo, el sistema de ficheros `nfs` espera un argumento binario de tipo `struct nfs_mount_data`. El propio programa `mount` analiza las opciones siguientes de la forma '`etiqueta=valor`', y las pone en la estructura mencionada: `rsize=n`, `wsiz=n`, `timeo=n`, `retrans=n`, `acregmin=n`, `acregmax=n`,

acdirmin=n, acdirmax=n, actimeo=n, retry=n, port=n, mountport=n, mounthost=nombre, mountprog=n, mountvers=n, nfsprog=n, nfsvers=n, namlen=n.

La opción addr=n se acepta pero no se tiene en cuenta. Asimismo se reconocen las siguientes opciones booleanas, posiblemente precedidas por no: bg, fg, soft, hard, intr, posix, cto, ac, tcp, udp, lock. Para más detalles, consulte nfs(5).

Entre las opciones especialmente útiles se incluyen

rsize=8192, wsize=8192

Esto hará que su conexión NFS sea mucho más rápida que con el tamaño de búfer predeterminado de 1024.

**hard** El programa que acceda a un fichero en un sistema de ficheros montado por NFS se colgará cuando el servidor falle. El proceso no puede interrumpirse ni matarse a menos que uno también especifique intr. Cuando el servidor NFS esté de nuevo disponible el programa continuará por donde estuviera como si nada. Esto es probablemente lo que uno quiere.

**soft** Esta opción permite al núcleo esperar un rato y rendirse después si el servidor NFS no responde durante un tiempo determinado. El tiempo se puede especificar con timeo=tiempo. Esta opción podría ser de utilidad si su servidor NFS no respondiera a veces, o si fuera rearrancado mientras algún proceso estuviera intentando obtener un fichero de él. Usualmente lo que hace es causar un montón de problemas.

**no lock** No emplear el mecanismo de candados. No arrancar el demonio lockd.

uid=valor y gid=valor

Estas opciones se reconocen, pero no tienen ningún efecto por lo que he podido ver.

Ninguna.

Justo como nfs, la implementación smb espera un argumento binario (una struct smb\_mount\_data) para la llamada al sistema mount. Este argumento se construye mediante smbmount(8) y la versión actual de mount (2.6h)



no sabe nada acerca de smb.

Ninguna.

Ninguna.

Vea las opciones de montaje para msdos. La opción dotsOK está explícitamente muerta en umsdos.

Antes de nada, las opciones de montaje para fat se reconocen todas. La opción dotsOK está explícitamente anulada en vfat. Además de todo eso, están

uni\_xlate

Traduce caracteres Unicode no manejados a secuencias de escape especiales. Esto le permite hacer copias de respaldo y restaurar ficheros cuyos nombres se han creado con caracteres Unicode. Sin esta opción, se emplea un '?' cuando no hay traducción posible. El carácter de escape es ':' porque es ilegal en el sistema de ficheros vfat. La secuencia de escape que se usa, donde u es el carácter Unicode, es: ':', (u & 0x3f), ((u>>6) & 0x3f), (u>>12).

posix Permite la coexistencia de dos ficheros cuyos nombres sólo se diferencien en la caja tipográfica.

nonumtail

Trata primero de construir un nombre corto sin número de secuencia, antes de intentar nombre~número.ext.

Ninguna.

Ninguna. Aunque no hay nada malo en xiafs, no se usa mucho, y no está mantenido ya. Probablemente uno no debería usarlo. Desde la versión de Linux 2.1.21, xiafs ya no forma parte de los fuentes del núcleo.

Un posible tipo más es un montaje a través del dispositivo de bucle.

Por ejemplo, la orden

```
mount /tmp/fdimage /mnt -t msdos -o loop=/dev/loop3,blocksize=1024
```

establecerá que el dispositivo de bucle /dev/loop3 corresponderá al fichero /tmp/fdimage, y entonces monta este dispositivo en /mnt. Este tipo de montaje conoce tres opciones, de nombres loop, offset y encryption, que son realmente opciones de losetup(8). Si no se menciona

explícitamente ningún dispositivo de bucle (sino que simplemente se da una opción ‘-o loop’), entonces mount intentará encontrar algún dispositivo de bucle en desuso y lo utilizará.

## FICHEROS

/etc/fstab tabla de sistemas de ficheros

/etc/mstab tabla de sistemas de ficheros montados

/etc/mstab~ fichero candado

/etc/mstab.tmp fichero temporal

mount(2), umount(2), fstab(5), umount(8), swapon(8), nfs(5), mountd(8), nfsd(8), mke2fs(8), tune2fs(8), losetup(8)

## FALLOS

Es posible que un sistema de ficheros corrupto cause una parada desastrosa.

Algunos sistemas de ficheros de Linux no admiten -o sync (el ext2fs sí soporta actualizaciones síncronas (al estilo BSD) cuando se monta con la opción sync).

La opción -o remount puede no ser capaz de cambiar parámetros de montaje (todos los parámetros específicos de ext2fs, excepto sb, se pueden cambiar con un remount, por ejemplo, pero no se pueden cambiar gid o umask para el fatfs).

## HISTORIA

Una orden mount apareció en la Versión 6 del UNIX de AT&T.