# Si usted es nuevo en Linux/Unix

## Introducción

Uno de los objetivos de Gnome es hacer su sistema fácil de usar, sin que necesite aprender los detalles técnicos sobre el sistema operativo. No obstante, hay algunas nociones básicas de Unix con las que debe familiarizarse, incluso cuando utilice la interfaz gráfica amigable suministrada por Gnome. Para los nuevos usuarios, estos comandos se engloban en este documento. Si usted necesita más información acerca de UNIX, debería leer la documentación que suminisitrada con su sistema. Existen también numerosos libros y guías disponibles en Internet para todas las versiones de UNIX.

La siguiente guía se aplica a todas las versiones de UNIX y todos los sistemas operativos parecidos a UNIX , incluidos tanto sistmas comerciales como Solaris y sistemas operativos de código abierto como BSD y Linux. Parte del material de esta guía está basado en la guía *Instalación de Linux e primeros pasos*, escrita por Matt Welsh, Phil Hughes, David Bandel, Boris Beletsky, Sean Dreilinger, Robert Kiesling, Evan Liebovitch, y Henry Pierce. La guía está disponible para descargar o consultar directamente en la dirección URL Proyecto de Documentación Linux[4] o del Grupo de Escritores de Código Abierto[5]. [6].

## Usuarios

UNIX es un sistema operativo multiusuario: fue designado para permitir a varios usuarios trabajar en la misma computadora, ya sea simultáneamente (utilizando varios terminales o en conexiones de red) o en turnos. En sistemas UNIX, para identificarse usted en el sistema, debe conectarse, lo que conlleva introducir su *nombre de acceso* (el nombre que el sistema utiliza para identificarle) y su *contrasena*, que es su clave personal para introducirse en su cuenta personal. Porque sólo usted conoce su clave de acceso, nadie más puede introducirse en el sistema con su nombre de acceso. Normalmente se escoge el nombre, el apellido o alguna variación de éstos como su nombre de acceso, por lo que si su nombre es: Sasha Beilinson, su nombre de acceso podría ser: Sasha.

Cada usuario tiene un espacio separado para guardar sus documentos (llamado *directorio personal*). UNIX posee un sistema de permisos (ver la sección de nombre *Permisos*), por lo cual en un Sistema UNIX apropiadamente configurado, un usuario no puede cambiar archivos pertenecientes a otro usuario o al sistema. Esto también permite a cada usuario, configurar varios aspectos del sistema — para sí mismo, sin afectar a otros.

En cada sistema UNIX, hay un usuario especial, llamado *administrador del sistema*, con el nombre de acceso root. Este usuario tiene *completo* control sobre el sistema — incluyendo acceso total a todos los archivos de sistema y de los usuarios. Tiene autoridad para cambiar las claves de acceso de los usuarios existentes y anadir nuevos usuarios, instalar y desinstalar software, además de otras cosas. El administrador del Sistema, es normalmente la persona responsable del correcto funcionamiento del sistema, por lo que si presenta algún problema, debe acudir a él.

**IMPORTANTE:** Incluso si usted es el único usuario en su ordenador (por ejemplo, si es ordenador personal), por lo que usted también es el administrador del sistema, es importante que cree una cuenta de usuario y la utilice para su trabajo diario, accediendo como administrador ("root") sólo cuando sea necesario para el mantenimiento del sistema. Dado que "root" puede hacer cualquier cosa, es fácil cometer errores que tengan consecuencias catastróficas. Imagine el usuario "root" como un sombrero mágico que le da a usted mucho poder, con el que puede, con un movimiento de manos, crear o destruir ciudades enteras. Ya que es fácil mover las manos de forma destructiva, no es una buena idea utilizar el sombrero mágico cuando no es necesario, a pesar de lo maravilloso que se siente.

## Archivo y nombres de archivos

En la mayoría de los sistemas operativos (incluyendo UNIX), existe el concepto de *archivo*, que es sólo un conjunto de información con un nombre (llamado *nombre del archivo*. Ejemplos de archivos, podrían ser su examen final de historia, un mensaje de correo electrónico, o un programa que pueda ser ejecutado. Esencialmente, cualquier cosa guardada en disco es guardado en un archivo individual.

### Nombres de archivos

Los archivos son identificados por sus nombres. Por ejemplo, el archivo que contiene una conferencia hablada, puede ser guardada bajo el nombre de archivo `talk.txt`. No existe un formato estándar para los nombres de los archivos, como existe en MS-DOS y otros sistemas operativos; en general, el nombre de un archivo puede contener cualquier carácter (excepto el carácter – vea la explicación de los caminos de nombres debajo) y es limitado ,en su extensión, a 256 caracteres.

> **IMPORTANTE:** A diferencia de MS-DOS, los nombres de archivo en UNIX son sensibles a mayúsculas o minúsculas: `midocumento.txt` y `MiDocumento.txt`, son considerados como dos archivos diferentes.

También debe conocer algunas convenciones en UNIX, que si bien no son obligatorias, normalmente es una buena idea seguirlas.

- Es costumbre utilizar el formato `nombrearchivo.extensión`, para nombres de archivo, en el que la extensión indica el tipo de archivo; por ejemplo, la extensión `txt` es normalmente utilizada para archivos de texto simple; en tanto que la extensión jpeg es utilizada para gráficos en formato JPEG, y así. En particular, la aplicación `Gestor de Archivos de Gnome` (`Nautilus`) utiliza extensiones para determinar el tipo de archivo. Usted puede ver o cambiar todas las extensiones de archivo reconocidas por `Gnome`, escogiendo la opción **tipos MIME** de la sección **Manejadores de documentos** en el `Centro de control de GNOME`. Observe que la convención estándar en UNIX es que los *ejecutables* no tienen extensiones.
- Los archivos y directorios cuyos nombres comienzan con un punto (.), son normalmente, *archivos de configuración*, esto significa que estos archivos contienen propiedades y preferencias para varias aplicaciones. Por ejemplo, Gnome guarda todos sus configuraciones en varios archivos en los directorios `.gnome` y `.gnome-desktop` en el directorio personal del usuario. Como la mayor parte del tiempo usted no necesita editar estos documentos manualmente, ni siquiera conocer su preciso nombre o localización, `Nautilus` no suele mostrar estos archivos. Usted puede cambiar la configuración como se indica en el manual de Nautilus[8].
- Los archivos cuyos nombres terminan con (~), por lo general son archivos de soporte (copias de seguridad) creados por varias aplicaciones. Por ejemplo, cuando usted edita un archivo `miarchivo.txt` con `emacs`, se guarda la versión previa en el archivo `miarchivo.txt~`.

### Caracteres comodín

Cuando usted entra órdenes desde la línea de comandos, puede utilizar los llamados *Caracteres comodín*, en lugar de un nombre de archivo exacto. El carácter comodín más común es *, que corresponde a cualquier secuencia de símbolos (incluyendo una línea vacía). Por ejemplo, la orden **ls *.txt** va a listar todos los archivos con la extensión `txt`, y la orden **rm capítulo*** va a borrar todos los archivos cuyos nombres comiencen con `capítulo` (**ls** y **rm** son órdenes UNIX para listar o borrar archivos). Otro carácter comodín es ?, que corresponde a cualquier símbolo individual: por ejemplo: **rm capítulo?.txt** borrará todos los archivos `capítulo1.txt, capítulo2.txt,` pero no el `capítulo10.txt`

Muchos de los nuevos usuarios de GNOME prefiere utilizar el `Gestor de Archivos GNOME` para realizar las operaciones con los archivos, antes que hacerlo desde la línea de comandos. Los caracteres comodín también son útiles para `Nautilus` en los diálogos de selección de archivos y de visualización de filtros.

### Uso de espacios, comas, etc. en los nombres de ficheros

Como se menciono anteriormente, un nombre de fichero puede contener no solo letras y números, sino también espacios, comas, etc. — cualquier carácter destino de la barra (/). Sin embargo, si esta usando comandos tecleados en la linea de comandos, debe tener cuidado cuando use estos ficheros. Para evitar problemas, es recomendable que ponga entre comillas (’) los nombres de ficheros con cualquier otra cosa que no sea letras, números, y puntos: para borrar `Mi Fichero`, debe teclear **rm ’Mi Fichero’** en vez de **rm Mi Fichero**.

Por supuesto, si usa exclusivamente herramientas gráficas como el gestor de ficheros de Gnome, no deberá preocuparse de estas cosas: para borrar el fichero `Mi Fichero`, simplemente arrastrelo a la papelera.

## Directorios y rutas

### Estructura de directorios

Ahora, discutiremos el concepto de directorios. Un *directorio* es una colección de archivos. Se puede pensar como una “carpeta” que contiene muchos documentos diferentes. A los directorios se les da nombres, por los que pueden ser identificados. Más aún, los directorios se mantienen en una estructura como de árbol, es decir, el directorio puede contener otros directorios. El directorio de más nivel es llamado el “directorio raíz” y denotado por `/`; que contiene los archivos de su sistema.

#### Rutas

Una *ruta* ("path") es realmente el “nombre completo” del fichero; contiene no solo el nombre del fichero, sino tambien su situación.Usted puede referirse a un archivo por su ruta, que se hace del nombre del documento, precedido por el nombre del directorio que contiene ese documento. Este, a su vez, es precedido por el nombre del directorio que contiene *este directorio* y así. Una ruta típica puede ser así: `/home/sasha/talk.txt` que se refiere al archivo `talk.txt` en el directorio `sasha`, el cual a su vez es un subdirectorio de `/home`.

Como puede ver, el directorio y el nombre del archivo están separados por una sola barra (/). Por esta razón los nombres de los archivos no pueden contener en sí mismos el carácter /. Los usuarios de MS-DOS encontrarán familiar esta convención, a pesar de que en el mundo del MS-DOS se utiliza la barra invertida (\). El directorio que contiene un subdirectorio dado, es conocido como el *directorio padre*. Aquí el directorio `home` es el padre del directorio `sasha`.

Cada usuario tiene un directorio personal ("home"), el cual es el directorio aparte que utiliza ese usuario para guardar sus archivos. Normalmente, los directorios personales de los usuarios están contenidos bajo `/home`, y son nombrados por el usuario que posee ese directorio, por lo que el directorio personal del usuario sasha sería `/home/sasha`.

### Nombres de directorios relativos

En cualquier momento, las órdenes que usted introduce son asumidas como *relativas* al directorio actual de trabajo. Usted puede pensar que su directorio de trabajo es el directorio en el que está actualmente "localizado". Cuando usted se conecta por primera vez, su directorio de trabajo es su directorio personal — para el usuario sasha, esto sería `/casa/sasha`. Cuando quiera referirse a un archivo lo puede hacer en relación con su actual directorio de trabajo, en lugar de especificar el nombre completo de la ruta del archivo.

Por ejemplo, si su directorio actual es `/home/sasha`, y tiene ahí un archivo llamado `talk.txt`, puede referirse a éste por el nombre del archivo: una orden como **emacs talk.txt** ejecutada desde el directorio `/home/sasha` es equivalente a **emacs /home/sasha/talk.txt** (`emacs` es un editor extremadamente poderoso para documentos de texto; los nuevos usuarios pueden preferir algo más simple, tal como `gnotepad`, pero para un usuario avanzado, `emacs` es indispensable).

Similarmente, si en `/home/sasha` tiene un subdirectorio llamado `textos` y, en ese subdirectorio un archivo llamado `teoria_campo.txt`, usted puede referirse a éste como `textos/teoria_campo.txt`.

Si usted comienza un nombre de un archivo (como `textos/teoria_campo.txt`) con otro carácter que no sea /, usted se está refiriendo al archivo en términos relativos a su actual directorio de trabajo. Esto es conocido como una ruta relativa. Por otro lado, si usted comienza el nombre del archivo con un /, el sistema interpreta esto como una ruta completa — esto es, una ruta que incluye la ruta completa al archivo, comenzando por el directorio raíz, /.El uso de una ruta completa es conocido como una *ruta absoluta*.

### Convenciones de ruta

Aquí hay algunas convenciones estándar que puede utilizar en las rutas:

`~/` — directorio personal del usuario

`./` — directorio actual de trabajo

`../` — directorio padre del directorio actual

Por ejemplo, si el directorio actual del usuario sasha es `/home/sasha/papers`, puede referirse al archivo `/home/sasha/talk.txt` como `~/talk.txt` o como `../talk.txt`.

## Permisos

Cada archivo en su sistema tiene un *dueno* — uno de los usuarios (normalmente el que ha creado este archivo), y un sistema de permisos que regula el acceso a éste archivo.

Para archivos ordinarios, existen 3 tipos de permisos de acceso: leer, escribir y ejecutar ("Read", "Write", "eXecute") ( el último sólo tiene sentido para archivos ejecutables). Estos permisos pueden ser establecidos independientemente para 3 categorías de usuarios: el dueno del archivo, los usuarios en el grupo que posee el archivo y todos los demás. Las discusiones de grupos de usuarios van más allá del alcance de este documento; las otras dos categorías se explican por sí mismas. Por tanto si los permisos en un archivo `/home/sasha/talk.txt` están establecer para leer y escribir por el usuario sasha, quien es el dueno del documento, y ser leído solo por todos los demás, sólo sasha podrá modificar este archivo.

Todos los nuevos archivos creados llevan algunos permisos estándar, por lo general leer/escribir para el usuario creador y leer sólo para todos los demás. Usted puede ver los permisos utilizando el Gestor de Archivos de GNOME, apretando el botón derecho del ratón en el archivo, y escogiendo **Propiedades** en el menú desplegable, y entonces la pestana *Permisos*. Utilizando este diálogo, puede también cambiar los permisos — sólo presione en el cuadrado que representa el permiso para modificar su estado. Por supuesto, sólo el dueno del archivo o el administrador del sistema puede cambiar los permisos de un archivo. Los usuarios avanzados también pueden cambiar los permisos de los archivos cuando se establecen en la creación de los mismos — vea las páginas del manual para su entorno de líneas de comandos, "shell"(normalmente **bash**, **csh** o **tsch**) y consulte la orden **umask**.

Un archivo también puede tener propiedades especiales de permiso como UID, GID y bit “sticky”. Estos permisos son sólo para usuarios avanzados — no los cambie a menos que usted sepa lo que está haciendo. (Si usted es curioso: estos permisos son típicamente utilizados en archivos ejecutables para permitir al usuario ejecutar *algunas* órdenes para leer o modificar archivos para los cuales el propio usuario no tiene acceso.)

Al igual que los archivos, los directorios también tienen permisos especiales. Otra vez, existen 3 posibles permisos: leer, escribir y ejecutar (“Read”,“Write” y “eXecute”). No obstante, tienen diferente significado: el llamado permiso de “leer” para un directorio, significa permiso para listar el contenido del directorio o buscar un archivo; “escribir” significa permiso para crear y eliminar archivos en el directorio, y “ejecutar” significa permiso para acceder a los archivos en el directorio.

Note que los permisos otorgados a un archivo dependen de los permisos del directorio en el cual el documento está localizado: para ser capaz de leer un archivo, el usuario necesita tener el permiso de leer para el propio archivo y el permiso “ejecutar” para el directorio que lo contiene. Por tanto, si el usuario sasha no quiere que nadie más vea sus archivos, puede lograr esto eliminando los permisos de ejecución de su directorio personal para todos los demás usuarios. De esta manera, sólo él (y, por supuesto, el administrador “root”) podrán leer cualquiera de sus archivos, sin importar cuales sean los permisos individuales de los archivos.

Una explicación detallada del sistema de permisos puede ser encontrada, por ejemplo, en las páginas info[9] del paquete de *Utilidades de Archivos* GNU.

## Enlaces simbólicos

Además de los archivos regulares, UNIX tiene también archivos especiales llamados *enlaces simbólicos* ("symbolic linksº *symlinks*, para acortar). Estos archivos no contienen datos; en su lugar solo son ªpuntadoresº ªtajosª otros archivos. Por ejemplo, sasha puede tener un symlink llamado `ft.txt` que apunta al documento `pruebas/teoriacampo.txt`; de esta manera cuando un programa trata de acceder al archivo `ft.txt`, el archivo `pruebas/teoriacampo.txt` será abierto en su lugar. Como puede ver por este ejemplo, el symlink y el archivo destino pueden tener nombres diferentes y ser localizados en directorios diferentes.

Note que eliminar, mover o renombrar un documento symlink no tiene efecto en el archivo destino: si sasha trata de eliminar el archivo `ft.txt`, es el symlink lo que se eliminará, y el archivo `pruebas/teoriacampo.txt` seguirá inalterado. También los permisos del symlink no tienen significado alguno, son los permisos del archivo destino los que determinan si el usuario tiene acceso a éste.

Los symlinks también pueden apuntar a directorios. Por ejemplo, en el servidor de FTP de GNOME (ftp.gnome.org), existe un archivo `/pub/GNOME/stable/releases/october-gnome`, que en el momento en que se escribe este manual, es un enlace simbólico al directorio `/pub/GNOME/stable/releases/gnome-1.0.53.` — como podra imaginar, “October Gnome” es simplemente otro nombre de la versión 1.0.53 de Gnome.

## Montar y desmontar dispositivos

Como hemos mencionado anteriormente, los directorios en un entorno UNIX están organizados en un árbol, cuyo directorio raíz es `/`. A diferencia de otros sistemas operativos como MS-DOS, no hay nombres especiales para los archivos en la unidad de disquete o en el CD-ROM: *todos* los archivos accesibles por su sistema deben aparecen en él árbol de directorios principal que empieza por `/`.

Por tanto, antes de que usted tenga acceso a los archivos en un disquete o CD-ROM, usted debe dar a su sistema una orden para incorporar los contenidos del disquete en el árbol directorio principal, al cual se le refiere como *montaje* del disquete. Puede verlo como el equivalente software a la conexion del dispositivo a su sistema. Típicamente, los contenidos del CD-ROM aparecen bajo el nombre `mnt/cdrom`; los del disquete bajo `/mnt/floppy` (éstos son los llamados *puntos de montaje* y son definidos en el archivo especial de configuración, `/etc/fstab`). El acceso a una unidad, de esta manera, no significa que el sistema copiará todos los archivos del CD al directorio `/mnt/cdrom`. En su lugar, esto significa que el directorio `/mnt/cdrom` *representa* al CD-ROM: Cuando un programa trata de tener acceso, digamos a un archivo llamado `/mnt/cdrom/index.html`, el sistema buscará el archivo `index.html` en el CD-ROM.

Por tanto, en pocas palabras: antes de que usted pueda utilizar archivos en una unidad, usted debe "montarlo". Similarmente *antes de sacar el disco del lector, usted debe desmontarlo.*

Cuando utilice GNOME, usualmente no tiene que preocuparse por montar y desmontar: GNOME busca el archivo con la configuración apropiada y localiza los iconos para todas las unidades en su escritorio. Al hacer doble clic en cualquiera de estos iconos, automáticamente se monta la unidad correspondiente (si no estaba montado ya) y ejecuta el gestor de archivos en el directorio apropiado. Similarmente, si usted hace doble clic en el icono unidad y escoge la orden **Sacar disco** del menú desplegable, GNOME desmonta automáticamente antes de sacarlo. Usted puede también montar/desmontar una unidad presionando con el botón derecho del ratón en el icono de su escritorio y escogiendo **Montar unidad** o **Desmontar unidad** del menú desplegable, o utilizando el aplique de montaje de discos.

Note que usted no puede desmontar una unidad si está siendo utilizado por algún programa; por ejemplo, si usted tiene abierta una ventana terminal en el directorio de la unidad que usted está tratando de desmontar. Entonces, recibe el mensaje de error "Controlador ocupado"mientras intenta desmontar la unidad, asegúrese de que ninguna de sus aplicaciones abiertas esté teniendo acceso a un archivo o directorio en esta unidad.

No obstante GNOME no puede impedir que usted saque el disco manualmente de la unidad —, en este caso es su responsabilidad el desmontar la unidad antes de hacerlo. Para unidades de CD o Zip, el sistema bloquea el botón de sacado de la unidad mientras la unidad esté montado, para los disquetes, esto es técnicamente imposible.

> **IMPORTANTE:** Si usted saca un disquete utilizando un botón de sacado de la unidad sin desmontarlo primero, usted puede perder sus datos!

Algunos sistemas tienen programas especiales como `supermount` o `magicdev`, que montan automáticamente una unidad cuando se inserta un disco y desmonta la unidad si ésta no se ha utilizado por un período de tiempo específico. En este caso, usted nunca deberá preocuparse de montar/desmontar unidades usted mismo; no necesita por tanto ni leer esta sección.

El permitir a los usuarios el montar y desmontar unidades conlleva algunos riesgos de seguridad, muchos sistemas multiusuarios se configuran de modo que sólo el usuario administrador "root"puede montar y desmontar una unidad. Ésta es la causa mas probable de los errores al intentar montar un dispositivo. En este caso, plantee este problema al administrador de su sistema.

Si la computadora es su estación de trabajo personal o el ordenador personal de casa y no le preocupa la seguridad, usted puede dar permiso de montar unidades a usuarios ordinarios. La manera más fácil de permitir ello es el uso de la aplicación *`linuxconf`* (que sólo puede ser ejecutada por el usuario administrador "root"). Sólo seleccione la unidad a la que quiere acceder en la sección *Unidades de acceso local* de la pestana *Opciones* de la opción *Montable por usuarios*. Su unidad será ahora montable por los usuarios.

Si *`linuxconf`* no está disponible, usted debe editar el archivo `/etc/fstab` para incluir acceso a usuarios. Esto se hace anadiendo el atributo del üsuarioª la unidad. Por ejemplo:

Si su archivo `fstab` contiene una línea como ésta:

```
/dev/cdrom /mnt/cdrom iso9660 exec,dev,ro,noauto 0 0
```

anada la palabra üsuarioª la cuarta columna:

```
/dev/cdrom /mnt/cdrom iso9660 user,exec,dev,ro,noauto 0 0
```

## Controladores y dispositivos

En Unix, la palabra "dispositivo" se usa para todos los dispositivos periféricos conectados al ordenador; esto incluye los discos duros, disqueteras y CDROM, tarjetas de vídeo y audio, puertos serie y paralelo, y demás cosas. Cada dispositivo tiene un nombre, como `/dev/hda`. Los nombres mas comunes se listan a continuación (para Linux; otras variedades de Unix pueden usar nombres de dispositivos ligeramente diferentes).

- `/dev/hd*` (donde *=a,b,c, . . . ): son los dispositivos IDE, como discos duros, lectores CDROM y dispositivos ZIP. `/dev/hda` representa el dispositivo maestro del primer controlador IDE (usualmente el primer disco duro, `C:` en windows), `/dev/hdb` es el dispositivo esclavo del primer controlador (puede ser un segundo disco duro o un CDROM), y así sucesivamente. Véase también la nota mas abajo acerca de los dispositivos ZIP.

- `/dev/sd*` (donde *=a,b,c, . . . ): son los dispositivos SCSI, normalmente discos duros.

> **NOTA:** Si los acrónimos IDE y SCSI son nuevos para usted, he aquí una breve explicación: hay dos tipos de interfaces para discos duros y otros dispositivos similares: IDE ( y sus parientes como EIDE, ATAPI, etc.) y SCSI. SCSI proporciona mejores prestaciones, pero es mas caro, así que se usa solo en los servidores. Si no esta seguro del tipo de dispositivos que tiene, lo mas probable es que sean IDE.

- `/dev/fd*` (donde *=0,1, etc) son las disqueteras; `/dev/fd0` es la primera disquetera (se corresponde con `A:` en windows), `/dev/fd1` es la segunda (`B:`), etc.
- `/dev/lp*` (donde *=0,1, etc) son los puertos paralelos; generalmente, estos puertos se usan para conectar impresoras a su ordenador. `/dev/lp0` se corresponde con `LPT1` en Windows, `/dev/lp1` con `LPT2`, etc.
- `/dev/ttyS*` (donde *=0,1, etc) son los puertos serie; estos puertos se usan generalmente un ratón o un módem. `/dev/ttyS0` se corresponde con `COM1` en Windows, `/dev/ttyS1` con `COM2`, etc.
- `/dev/audio` y `/dev/dsp` — estos dos dispositivos se usan para la tarjeta de audio (no hay equivalentes, ya que se usan para diferentes tipos de ficheros de audio).

Además, es una practica común disponer de enlaces simbólicos, `/dev/floppy`, `/dev/modem` y `/dev/cdrom`, apuntando a los nombres reales de los dispositivos que correspondan a la disquetera, módem, y CDROM, respectivamente.

En raras ocasiones necesitara usar los nombres de los dispositivos. En particular, si necesita acceder a un fichero en dispositivo, no usara el nombre del dispositivo (como `/dev/fd0`); en su lugar, primero montara el dispositivo de forma vera que su contenido como un subdirectorio (por ejemplo, `/mnt/floppy`) en el árbol de directorios principal, y usara entonces ese directorio para acceder a los ficheros; Véase la sección de nombre *Montar y desmontar dispositivos* para mas información. Prácticamente el único momento en que necesitara usar los nombres de los dispositivos sera cuando este configurando algún nuevo programa. Por ejemplo, un programa de fax puede pedirle el nombre del dispositivo que representa su módem ( en cuyo caso puede o bien darle el nombre real del dispositivo, como `/dev/ttyS1`, o bien usar simplemente un enlace simbólico `/dev/modem`).

Y solo para satisfacer su curiosidad: también hay un dispositivo llamado `/dev/null` que actúa como un "agujero negro": puede enviar allí cualquier información, y jamas volverá. Así que si no quiere que le molesten los mensajes de error, redirijalos a `/dev/null` -:).

## Particiones

Observe que se puede subdividir un disco duro (o un dispositivo similar) en partes que se comportan a todos los efectos como discos independientes, aunque residan físicamente en el mismo disco. Estas partes se llaman "particiones" (en Windows, se conocen como "discos lógicos"). Por ejemplo, puede dividir el disco duro en varias particiones, e instalar diferentes sistemas operativos en diferentes particiones; puede formatear de nuevo cada partición independientemente del resto. La división del disco duro se realiza normalmente durante la instalación del sistema operativo; para mas información remitase a su guia de instalación.

Si ha dividido su disco duro, entonces cada partición se considera como un dispositivo separado. Por ejemplo, si su disco duro es `/dev/hda`, entonces nos referiremos a la primera partición de este dispositivo como `/dev/hda1`, la segunda como `/dev/hda2`, y así sucesivamente.

**Particionar discos ZIP**

Por razones que desconocemos, los disco ZIP preformateados a la venta en las tiendas o formateados usando las herramientas ZIP de Iomega en Windows están particionados de una forma curiosa: tienen solo una partición (de tipo windows, por supuesto). De este modo, si su dispositivo ZIP es `/dev/hdc`, el nombre correcto del dispositivo que debería usar para estos discos es `/dev/hdc4`.

## Interfaz gráfica de usuario: el sistema de ventanas X, el gestor de ventanas, y los entornos de escritorios.

Unix es un sistema modular: se compone de muchos componentes de forma que el usuario (o el administrador de sistemas) pueda escoger aquellos componentes que necesite. En particular, hay varias capas de software responsables de la interfaz gráfica de usuario. Estas capas son: el sistema de ventanas X, el gestor de ventanas, y el entorno de escritorio.

El *sistema de ventanas X* (También conocido como X, o X11) es el componente de los sistemas Unix responsable de virtualmente todas las operaciones gráficas básicas — en particular, de dibujar los iconos, los fondos, y las ventanas en las que se ejecutan las aplicaciones. Sin las X, solo tendríamos la linea de comandos. X11 determina la resolución de la pantalla y la profundidad de color, mueve el cursor del ratón alrededor de la pantalla, etc. Sirve de base para otros componentes de la interfaz gráfica de usuario como los gestores de ventanas y los entornos de escritorio.

Los *gestores de ventanas* amplian las capacidades del sistema de ventanas X poniendo bordes y botones alrededor de la ventanas, lo que permite al usuario moverlas, cerrarlas, ocultarlas o cambiar su tamano. X11 se usa casi siempre en combinación con un gestor de ventanas, ya que seria prácticamente inutilizable sin el. Hay muchos gestores de ventanas disponibles para X11; los mas populares son `fvwm`, `mwm`, `kwm` (usado por KDE), `Enlightenment`, y `Sawfish`.

Finalmente, un *entorno de escritorio* va un paso mas allá del gestor de ventanas, anadiendo un gestor gráfico de ficheros desde el cual podrá arrastrar y soltar elementos sobre su escritorio, un panel que puede usarse para lanzar las aplicaciones usadas frecuentemente, y un conjunto de aplicaciones y utilidades. Hay varios entornos de escritorio disponibles para todas las versiones de Unix; Los mas populares son GNOME[10], KDE[11] y CDE[12] (pronto sustituido por GNOME).

casi todos los entornos de escritorio contienen un gestor de ventanas como parte integral; por ejemplo, KDE contiene su propio gestor de ficheros, `kwm` (es posible usar KDE con otro gestor de ventanas, pero poca gente lo hace). Gnome no dispone de su propio gestor de ventanas; le permite escoger cualquier gestor de ventanas que tenga ya en su sistema. Para hacer la vida mas fácil a los nuevos usuarios, se distribuye generalmente el gestor de ventanas `Sawfish` con Gnome y se usa por defecto; puede cambiar a otro gestor de ventanas usando el `Centro de control de GNOME`. Observe en cualquier caso que necesita un gestor de ventanas compatible con Gnome para usar algunas de las funcionalidades de Gnome, como el gestor de sesiones, el aplique barra de tareas, etc.

## Notas

1. gnome-help:fdl
2. http://www.fsf.org
3. gnome-help:fdl
4. http://www.linuxdoc.org
5. http://www.oswg.org
6. Una versión en castellano de esta guia se encuentra en las paginas de
   7. http://LuCAS.hispalinux.es/Manuales-LuCAS/LIPP/lipp-1.1-html-2/

.

7. http://LuCAS.hispalinux.es/Manuales-LuCAS/LIPP/lipp-1.1-html-2/
8. gnome-help:nautilus
9. info:fileutils
10. http://www.gnome.org
11. http://www.kde.org
12. http://www.sun.com/solaris/cde/