

ADMINISTRACIÓN DEL SISTEMA LINUX



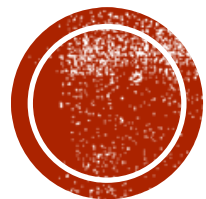
OBJETIVOS



ÍNDICE DE CONTENIDOS

1. El sistema de archivos.
2. El Administrador de archivos.
3. Comandos de manipulación.
4. Los usuarios.
5. Las copias de seguridad.
6. El monitor del sistema.





EL SISTEMA DE ARCHIVOS



EL SISTEMA DE ARCHIVOS

- Al igual que en otros SO, la información se almacena en ficheros y estos a su vez se pueden agrupar en directorios y subdirectorios formando una estructura en árbol similar a los sistemas MS-DOS/Windows.
- La gran diferencia respecto a estos sistemas es que en Linux (y Unix) existe un solo árbol de directorios y no uno para cada partición o disco del sistema.
- Existe un directorio que contiene el resto de directorios y archivos del sistema; se le llama **directorio raíz**.
- El **directorio home** es el directorio asignado a cada usuario del sistema Linux.
- Una característica importante en Linux es que **realiza distinción entre mayúsculas y minúsculas**. Tanto para nombres de carpetas, archivos, nombres de programas, comandos y cualquier elemento.



EL SISTEMA DE ARCHIVOS

- Las ubicaciones de los ficheros en el sistema siguen unas normas determinadas con el objeto de aumentar el nivel de organización.
- Algunos de los directorios más importantes son los siguientes:
 - /bin
 - /sbin
 - /home
 - /usr
 - /dev
 - /lib
 - /etc
 - /media



EL SISTEMA DE ARCHIVOS

- **/bin:** estos directorios contienen la mayoría de los ficheros y los comandos más comunes del sistema Linux.
- **/sbin:** están los archivos de configuración del sistema.
- **/home:** es un directorio donde se tienen los directorios personales de todos los miembros del sistema.
- **/usr:** contiene utilidades y programas generales de usuario. Este directorio contiene los programas de uso general para todos los usuarios.
- **/dev:** directorio que almacena ficheros de dispositivos. Estos ficheros no son realmente ficheros sino que es la forma en la que los sistemas Linux implementan los controladores de dispositivos.



EL SISTEMA DE ARCHIVOS

- **/lib:** contiene librerías y compiladores del sistemas. Contiene las bibliotecas necesarias para que se ejecuten los programas que tenemos en **/bin** y **/sbin** únicamente.
- **/etc:** este directorio contiene los ficheros de configuración de todo el sistema, con lo cual es de vital importancia. Normalmente los ficheros que contiene son ficheros de configuración de tipo texto sin formato. Además, suelen tener solo permisos de lectura para usuarios normales, es decir, solo el usuario con privilegios de administración los puede modificar.
- **/media:** contiene todas las unidades físicas que tenemos montadas: discos duros, unidades de CD/DVD, pen drive, etc.



EL SISTEMA DE ARCHIVOS

2. LOS FICHEROS DE DISPOSITIVO (/dev)

- Una de las características especiales de Linux es su forma de acceder o reconocer los dispositivos físicos para poder hacer uso de ellos.
- Para Linux, cualquier dispositivo físico depende directamente de los llamados manejadores de dispositivos, que se integran en la estructura de ficheros del sistema dentro del directorio */dev*.
- De esta forma, el acceso a los diferentes dispositivos físicos se hace de una forma homogénea; la entrada/salida se realiza siempre como si se accediera a ficheros.
- Para que un dispositivo pueda ser “manejado” a través de estos ficheros es necesario que haya un driver o controlador a bajo nivel instalado.
- Por lo tanto, todos los ficheros que contiene el directorio */dev* realmente no son ficheros de disco sino que son manejadores de dispositivos creados por los drivers.



EL SISTEMA DE ARCHIVOS

2. LOS FICHEROS DE DISPOSITIVO (/dev)

- Los nombres de estos “ficheros” siguen un estándar para identificarlos con los dispositivos físicos a los que están asociados.
- A continuación, se presenta una lista de los más comunes:
 - UNIDAD DE CD-ROM O DVD:
 - /dev/cdrom
 - /dev/dvd
 - Dispositivos conectados a un bus IDE:
 - /dev/hda
 - /dev/hda1 ...
 - Dispositivos conectados al bus USB
 - /dev/usb
 - Ratón:
 - /dev/psaux



EL SISTEMA DE ARCHIVOS

3. LOS PERMISOS

- Linux mantiene un sistema de permisos de acceso a los ficheros muy estricto, a fin de controlar qué es lo que se puede hacer con ellos, y quien lo puede hacer.
- Estos permisos se identifican con letras y son:
 - **r**: permiso de lectura de fichero.
 - **w**: permiso de escritura en el fichero.
 - **x**: permiso de ejecución del fichero.
 - **s**: permiso para cambiar el propietario del fichero.
- Al contrario que en Windows los programas ejecutables de Linux no están marcados por una determinada extensión (.exe) sino por un atributo, el permiso de ejecución x.
- Si se elimina este atributo a un programa, Linux no será capaz de ejecutarlo.

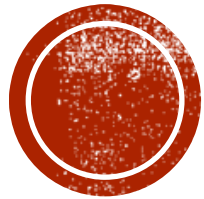


EL SISTEMA DE ARCHIVOS

3. LOS PERMISOS

- A su vez cada uno de estos permisos se aplica:
 - ***Al dueño del fichero (u).***
 - ***Al grupo de usuarios al que pertenece el dueño (g).***
 - ***Al resto de usuarios (a).***
- Así un fichero determinado puede tener permiso para ser leído, escrito y ejecutado por su dueño, leído y ejecutado por el grupo al que pertenece y no tener ningún tipo de acceso para los demás usuarios.
- Como se puede entender este tipo de mecanismo es especialmente útil cuando se trabaja en grupo en un determinado proyecto.





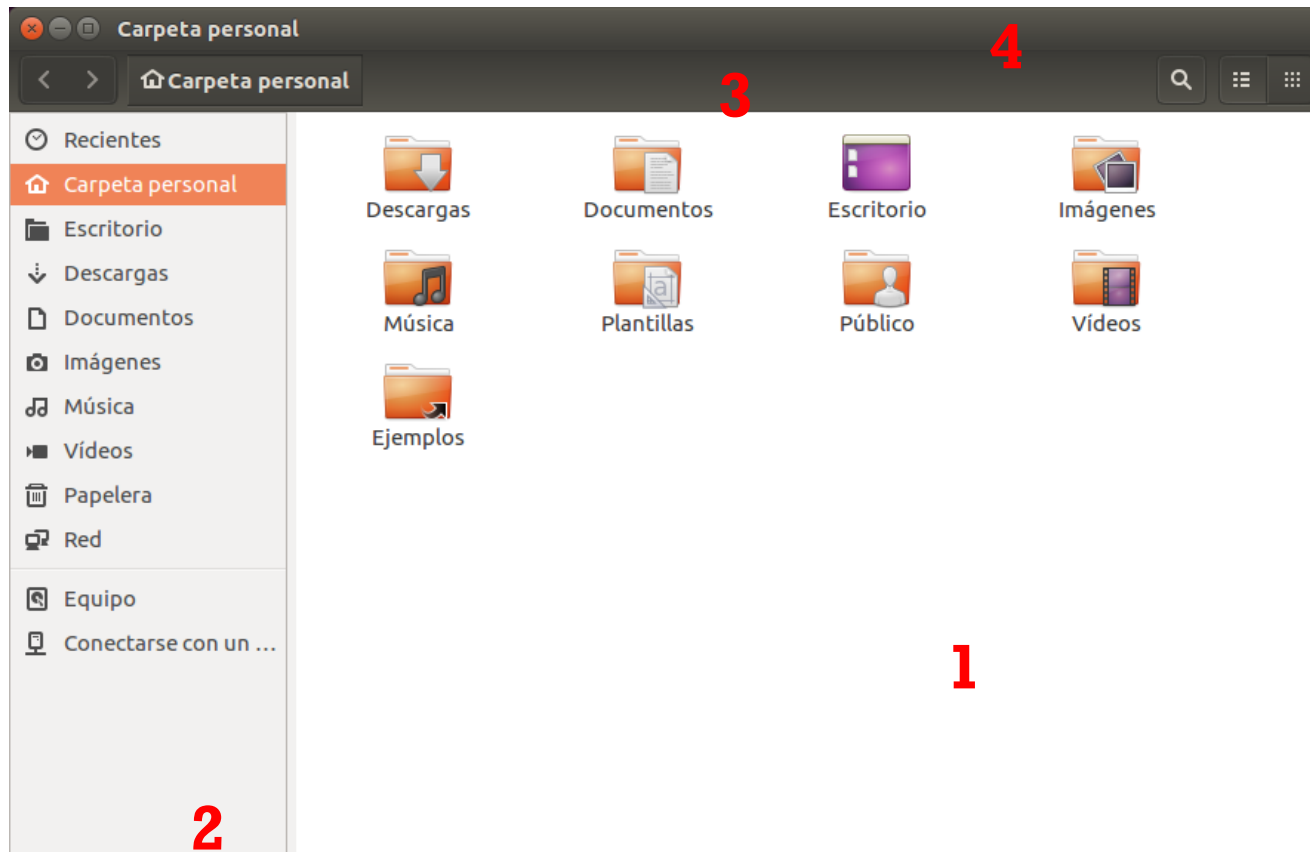
ADMINISTRADOR DE ARCHIVOS

ADMINISTRADOR DE ARCHIVOS

- La aplicación que permite organizar los archivos es **Nautilus**.
- Para acceder a ella, hay que ir al lanzador de Carpeta personal y aparecerá el administrador de archivos:



ADMINISTRADOR DE ARCHIVOS



ADMINISTRADOR DE ARCHIVOS

1. **Panel de visualización:** Es la zona central de la aplicación en la que se listan los contenidos de la carpeta.
2. **Panel lateral izquierdo:** Muestra una lista de accesos directos a los lugares básicos del sistema. Para mostrar u ocultar el panel se puede pulsar la tecla [F9] para hacerlo alternativamente.
3. **Barra de lugar:** Muestra cada nivel de profundidad en el árbol de directorios. Toma como punto de partida la carpeta personal del usuario ofreciendo la posibilidad de ascender hasta la raíz con la flecha de la izquierda. Además dispone de una serie de botones que ofrecen las opciones básicas de navegación por las carpetas: **atrás**, adelante y buscar.
4. **Barra de menú:** Nautilus cuenta con un menú principal, similar al de otras aplicaciones que se ejecutan en un entorno de ventanas, cuyas opciones son suficientes descriptivas para cualquier usuario. Para que se muestre hay que situar el ratón sobre dicha opción.



ADMINISTRADOR DE ARCHIVOS

- Es posible compartir una carpeta con otros equipos de la red, ya dispongan de Linux o de Windows.





COMANDOS DE MANIPULACIÓN



COMANDOS DE MANIPULACIÓN

- Para trabajar con el sistema de archivos en modo orden, se puede hacer en modo gráfico o en modo texto.
- Para trabajar en modo gráfico, pulsamos [Ctrl] + [Alt] + [T]
- En ella, el sistema presenta el **prompt** indicando que ya está preparado para aceptar los comandos que desee.



COMANDOS DE MANIPULACIÓN

- Los comandos básicos para trabajar con la estructura de **directorios** del sistema:
 - **pwd**
 - **cd [nombre_directorio]**
 - **ls [-opciones][nombre_directorio]**
 - **mkdir *directorio***
 - **rmdir *directorio***
- **Comandos para salir de la sesión: logout o exit**
- **Comandos de ayuda: man**



COMANDOS DE MANIPULACIÓN

- **pwd**

- Cuando se trabaja en modo orden hay que tener en cuenta siempre la ubicación donde se encuentra, ya que muchos comandos que hacen referencia a directorios utilizan el directorio actual si o se especifica otra cosa.
- Al directorio donde se encuentra se le llama directorio actual o directorio activo.
- El comando *pwd* muestra en pantalla la ruta completa del directorio actual o activo.
- En muchas ocasiones, el *prompt* del sistema se configura para mostrar el directorio activo pero sin no es así se puede utilizar el comando *pwd* para visualizarlo.



COMANDOS DE MANIPULACIÓN

- **cd [nombre_directorio]**

- Cambia el directorio actual por el especificado como parámetro. Utilizando sin parámetros el comando `cd` cambia al directorio *home* del usuario.

- Ejemplos:

`cd /usr/local/doc`
utilizando la ruta

Ejemplo del comando `cd`
absoluta.

`cd local/doc`
/usr, igual que
ruta relativa.

Suponiendo que se encuentra en
el anterior pero usando

`cd ..`
por encima del

Cambia al directorio de un nivel
actual.

`cd ../usuario1/apuntes`
una ruta

Ejemplo del comando `cd` utilizando
relativa.

`cd`
usuario.

Cambia al directorio *home* del



COMANDOS DE MANIPULACIÓN

- **ls [-opciones] [nombre_directorio]**

- Visualiza el contenido del directorio especificado como parámetro. Si no se indica ningún directorio se utiliza el directorio actual.
- Este comando tiene multitud de opciones para alterar el tipo de información mostrada.
- Ejemplos:

ls -l Lista extendida de archivos y directorios del directorio actual

ls -a Lista de todos los archivos y directorios, incluidos los ocultos, que Linux empiezan por el carácter punto “.”.

ls -R Lista de archivos y directorios y de todos los directorios dentro de él.



COMANDOS DE MANIPULACIÓN

- ***mkdir directorio***

- Crea un nuevo directorio.
- Si es necesario crea subdirectorios intermedios escribiendo ***-p***.
- Ejemplo: ***mkdir -p docs/linuxdocs/ejercicio/prueba***
- Se crean todos los subdirectorios que no existan (docs, linuxdocs, ejercicio y prueba)

- ***rmdir directorio***

- Borra un directorio vacío, si tiene contenido no lo borrará.



COMANDOS DE MANIPULACIÓN

- **Comandos para salir de la sesión: logout o exit**
 - Con *logout* el sistema finaliza la sesión y vuelve a mostrar la petición de *login* para una nueva sesión.
 - El comando *exit* es equivalente y se utiliza para el cierre del terminal en modo gráfico.
- **Comando de ayuda: man**
 - La forma de ejecutarlo es teclear *man* seguido del nombre del comando sobre el que queremos obtener ayuda. Por ejemplo *man ls*
 - La ayuda que ofrece el comando *man* se presenta paginada y formateada con algún programa formateador de textos que esté instalado en nuestro sistema.
 - Para salir de nuevo al *prompt* del sistema, pulsar la tecla [*q*].



COMANDOS DE MANIPULACIÓN

2. *COMANDOS DE MANIPULACIÓN DE ARCHIVOS*

- Los comandos básicos para trabajar con la estructura de **ficheros o archivos** del sistema:
 - **cat**
 - **cp**
 - **mv**
 - **rm**
 - **Touch**



COMANDOS DE MANIPULACIÓN

- **cat**
- Entre sus múltiples opciones, está la posibilidad de crear un archivo, imprimirlo por pantalla su contenido...
 - El siguiente comando creará un archivo de texto vacío, de nombre *ejemplo1*, y nos permitirá teclear el contenido que deseemos introducirle.
 - ***cat > ejemplo1***
 - La siguiente orden nos permitirá imprimir por pantalla todas las líneas de texto del archivo de nombre *ejemplo1*. El archivo indicado debe encontrarse en el directorio de trabajo actual.
 - ***cat ejemplo1***
- Para acabar de cerrar el editor y guardar hay que presionar ***Ctrl+D***.



COMANDOS DE MANIPULACIÓN

■ **cp file1 file2**

- Se utiliza para copiar ficheros.
- Crea una copia de file1 y le llama file2.
- Si file2 no existe, lo crea con los mismos atributos de file1.
- Si file2 ya existe, su contenido queda destruido y es sustituido por el de file1.
- El fichero file2 estará en el mismo directorio que file1.
- Para copiar un archivo de una carpeta a otra y que se quede en los dos lugares:

cp *direccionOrigen direccionDestino*



COMANDOS DE MANIPULACIÓN

■ **cp file1 file2**

- Se utiliza para copiar ficheros.
- Crea una copia de file1 y le llama file2.
- Si file2 no existe, lo crea con los mismos atributos de file1.
- Si file2 ya existe, su contenido queda destruido y es sustituido por el de file1.
- El fichero file2 estará en el mismo directorio que file1.
- Para copiar un archivo de una carpeta a otra y que se quede en los dos lugares:

cp *direccionOrigen direccionDestino*



COMANDOS DE MANIPULACIÓN

■ **mv file1 [file2] directorio**

- Permite mover un archivo hacia otro, o varios archivos hacia un directorio.
- Este permite a su vez renombrar archivos o directorios.
- Si el último argumento es un directorio, mueve el fichero anterior a ese directorio.
- Si son solo dos nombres de archivos, renombra el primero al segundo.
 - -i: pregunta antes de sobrescribir el destino si existiera.
 - -u: actualiza el destino solo si este es más reciente.



COMANDOS DE MANIPULACIÓN

- **rm [opciones] <archivos | directorios>**
 - Se utiliza para borrar archivos.
 - Algunas opciones
 - **-r**: borra recursivamente un directorio aunque contenga archivos y/o directorios.
 - **-f**: borra forzosamente en caso de que no se tenga permiso de escritura en forma directa.
 - **-i**: ejecuta el comando de forma interactiva.
- **touch**
 - Crea un archivo vacío, si el archivo existe actualiza la hora de modificación.
 - Puede crear varios archivos a la vez.
 - Si no se indica el directorio se crea en el que se está.
 - Para crear archivo `prueba1.txt` en `/home`, seria:
 - ***touch /home/prueba1.txt***





USUARIOS

- Como ya se ha afirmado Linux es un sistema multiusuario, lo cual permite que varios usuarios puedan conectarse y trabajar en él de forma simultánea.
- Las conexiones como ya se ha visto se pueden realizar a través de varias terminales locales o utilizando servicios de red como el Telnet y el SSH.
- Un usuario se caracteriza por su login el cual debe indicar para conectarse al sistema, además de su password o contraseña.
- Además puede poseer un conjunto de datos adicionales mencionados más adelante.
- El usuario con más privilegios en Linux es aquel cuyo **login** es **root**.
- Este es el único con derechos suficientes para crear o eliminar a otros usuarios, además de acceder a todo el sistema de ficheros sin ninguna restricción.



USUARIOS

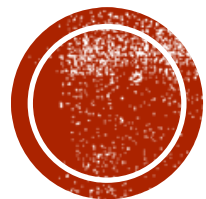
- En Linux además existen grupos de usuarios también administrados por root o por un usuario designado por este.
- Los grupos permiten otorgar los mismo privilegios a un conjunto de usuarios.
- Siempre que se añada un usuario al sistema se creará un grupo con su mismo nombre, llamado grupo primario.
- Durante la creación o posteriormente, se podrá incorporar el usuario a otros grupos secundarios.
- Tanto los usuarios como los grupos se identifican por el sistema a través de un identificador (ID) numérico.
- El usuario root siempre tiene el ID cero.
- Cada usuario cuando se conecta al sistema posee un identificador de usuario asociado (uid) y un identificador de grupo (gid).



USUARIOS

- Para crear un usuario se usa el comando **useradd**, aunque es mucho mejor usar un script que viene instalado en Linux, que se llama **adduser**.
- Este script controla como funciona **useradd** y permite realizar funciones avanzadas, como crear automáticamente el directorio del usuario, configurar su perfil, etc.



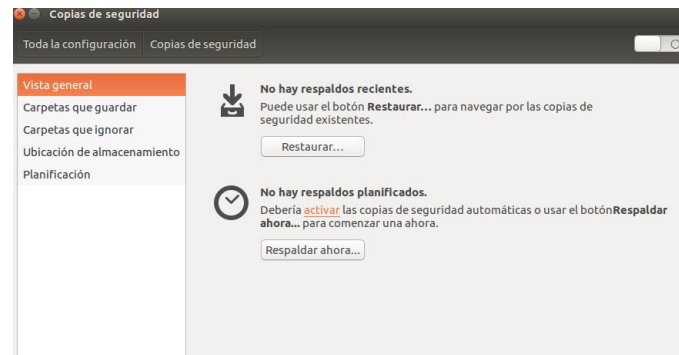


LAS COPIAS DE SEGURIDAD

LAS COPIAS DE SEGURIDAD

1. COMO HACER COPIA DE SEGURIDAD

- Ir al lanzador Configuración del sistema, seleccionamos Copias de Seguridad y se verá la siguiente pantalla:



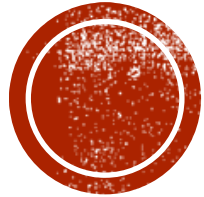
- Se muestran dos opciones:
 - **Restaurar:** para mostrará la pantalla para recuperar una copia de seguridad realizada previamente.
 - **Respalidar ahora:** para realizar una copia de seguridad ahora



LAS COPIAS DE SEGURIDAD

- En el menú de la izquierda se observan varias opciones:
 - **Vista General:** para realizar copias de seguridad.
 - **Carpetas que guardar:** para seleccionar las carpetas que deseamos que estén en la copia de seguridad.
 - **Carpetas que ignorar:** para seleccionar las carpetas que no deseamos que estén en la copia de seguridad
 - **Ubicación de almacenamiento:** para seleccionar donde guardar la copia de seguridad.
 - **Planificación:** para programar las copias de seguridad y cuanto tiempo deseamos que se mantengan en el disco.





EL MONITOR DEL SISTEMA

EL MONITOR DEL SISTEMA

- Esta utilidad nos permite realizar distintas operaciones, así como ver estadísticas sobre el rendimiento del equipo. Vamos al lanzador y escribimos Monitor del sistema.
- En la opción de **procesos** se muestran los que se están ejecutando en el sistema junto con información diversa sobre ellos.
- En el menú contextual de uno de los procesos se podrá seleccionar:



EL MONITOR DEL SISTEMA

- **Detener:** dejará de ejecutarse temporalmente.
- **Continuar:** continuará el proceso detenido temporalmente.
- **Finalizar:** dejará de ejecutarse el proceso.
- **Matar:** si no se puede finalizar, seleccionamos matar y finalizará definitivamente.
- **Cambiar la prioridad:** se abrirá una ventana donde podrá modificar la prioridad de dicho proceso.
- **Mapas de memoria:** se abrirá una ventana donde se muestre el mapa de memoria del proceso.
- **Archivos abiertos:** se abrirá una ventana donde se muestran los archivos que tiene abiertos dicho proceso.
- **Propiedades:** para ver las características del proceso.



EL MONITOR DEL SISTEMA

- En la opción de **recursos** en ella se muestra la información diversa sobre el uso de la CPU, de la memoria y de la red.
- En la opción de **Sistema de Archivos** en ella se muestra la información diversa sobre el sistema de archivos.

