SISTEMAS INFORMÁTICOS UD-3

SISTEMAS OPERATIVOS. GESTION DE ARCHIVOS Y ALMACENAMIENTO

GESTIÓN DE ARCHIVOS POR LÍNEAS DE COMANDOS

ÍNDICE

[TERMINAL O SHELL 3](#_Toc126409673)

[PERMISOS 3](#_Toc126409674)

[USUARIO ROOT 5](#_Toc126409675)

[COMANDOS MÁS UTILIZADOS EN GESTIÓN DE ARCHIVOS 5](#_Toc126409676)

[NOMBRES DE FICHEROS EN LINUX 6](#_Toc126409677)

[TIPOS DE FICHEROS 6](#_Toc126409678)

# TERMINAL O SHELL

El intérprete de comandos o Shell es la interfaz entre el usuario y el SO. Recibe las órdenes a través de la línea de comandos, las interpreta, las ejecuta y muestra el resultado.

Al entrar al sistema nos aparece un prompt con este formato:

usuario@nombre-del-equipo:directorio-actual$



Si estamos en el directorio “home”, éste se representa con el carácter “~”.

$ indica que es un usuario normal, con el administrador saldría #.

El usuario root es el administrador del sistema y puede realizar cualquier tarea de administración.



# PERMISOS

Imagen de la pantalla de un celular de un mensaje en letras blancas

Descripción generada automáticamente con confianza bajaA través del comando ls -l podemos observar los permisos de un usuario dentro de un directorio. Por ejemplo, en /home/alumno vamos a encontrar lo siguiente:

Imagen de la pantalla de un computador

Descripción generada automáticamente con confianza baja

Cada uno de estos archivos (en este caso todos son directorios) se compone de:

* **Tipo de archivo.** Es la primera letra y define el tipo de archivo. Una “d” indica directorio, una “l” un enlace simbólico, un “-“ un archivo regular, una “c” un dispositivo que trabaja carácter a carácter, una “b” un dispositivo que trabaja bloque a bloque…
* **Permisos** (compuesto por hasta 9 letras):
  + **Permisos del creador.** Son las 3 primeras letras (en el ejemplo “rwx”, permisos de lectura (r), escritura (w) y ejecución (x).
  + **Permisos del grupo principal al que pertenece el usuario creador.** Las siguientes 3 letras siguientes (en el ejemplo “r-x”, permisos de lectura y ejecución, no de escritura).
  + **Permisos del resto de usuarios.** Las últimas 3 letras (en el ejemplo “r-x”, permisos de lectura y ejecución, no de escritura).
* **Recuento de enlaces duros.**
* **Nombre del propietario.**
* **Nombre del grupo.**
* **Espacio que ocupa.**
* **Fecha de creación** (mes – día – hora)
* **Nombre del archivo.**

Texto

Descripción generada automáticamenteLos permisos definen lo que puede hacer cada usuario, por ejemplo, si queremos crear un directorio dentro de /alumno podremos hacerlo, para ello haremos uso del comando mkdir:

Si quisiéramos borrar el directorio deberíamos usar el comando rmdir seguido del nombre del directorio.

Si ahora quisiéramos crear un directorio en el directorio raíz (/) nos encontraríamos con que la terminal no nos dejará ejecutar la orden:



Esto es debido a que el usuario alumno a pesar de tener permisos de administrador no se ha logado como tal, para ello deberemos hacer uso del comando sudo su:

Texto

Descripción generada automáticamente

Si ahora intentamos crear un directorio la terminal sí que nos lo permitirá:

Texto

Descripción generada automáticamente

# USUARIO ROOT

Para ejecutar una tarea de forma puntual como root se utiliza el comando sudo: sudo comando. Para ejecutar múltiples tareas se puede abrir un Shell de root ejecutando: sudo bash.

También se puede activar la cuenta de root estableciendo su contraseña con sudo passwd root. Para cambiar a otro usuario: su usuario.

# COMANDOS MÁS UTILIZADOS EN GESTIÓN DE ARCHIVOS

|  |  |
| --- | --- |
| COMANDO | DESCRIPCIÓN |
| cd | Cambia el directorio. |
| compress | Comprime archivos (extensión .Z). |
| cp | Copia un archivo. |
| chmod | Cambia los permisos de un archivo o directorio. |
| chown | Cambia el propietario del archivo o directorio. |
| df | Muestra el espacio libre en disco. |
| du | Muestra el espacio en disco utilizado. |
| fdformat | Formatea un disquete. |
| fdisk | Particiona unidades. |
| file | Determina el tipo de archivo a través del análisis parcial de su contenido. |
| find | Encuentra un archivo. |
| fsck | Chequea el sistema de archivos. |
| gzip | Descomprime un archivo en formato GZip. |
| ln | Crea un enlace simbólico o físico depende de la opción. |
| ls | Sirve para listar el contenido de un directorio. |
| mkdir | Crea un directorio. |
| mkfs | Crea un nuevo sistema de archivos. |
| mkswap | Crea un espacio de intercambio. |
| more | Exhibe el contenido de un archivo. |
| mount | Monta una unidad o partición en el sistema de archivos. |
| mv | Mueve un archivo y se utiliza para renombrar. |
| pwd | Devuelve la cadena correspondiente al directorio actual. |
| rm | Borra un archivo. |
| rmdir | Borra un directorio. |
| swapon | Activa el espacio de intercambio. |
| swapoff | Desactiva el espacio de intercambio. |
| tar | Empaqueta o desempaqueta un archivo en formato .tar. |
| type | Muestra la ubicación de un archivo señalando su “path”. |
| umount | Desmonta una unidad montada. |

# NOMBRES DE FICHEROS EN LINUX

Los ficheros en Linux pueden tener una extensión de hasta 255 caracteres y seguir una serie de reglas:

* No se puede emplear la barra inclinada (“/”);
* No se recomienda utilizar caracteres que puedan tener un significado especial para la terminal y que podrían dar problemas, en caso de utilizarlos se deberá entrecomillar el nombre. Por ejemplo, =, ^, ~, ´, ”, `, \*, -, ?, [, ], (, ), !, &, <, >.
* Pueden contener espacios si se entrecomilla el nombre completo o se utiliza la barra “\” antes del espacio, esto informará a la Shell de que debe escapar el siguiente carácter (el del espacio).
* No es necesario utilizar extensiones, pero es recomendable utilizarlas.
* Para crear un fichero oculto hay que anteponer un punto (“.”). Por ejemplo, touch .fichero.txt, creará un fichero de nombre “fichero” y extensión “.txt” siendo además oculto.

Todos los directorios disponen de al menos dos entradas ocultas:

* “.” El directorio actual.
* “..” El directorio padre.

Estas entradas se crean automáticamente cuando se crea el directorio.

Texto

Descripción generada automáticamente

Cualquier archivo o directorio se localiza e identifica dentro del árbol de directorios gracias a su ruta, por ello, no pueden existir dos archivos con el mismo nombre en un mismo directorio.

El directorio actual “.” hace referencia a él mismo y se emplea a menudo cuando se trabaja con rutas relativas. El directorio padre se utiliza para moverse por el árbol de directorios.

# TIPOS DE FICHEROS

* **Regulares (-)**: Ficheros que contienen información de diversa naturaleza. Incluyen los ejecutables.
* **Enlaces (l).** Son los enlaces blandos.
* **Directorios (d)**: Almacenan otros ficheros.
* **Dispositivos**: Son archivos que representan a dispositivos físicos y se localizan en su mayoría en el directorio /dev. Tenemos dos tipos:
  + **Los que trabajan carácter a carácter (c)**. Son aquellos que no disponen de un sistema de archivos, como impresoras, teclados, etc. Transfieren los datos carácter a carácter.
  + **Los que trabajan por bloques (b).** Son aquellos que almacenan la información bloques de datos, como los HDD, los SSD, las unidades flash…

También existen dispositivos virtuales que tienen un tratamiento especial como /dev/null que es una cubeta de bits que se emplea para eliminar o descartar toda la información que sea enviada a él (funciona como una papelera).

El comando ls dispone de la opción “--color” activada por defecto en Ubuntu, que permite discriminar el tipo de archivo según el color:

* Blanco: archivo regular
* Verde: archivo ejecutable
* Azul: directorio.
* Cian: enlace simbólico.
* Rojo: enlace roto.

