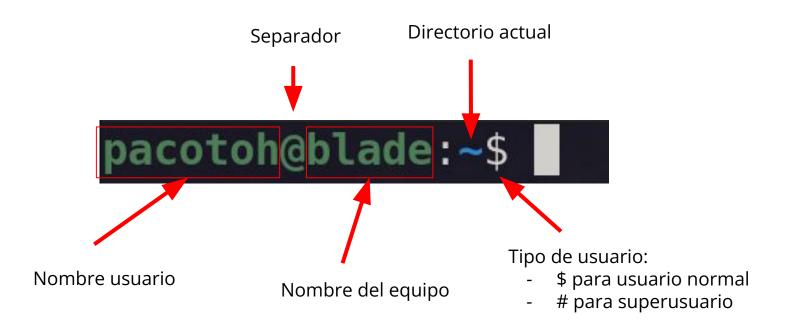


## UTO. Administración básica de Linux

PSP - DAM Francisco Gallego Perona

# El prompt



# Navegación

Lo primero que debemos aprender es a movernos de forma fluida por el sistema de ficheros (en la terminal). Para ello contamos con los comandos:

- pwd: Imprime el nombre del directorio actual (directorio de trabajo o working directory)
- **cd**: Cambia al directorio indicado como parámetro
- Is: Lista el contenido de un directorio

# Navegación: comando pwd

En el siguiente caso vemos el uso del comando pwd en el directorio "home" del usuario "pacotoh". Como vemos, nos muestra la ruta del sistema en la que nos encontramos actualmente.

```
pacotoh@blade:~$ pwd
/home/pacotoh
pacotoh@blade:~$
```

El comando cd nos sirve para movernos por los directorios del sistema. En la imagen podemos ver que nos encontrábamos en /home/pacotoh y nos movemos a la carpeta /home/pacotoh/Documents.

```
pacotoh@blade:~$ pwd
/home/pacotoh
pacotoh@blade:~$ cd Documents/
pacotoh@blade:~/Documents$ pwd
/home/pacotoh/Documents
pacotoh@blade:~/Documents$
```

Hay varios parámetros especiales para el comando cd, entre ellos están:

- cd → Si no indicamos ningún parámetro, nos moverá al directorio home del usuario logueado en la sesión.
- $\bullet$  cd.  $\rightarrow$  Con un punto indicamos que nos movemos al directorio actual.
- cd .. → Con dos puntos nos movemos al directorio anterior en la jerarquía.
- $cd \rightarrow Nos movemos al directorio de trabajo anterior.$

En la siguiente imagen podemos ver un listado de todas las carpetas dentro del directorio actual (/home/pacotoh/Documents/)

```
pacotoh@blade:~/Documents$ ls
BlackmagicDesign Blender Code Godot lmms LUTs Meshroom
pacotoh@blade:~/Documents$
```

También podemos utilizar **is** con opciones para mostrar listados detallados o de ficheros especiales (como ficheros ocultos).

Algunos de estos listados son:

- ls -l
- II (esta opción es la misma que ls -la)
- **Is -la** (además del listado más detallado nos muestra los ficheros ocultos)

```
pacotoh@blade:~/Documents/Blender$ ls -lh
total 296M
drwxr-xr-x 5 pacotoh pacotoh 4,0K ago 3 10:51 3.2
-rwxr-xr-x 1 pacotoh pacotoh 294M ago 3 01:52 blender
-rw-r--r-- 1 pacotoh pacotoh 5,6K ene 28 2022 blender.desktop
-rwxr-xr-x 1 pacotoh pacotoh 713 ene 28 2022 blender-softwaregl
-rw-r--r-- 1 pacotoh pacotoh 1,7K ene 28 2022 blender.svg
-rw-r--r-- 1 pacotoh pacotoh 3,8K ene 28 2022 blender-symbolic.svg
-rwxr-xr-x 1 pacotoh pacotoh 2,1M jul 21 12:15 blender-thumbnailer
-rw-r--r-- 1 pacotoh pacotoh 4,7K ene 28
                                         2022 copyright.txt
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 lib
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 license
-rw-r--r-- 1 pacotoh pacotoh 5,1K may 2 12:54 readme.html
pacotoh@blade:~/Documents/Blender$
```

Opción ls -lh:

Tipo de fichero: d (directorio) - (fichero) l (enlace)

```
pacotoh@blade: / Documents/Blender$ ls -lh
total 296M
drwxr xr-x 5 pacotoh pacotoh 4,0K ago 3 10:51 3.2
-rwxr-xr-x 1 pacotoh pacotoh 294M ago 3 01:52 blender
-rw-r--r-- 1 pacotoh pacotoh 5,6K ene 28 2022 blender.desktop
-rwxr-xr-x 1 pacotoh pacotoh 713 ene 28 2022 blender-softwaregl
-rw-r--r-- 1 pacotoh pacotoh 1,7K ene 28 2022 blender.svg
-rw-r--r-- 1 pacotoh pacotoh 3,8K ene 28 2022 blender-symbolic.svg
-rwxr-xr-x 1 pacotoh pacotoh 2,1M jul 21 12:15 blender-thumbnailer
-rw-r--r-- 1 pacotoh pacotoh 4,7K ene 28
                                         2022 copyright.txt
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 lib
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 license
-rw-r--r-- 1 pacotoh pacotoh 5,1K may 2 12:54 readme.html
pacotoh@blade:~/Documents/Blender$
```

Opción ls -lh:

Permisos de owner, group y others

```
pacotoh@blade: -/Documents/Blender$ ls -lh
total 296M
drwxr-xr-x 5 pacotoh pacotoh 4,0K ago 3 10:51 3.2
-rwxr-xr-x 1 pacotoh pacotoh 294M ago 3 01:52 blender
-rw-r--r-- 1 pacotoh pacotoh 5,6K ene 28 2022 blender.desktop
-rwxr-xr-x 1 pacotoh pacotoh 713 ene 28 2022 blender-softwaregl
-rw-r--r-- 1 pacotoh pacotoh 1,7K ene 28 2022 blender.svg
-rw-r--r-- 1 pacotoh pacotoh 3,8K ene 28 2022 blender-symbolic.svg
-rwxr-xr-x 1 pacotoh pacotoh 2,1M jul 21 12:15 blender-thumbnailer
-rw-r--r-- 1 pacotoh pacotoh 4,7K ene 28 2022 copyright.txt
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 lib
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 license
-rw-r--r-- 1 pacotoh pacotoh 5,1K may 2 12:54 readme.html
pacotoh@blade:~/Documents/Blender$
```

Número de enlaces al contenido

```
pacotoh@blade:~/Documents/Blender$ ls -lh
total 296M
drwxr-xr-x 5 pacotoh pacotoh 4,0K ago 3 10:51 3.2
-rwxr-xr-x 1 pacotoh pacotoh 294M ago 3 01:52 blender
-rw-r--r-- 1 pacotoh pacotoh 5,6K ene 28 2022 blender.desktop
-rwxr-xr-x 1 pacotoh pacotoh 713 ene 28 2022 blender-softwaregl
-rw-r--r-- 1 pacotoh pacotoh 1,7K ene 28 2022 blender.svg
-rw-r--r-- 1 pacotoh pacotoh 3,8K ene 28 2022 blender-symbolic.svg
-rwxr-xr-x 1 pacotoh pacotoh 2,1M jul 21 12:15 blender-thumbnailer
-rw-r--r-- 1 pacotoh pacotoh 4,7K ene 28 2022 copyright.txt
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 lib
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 license
-rw-r--r-- 1 pacotoh pacotoh 5,1K may 2 12:54 readme.html
pacotoh@blade:~/Documents/Blender$
```

Owner y grupo del owner

```
pacotoh@blade:~/Documents/Blander$ ls -lh
total 296M
drwxr-xr-x 5 pacotoh pacotoh 4,0K ago 3 10:51 3.2
-rwxr-xr-x 1 pacotoh pacotoh 294M ago 3 01:52 blender
-rw-r--r-- 1 pacotoh pacotoh 5,6K ene 28 2022 blender.desktop
-rwxr-xr-x 1 pacotoh pacotoh 713 ene 28 2022 blender-softwaregl
-rw-r--r-- 1 pacotoh pacotoh 1,7K ene 28 2022 blender.svg
-rw-r--r-- 1 pacotoh pacotoh 3,8K ene 28 2022 blender-symbolic.svg
-rwxr-xr-x 1 pacotoh pacotoh 2,1M jul 21 12:15 blender-thumbnailer
-rw-r--r-- 1 pacotoh pacotoh 4,7K ene 28
                                         2022 copyright.txt
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 lib
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 license
-rw-r--r-- 1 pacotoh pacotoh 5,1K may 2 12:54 readme.html
pacotoh@blade:~/Documents/Blender$
```

Tamaño del fichero

```
pacotoh@blade:~/Documents/Blender$ ls _-tn
total 296M
drwxr-xr-x 5 pacotoh pacotoh 4,0K ago 3 10:51 3.2
-rwxr-xr-x 1 pacotoh pacotoh 294M ago 3 01:52 blender
-rw-r--r-- 1 pacotoh pacotoh 5,6K ene 28 2022 blender.desktop
-rwxr-xr-x 1 pacotoh pacotoh 713 ene 28 2022 blender-softwaregl
-rw-r--r-- 1 pacotoh pacotoh 1,7K ene 28 2022 blender.svg
-rw-r--r-- 1 pacotoh pacotoh 3,8K ene 28 2022 blender-symbolic.svg
-rwxr-xr-x 1 pacotoh pacotoh 2,1M jul 21 12:15 blender-thumbnailer
-rw-r--r-- 1 pacotoh pacotoh 4,7K ene 28
                                         2022 copyright.txt
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 lib
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 license
-rw-r--r-- 1 pacotoh pacotoh 5,1K may 2 12:54 readme.html
pacotoh@blade:~/Documents/Blender$
```

Última fecha de modificación

```
pacotoh@blade:~/Documents/Blender$ ls -lh
total 296M
drwxr-xr-x 5 pacotoh pacotoh 4,0K ago 3 10:51 3.2
-rwxr-xr-x 1 pacotoh pacotoh 294M ago 3 01:52 blender
-rw-r--r-- 1 pacotoh pacotoh 5,6K ene 28 2022 blender.desktop
-rwxr-xr-x 1 pacotoh pacotoh 713 ene 28 2022 blender-softwaregl
-rw-r--r-- 1 pacotoh pacotoh 1,7K ene 28 2022 blender.svg
-rw-r--r-- 1 pacotoh pacotoh 3,8K ene 28 2022 blender-symbolic.svg
-rwxr-xr-x 1 pacotoh pacotoh 2,1M jul 21 12:15 blender-thumbnailer
-rw-r--r-- 1 pacotoh pacotoh 4,7K ene 28 2022 copyright.txt
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 lib
drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 license
-rw-r--r-- 1 pacotoh pacotoh 5,1K may 2 12:54 readme.html
pacotoh@blade:~/Documents/Blender$
```

Nombre del fichero Opción ls -lh: pacotoh@blade:~/Documents/Blender\$ ls -lh total 296M drwxr-xr-x 5 pacotoh pacotoh 4,0K ago 3 10:51 3.2 -rwxr-xr-x 1 pacotoh pacotoh 294M ago 3 01:52 blender -rw-r--r-- 1 pacotoh pacotoh 5,6K ene 28 2022 blender.desktop -rwxr-xr-x 1 pacotoh pacotoh 713 ene 28 2022 blender-softwaregl -rw-r--r-- 1 pacotoh pacotoh 1,7K ene 28 2022 blender.svg -rw-r--r-- 1 pacotoh pacotoh 3,8K ene 28 2022 blender-symbolic.svg -rwxr-xr-x 1 pacotoh pacotoh 2,1M jul 21 12:15 blender-thumbnailer -rw-r--r-- 1 pacotoh pacotoh 4,7K ene 28 2022 copyright.txt drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 lib drwxr-xr-x 2 pacotoh pacotoh 4,0K ago 3 10:51 license -rw-r--r-- 1 pacotoh pacotoh 5,1K may 2 12:54 readme.html pacotoh@blade:~/Documents/Blender\$

# Rutas absolutas y relativas

Para cambiar el directorio de trabajo, sabemos que podemos usar el comando **cd** seguido del **pathname** al que nos queremos mover, pero tenemos dos opciones para representar dicho pathname:

 Ruta absoluta (absolute pathname): Es la ruta completa, desde la raíz (/) hasta el directorio/fichero deseado. NO DEPENDE DE DÓNDE NOS ENCONTRAMOS

Ejemplo de ruta absoluta: /usr/bin

• Ruta relativa (relative pathname): Es la ruta dependiendo de dónde nos encontramos.

Ejemplo de ruta relativa:

Si estamos en el directorio **/etc/** podemos navegar hasta conf.d con la ruta  $\rightarrow$  **alsa/conf.d/** 

# Ejercicio de rutas relativas y absolutas

- Nos encontramos en el directorio /home y queremos movernos al directorio tmp que se encuentra a su vez en el directorio var, dentro del directorio raíz.
  - a) Indica la ruta **absoluta** al directorio **tmp**
  - b) Indica la ruta **relativa** al directorio **tmp**
- 2. Nos encontramos en el directorio **/usr/bin** y queremos ir al directorio anterior, usr. Indica dos opciones para movernos a dicho directorio.

## Explorando el sistema

Existen varios comandos útiles para explorar el sistema de ficheros. Algunos de ellos son:

**file**: sirve para determinar el tipo de un fichero

# Explorando el sistema

Otros comandos importantes para ver el contenido de ficheros son:

- **less**: Con este comando podemos **navegar** el contenido de un fichero (es un programa de los llamados "paginadores" o "pagers", programas que te permiten ver el contenido de un fichero en distintas páginas.
- **cat**: Con este comando podemos **mostrar por terminal** el contenido de un fichero.
- more: Comando heredado de Unix, sustituido por less.

## Explorando el sistema: directorios en Linux

- / → El directorio raíz, donde empieza todo.
- /bin → Contiene ficheros binarios (programas) que deben estar presentes en el sistema para arrancar y ser ejecutados.
- /boot → Contiene el kernel Linux
  - /boot/grub/grub.conf y menu.lst  $\rightarrow$  usados para configurar el boot loader /boot/vmlinuz  $\rightarrow$  Kernel Linux
- /dev → Directorio especial donde se encuentran los device nodes (se utilizan para realizar llamadas I/O)

## Explorando el sistema: directorios en Linux

- /etc → Contiene ficheros de configuración. También contiene una colección de shell scripts que inician servicios en el arranque del sistema.
  - /etc/crontab → Fichero que define los trabajos automatizados que se ejecutarán en el sistema.
  - $\circ$  /etc/fstab  $\to$  Tabla de relaciones entre dispositivos de almacenamiento y su correspondiente punto de montaje.
  - o /etc/passwd → Lista de cuentas de usuario.
- /home → Directorio donde se encuentran los directorios de los usuarios ordinarios (un usuario ordinario estaría limitado a su correspondiente home)
- ✓ Iib → Contiene librerías necesarias para el sistema (similar a los DLLs en Windows)
- /media → Contiene los puntos de montaje de los dispositivos removibles como USBs.

# Creación de ficheros

OJO: Los comandos touch y cat se pueden utilizar para crear y mostrar múltiples ficheros. Ejemplo: **touch uno dos tres cuatro** → Crea los 4 ficheros

En Linux tenemos varias formas de crear ficheros, algunas de estas son:

- Comando touch [NOMBRE\_FICHERO]: Nos creará un fichero vacío.
- Comando nano/vi/vim (editores): Podemos crear ficheros y modificarlos directamente en la terminal con estos comandos (nosotros usaremos nano como editor por defecto)
- echo "El mensaje que queramos" > nuestro\_fichero
- cat fichero\_antiguo > fichero\_nuevo
- cp fichero\_antiguo fichero\_nuevo

Son muchas las opciones, todas igual de válidas.

## Creación de directorios

Para crear directorios usamos el comando:

mkdir [DIRECTORIO]

Donde [DIRECTORIO] pueden ser 1 o más directorios a crear, separados por espacios.

Para crear ficheros/directorios podemos usar, tanto rutas relativas como rutas absolutas.

mkdir /home/pacotoh/NuevosDocumentos

mkdir NuevosDocumentos (estando en la carpeta /home/pacotoh)

# Eliminación de ficheros

El comando para eliminar ficheros es **rm**.

### rm [FICHERO]+

Para eliminar directorios debemos tener en cuenta que pueden contener ficheros o más carpetas dentro, por lo que el borrado debe ser recursivo.

#### rm -r [DIRECTORIO]+

También podemos utilizar el comando **rmdir** (el comando rm es preferible)

## Actualización del sistema

Los comandos para mantener las versiones de los paquetes, así como del núcleo de nuestro sistema Linux son los siguientes:

- sudo apt update → Actualiza la lista de paquetes disponibles y sus versiones (ver /etc/apt/sources.list)
- sudo apt upgrade ightarrow Instala nuevas versiones de los paquetes que se tienen en el sistema
- sudo apt dist-upgrade
- sudo reboot

Una opción rápida es usar los comandos update y upgrade juntos de la siguiente forma:

#### sudo apt update; sudo apt upgrade -y

La opción -y hace que el upgrade no te pida confirmar la acción.

La administración de usuarios en GNU/Linux se basa en archivos de configuración:

- /etc/passwd: Se encarga de almacenar los datos de los usuarios del sistema operativo.
- **/etc/shadow**: Archivo que solo puede ser accedido por root, en el que se guardan las contraseñas (encriptadas), entre otras cosas.
- **/etc/group**: Guarda toda la información relacionada con los grupos de usuarios a los que pertenecen los usuarios del sistema.

Los usuarios de Unix/Linux se identifican por un número único de usuario (User ID, UID). Pertenecen también a un grupo principal de usuario, identificado por un número único de grupo (Group ID, GID).

El usuario puede pertenecer a más grupos además del principal.

Podemos identificar 3 tipos de usuarios:

- Superusuario (root).
- Usuario especial: No pueden iniciar sesión y pertenecen al sistema, teniendo acceso a servicios específicos.
- 3. **Usuarios normales**: con funcionalidad y permisos limitados.

Cómo **añadir** usuarios (siempre como usuario root):

#### useradd [OPCIONES] [NOMBRE\_USUARIO]

Las **opciones** pueden ser:

- -d: Especifica el directorio home del usuario.
- -s: Especifica el shell del usuario.
- -g: Especifica el grupo primario del usuario.
- -G: Especifica los grupos secundarios del usuario.
- -M: Especifica que no se cree un directorio de inicio para el usuario.
- -e: Especifica la fecha de expiración del usuario.
- -uid: Especifica el identificador de usuario para el usuario.

Otra opción más básica para añadir usuarios es:

adduser [NOMBRE\_USUARIO] [GRUPO\_USUARIO]

Con esta opción se requiere una contraseña para el usuario en el momento de su creación. Sin embargo, en la opción de useradd es necesario crear la contraseña posteriormente.

También creará la carpeta home del usuario con el comando adduser.

Ejemplo de useradd:

useradd -d /home/frodo -s /bin/bash -u 800 -c "Frodo Bolson" frodo

Cómo modificar usuarios:

usermod -d /home/bilbo frodo

De esta forma estamos cambiando el directorio home de frodo al de bilbo.

#### **Actividad:**

Crea un usuario llamado "gimli" y modifícale su directorio personal a /home/gimli\_hijo\_de\_gloin. Crea una captura del resultado.

#### Cómo **eliminar** usuarios:

Para eliminar usuarios se utiliza el comando userdel, que elimina la cuenta del sistema. Hay 3 opciones de uso:

- userdel legolas → Elimina la cuenta pero no el directorio de trabajo (home)
- **userdel -r legolas** → Elimina la cuenta y el directorio de trabajo
- userdel -f legolas → Igual que la opción -r, pero con la opción -r, si el usuario está conectado en una sesión, no lo desconecta. Con la opción -f fuerza la eliminación sin importar el precio.

#### Cómo **eliminar** usuarios:

Para eliminar usuarios se utiliza el comando userdel, que elimina la cuenta del sistema. Hay 3 opciones de uso:

- userdel legolas → Elimina la cuenta pero no el directorio de trabajo (home)
- **userdel -r legolas** → Elimina la cuenta y el directorio de trabajo
- userdel -f legolas → Igual que la opción -r, pero con la opción -r, si el usuario está conectado en una sesión, no lo desconecta. Con la opción -f fuerza la eliminación sin importar el precio.

Cómo cambiar contraseñas de usuarios:

Crear el usuario con useradd es el primer paso; el segundo es asignarle una contraseña a ese usuario, mediante el comando passwd.

### passwd gimli

A partir de la introducción de este comando se nos pide la nueva contraseña y la confirmación de dicha contraseña.

## Ejercicio de administración de usuarios

Crea capturas en las que se vean todos los comandos utilizados:

Crea los grupos: hobbits, dwarfs, elven, humans y gandalf

Crea los usuarios frodo, bilbo y sam en el grupo hobbits

Crea los usuarios: gimli y gloin en el grupo dwarfs

Crea al usuario legolas en el grupo elven

Crea a aragorn y boromir en el grupo humans

Crea al usuario gandalf en el grupo gandalf

Establece la contraseña de gandalf a mithrandir

Entra en la sesión como el usuario gandalf

## Administración de aplicaciones

Comandos relacionados con la instalación de paquetes en GNU/Linux:

- dpkg: Gestor de paquetes .deb para instalar aplicaciones Debian manualmente.
- apt-get: Comando de bajo nivel para instalar paquetes y dependencias de forma automática.
- **apt**: Versión moderna del gestor de paquetes apt-get.

Otras herramientas de gestión de paquetes existentes son aptitude o synaptic, que no vienen por defecto en la instalación de Ubuntu Server.

## Administración de aplicaciones

#### Comando apt-get:

Herramienta de gestión de paquetes desde la línea de comandos asociado a las fuentes y los repositorios del archivo /etc/apt/source.list.

#### Algunas de las opciones del apt-get son:

- **apt-get update**: Actualiza los repositorios en el sources.list
- **apt-get install**: Instala uno o varios paquetes
- **apt-get upgrade**: Actualiza los programas y paquetes instalados
- **apt-get dist-upgrade**: Actualiza la totalidad de los paquetes instalados
- **apt-get remove**: Desinstala uno o varios paquetes instalados (mantiene archivos de configuración)
- **apt-get purge**: Desinstala los paquetes y los archivos de configuración de las utilidades desinstaladas.
- **apt-get autoremove**: Desinstala las dependencias de programas que en su día eran necesarias, pero ahora ya no.

## Administración de aplicaciones

#### Comando dpkg:

Muchas veces es necesario instalar programas manualmente, pues no se encuentran en los repositorios y solo se tiene un archivo .deb para su instalación.

#### dpkg -i nombre\_archivo.deb

Del mismo modo que instalamos con -i (install), podemos desinstalar con -r (remove) o ver el contenido del .deb con -l (list).

## Administración de aplicaciones

apt: una forma más amigable de gestionar paquetes.

apt-get tiene muchos parámetros y opciones; con apt solo se tiene acceso a las más usadas, siendo esta una variante más fácil y sencilla.

Ejemplo:

sudo apt install vlc

## Permisos

En GNU/Linux, el acceso, tanto de usuarios como de los grupos de usuarios, a ciertos recursos del sistema se determina mediante permisos.

Permisos de owner, group y others

#### Permisos

Establecer permisos: Comando chmod (change mode)

Para determinar el tipo de usuario:

- u: dueño del archivo
- g: grupo al que pertenece el archivo
- o: otros, todos los usuarios que no son, ni el dueño ni del grupo

#### Ejemplos:

```
chmod u+x script.sh
chmod -x script.sh
```

chmod o+r+w script.sh

#### Permisos

Los permisos de acceso para usuario, grupo y otros se almacenan en bits que indica si tiene el permiso (1) o no lo tiene (0).

Por ejemplo:

u rwx	g rwx	o rwx

En este caso, convertimos los bits a octal (de 0 a 7) y tenemos que los permisos a aplicar serán los siguientes:

chmod 754 fichero.txt

# Cambio de propietario y grupo

Comando chown para cambiar el propietario de un archivo.

Comando chgrp para cambiar el grupo de usuarios al que pertenece el archivo.

chown [USUARIO] [ARCHIVO/S]

chgrp [GRUPO] [ARCHIVO/S]

Una forma rápida para cambiar el usuario y el grupo a la vez es la siguiente:

chown pepe.alumno fichero.pdf

chown pepe:alumno fichero.pdf

Se puede usar tanto . como : como separador de usuario.grupo.

# Ejercicio de Permisos

Crea una carpeta en tu directorio personal y asigna permisos para que solo el propietario tenga permisos de lectura, escritura y ejecución, el resto no debe tener ningún permiso.

Usa dos formas distintas, usando **octal** y mediante las opciones para **dar y quitar permisos**.

# Mover y renombrar ficheros

El comando my sirve para mover y renombrar ficheros.

Ejemplo:

mv fichero fichero2 → Renombramos el fichero

**mv fichero directorio** → Movemos fichero al directorio

# Comandos de ayuda de comandos

 type: Indica el tipo de comando que el shell va a ejecutar, dado un nombre de comando.

type [comando]

#### Ejemplos:

```
type type \rightarrow type is a shell builtin
type ls \rightarrow ls is aliased to `ls --color=auto'
type cp \rightarrow cp is /usr/bin/cp
```

## Comandos de ayuda de comandos

- which

Algunas veces hay más de una versión instalada de un programa ejecutable en el sistema → Esto es muy usual en servidores de gran tamaño.

El comando which nos devuelve la localización del comando.

```
pacotoh@blade:~$ which ls
/usr/bin/ls
```

- help

Bash tiene una ayuda preinstalada. Para usarla podemos escribir "help" y el

nombre del comando.

```
pacotoh@blade:~$ help cd
cd: cd [-L|[-P [-e]] [-@]] [dir]
    Change the shell working directory.
    Change the current directory to DIR. The default DIR is the value of the
   HOME shell variable.
    The variable CDPATH defines the search path for the directory containing
    DIR. Alternative directory names in CDPATH are separated by a colon (:).
   A null directory name is the same as the current directory. If DIR begins
   with a slash (/), then CDPATH is not used.
    If the directory is not found, and the shell option `cdable vars' is set,
    the word is assumed to be a variable name. If that variable has a value,
    its value is used for DIR.
   Options:
                force symbolic links to be followed: resolve symbolic
               links in DIR after processing instances of `..'
               use the physical directory structure without following
               symbolic links: resolve symbolic links in DIR before
               processing instances of `..'
               if the -P option is supplied, and the current working
               directory cannot be determined successfully, exit with
```

- man

La mayoría de programas ejecutables en Linux cuentan con una documentación formal llamada "manual" o "man page". El programa man sirve

para ver dicha documentación.

```
LS(1)
                                                                         LS(1)
                                 User Commands
NAME
       ls - list directory contents
SYNOPSIS
       ls [OPTION]... [FILE]...
DESCRIPTION
       List information about the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
       fied.
       Mandatory arguments to long options are mandatory for short options
       too.
       -a. --all
              do not ignore entries starting with .
       -A. --almost-all
              do not list implied . and ..
       --author
 Manual page ls(1) line 1 (press h for help or q to quit)
```

apropos

Este comando sirve para buscar la lista de páginas de manual basada en un término de búsqueda.

https://web.archive.org/web/20170601064537/http://plan9.bell-labs.com/7thEdMan/v7vol1.pdf

apropos [comando]

```
pacotoh@blade:~$ apropos ls
                     - reposition read/write file offset
llseek (2)
                     - ALSA sequencer connection manager
aconnect (1)
                     - add shells to the list of valid login shells
add-shell (8)
afs syscall (2)
                     - unimplemented system calls
                     - command-line utility to gather information about the A..
alsa-info (8)
                     - command-line sound tester for ALSA sound card driver
alsabat (1)
                     - advanced controls for ALSA soundcard driver
alsactl (1)
alsactl init (7)
                     - alsa control management - initialization
```

- whatis

Este comando muestra el nombre y la descripción (en una línea) de la página de manual coincidente con el comando concreto.

#### whatis [comando]

```
pacotoh@blade:~$ whatis ls
ls (1) - list directory contents
pacotoh@blade:~$ ■
```

- info

El proyecto GNU añade una alternativa al comando "man" para ver la **información de sus programas**. La información con este comando es mostrada con un lector especial, que nos permite navegar por el documento.

- info

```
Next: Introduction, Up: (dir)
The GNU nano text editor
*********
This manual documents GNU 'nano', version 6.2.
* Menu:
* Introduction::
* Invoking::
 Editor Basics::
<u>* The Help Viewer::</u>
 The File Browser::
 Command-line Options::
 Feature Toggles::
 Nanorc Files::
 <u>Pico Compatibility</u>::
  Building and its Options::
```

- info → Comandos dentro de info

Command	Action	
?	Display command help	
PgUp or Backspace	Display previous page	
PgDn or Space	Display next page	
n	Next - Display the next node	
р	Previous - Display the previous node	
u	Up - Display the parent node of the currently displayed node, usually a menu.	
Enter	Follow the hyperlink at the cursor location	
q	Quit	

- Ficheros **README** y otra documentación

Algunos paquetes de software instalados en el sistema tienen ficheros que residen en /usr/share/doc. La mayoría de estos están almacenados en texto plano y pueden ser leídos mediante los comandos (nano, cat, less, zless).

#### **Ejercicio:**

Entra en el directorio /usr/share/doc/git y abre la documentación del README.md, así como la de los ficheros comprimidos gz.

Este es el primer paso a la automatización del sistema, la base de la programación en shell. Vamos a crear nuestros propios comandos usando alias.

- Imagina que quieres ejecutar varios comandos, uno detrás del otro, y quieres repetir dichos comandos en numerosas ocasiones (comandos como la actualización del sistema o realizar una copia de seguridad)
- Podemos ejecutar dichos comandos usando; después de cada comando:

comando1; comando2; comando3

Un ejemplo de la ejecución de varios comandos:

cd /usr; ls -l > ~/Documents/usr\_file; cd /

Como vemos, tenemos 3 comandos combinados en una sola línea:

- Cambiamos al directorio /usr
- 2) Hacemos ls -l y redirigimos la salida a usr\_file, dentro de documentos
- 3) Nos movemos al directorio raíz /

En lugar de tener que utilizar los comandos uno a uno o buscarlos en el historial para ejecutar los 3 a la vez, podemos crear un **alias**, que consiste en un comando nuevo que ejecuta la línea con los 3 comandos juntos.

Se hace de la siguiente forma:

alias usr\_list='cd /usr; ls -l > ~/Documents/usr\_file; cd /'

Una vez introducido podemos comprobar que el alias existe en el sistema usando el comando:

alias → Nos muestra la lista de alias definidos en la sesión

Para eliminar un alias en la sesión podemos utilizar el comando:

**unalias usr\_list** → Eliminamos el alias creado anteriormente

Los alias que creemos se borrarán en cuanto salgamos de la sesión del usuario. Para mantenerlos en el sistema aunque cerremos la sesión debemos modificar el fichero .bashrc, en la carpeta del usuario.

El . delante del fichero indica que se trata de un fichero oculto. Para verlo podemos utilizar el comando ls con la opción -a.

```
GNU nano 6.2
                                        .bashrc
    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
# colored GCC warnings and errors
#export GCC COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:q>
# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
# custom aliases
alias up='sudo apt update; sudo apt upgrade -y'
# Add an "alert" alias for long running commands. Use like so:
    sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || ech>
```

# Redirección y operaciones de I/O

#### Resumen de comandos:

- Salida estándar y de error
- cat
- sort
- uniq
- grep
- WC
- head
- tail
- tee

Muchos de los programas que hemos usado producen una salida de algún tipo. Los tipos de salida son:

- Resultado de un programa  $\rightarrow$  Ej: un listado en el comando ls
- El estado y mensajes de error → Cómo ha terminado nuestro programa

#### TODO ES UN FICHERO, HASTA LA SALIDA ESTÁNDAR.

Cómo redirigimos nuestra salida estándar a un fichero → usando >

Redirección de la salida estándar:

**Is -I /usr/bin > Is-output**  $\rightarrow$  En este caso se redirige la salida del ls -I a un fichero llamado Is-output

**Is -I /bin/usr > Is-output** → ¿Qué pasa en este caso? ¿Qué contiene ls-output?

Concatenación de la salida estándar:

Is -l /usr/bin >> Is-output

En este caso, sea lo que tengamos en el fichero ls-output, añadimos al final lo obtenido del comando utilizado (ls -l /usr/bin).

Redirección de la salida de error:

ls -l /bin/usr 2> ls-output-error

Sabemos que, el directorio /bin/usr no existe en nuestro sistema, por lo que tendremos un error en la ejecución del comando ls -l.

Utilizando la redirección 2> mandaremos la salida de error al fichero indicado.

Redirección de salida estándar y salida de error a la vez:

**Is -I /bin/usr > Is-output 2>&1**  $\rightarrow$  El orden importa. Primero salida estándar y por último salida de error (SIEMPRE).

Para ver lo que es un descriptor de fichero:

https://es.wikipedia.org/wiki/Descriptor de archivo

Redirección de salida estándar y salida de error a la vez (II):

Esta es la **nueva y mejor versión** de la redirección estándar y error a la vez.

Is -I /bin/usr &> Is-output

Is -I /bin/usr &>> Is-output → También se puede concatenar

¿Y si no queremos usar la salida de error para nada?

La podemos descartar usando:

**Is -I /bin/usr 2> /dev/null** → /dev/null es un tipo especial de fichero llamado **bit bucket**, que acepta una entrada y no hace absolutamente nada con ella.

https://es.wikipedia.org/wiki//dev/null

#### Pipelines:

La forma de redirigir desde la salida estándar de un comando hacia la entrada estándar de otro comando se realiza mediante pipes o tuberías.

comando1 | comando2

Ejemplo:

**Is -l /usr/bin | less** → La salida del ls -l es pasada al comando less

Comando cat:

Lee uno o varios ficheros y los copia a la salida estándar.

cat [FICHERO]+

Un ejemplo útil de cat sería el siguiente:

Imaginemos que tenemos en nuestro directorio varios ficheros que empiezan por la palabra test. Queremos redirigirlos TODOS a un nuevo fichero.

```
pacotoh@blade:~/Documents/so/cat$ ls test1 test2 test3 test4 pacotoh@blade:~/Documents/so/cat$
```

Usando el comando cat podemos redirigirlos de la siguiente manera:

```
pacotoh@blade:~/Documents/so/cat$ cat test* > todos_test
pacotoh@blade:~/Documents/so/cat$ ls
test1 test2 test3 test4 todos_test
pacotoh@blade:~/Documents/so/cat$
```

Hemos usado un cat para introducir todo el contenido de los ficheros test en un nuevo fichero llamado todos\_test.

Crea el mismo ejemplo introduciendo texto dentro de cada uno de los ficheros test y viendo cómo queda el fichero final de todos\_test.

Comando sort:

Is /bin /usr/bin | sort | less

Con el comando sort ordenamos la salida estándar. En este caso ordenamos la salida de los ls, es decir, del listado de ficheros y directorios de /bin y /usr/bin.

Por último, redirigimos la salida del sort al comando less, haciendo que podamos navegar por la salida desde el principio hasta el final.

Comando uniq:

Is /bin /usr/bin | sort | uniq | less

El comando uniq es muy utilizado con el comando sort. Con este comando eliminamos los duplicados de la salida estándar del sort.

Para ver solo los duplicados usamos la opción -d:

Is /bin /usr/bin | sort | uniq -d | less

#### Comando wc:

El comando wc (word count) es usado para mostrar el número de líneas, palabras y bytes que contiene un fichero.

wc [FICHERO]+

#### Ejemplo:

```
pacotoh@blade:~/Documents$ echo "Hola mundo" > hola_mundo
pacotoh@blade:~/Documents$ wc hola_mundo
1 2 11 hola_mundo
pacotoh@blade:~/Documents$
```

Si queremos mostrar **solo la cuenta de las líneas** dentro de un fichero:

wc -l [FICHERO]

Ejemplo:

```
pacotoh@blade:~/Documents$ echo "adios" > adios_mundo
pacotoh@blade:~/Documents$ wc -l hola_mundo adios_mundo
  1 hola_mundo
  1 adios_mundo
  2 total
```

Comando grep:

Es un comando muy poderoso para buscar patrones de texto.

grep [PATRÓN] [FICHERO]+