

T5.2: Linux

Ubuntu

Modesto Tomás Saavedra 22/23

Índice de contenidos:

5.2.1 Intérprete de comandos.

5.2.2 Gestor de paquetes.

5.2.3 Instalación manual de paquetes

5.2.4 Estructura de directorios

5.2.5 Usuarios y grupos

5.2.6 Permisos

5.2.1 Intérprete de comandos

Bash

- Es una **interfaz** entre el usuario y el sistema operativo.
- Funciona mediante un bucle **read-eval-print**.
 - Recibe las órdenes a través de la línea de comandos, las interpreta y muestra su resultado.
- Resulta fundamental para obtener el máximo rendimiento del sistema, ya que nos permite configurar y automatizar multitud de aspectos.
- Para iniciar el intérprete de comandos hay que pulsar **CTRL + T**.
- Una vez que se accede al sistema se muestra un prompt con el siguiente aspecto:

```
usuario@nombre-del-equipo:~$
```

5.2.1 Intérprete de comandos

Bash

- En el caso de que aparezca el carácter ~ es porque el terminal se encuentra en el directorio **/home**.
- El símbolo \$ o # indica si es un usuario normal o administrador del sistema.
- El usuario **root** es el administrador del sistema y **puede realizar cualquier tarea de administración**. En algunas distribuciones se puede acceder directamente al sistema como usuario root, pero en otras distribuciones, como en Ubuntu, te obligan a acceder al sistema como otro usuario y luego cambiar al usuario root.
- Si se desea ejecutar una tarea de forma puntual con permisos de administrador (el usuario debe de tenerlos), se puede utilizar el **comando sudo** de la siguiente forma:

```
$ sudo <comando>
```

- Si se necesita ejecutar muchas tareas como root se puede cambiar al usuario root ejecutando:

```
$ sudo -i
```

5.2.2 Gestor de paquetes

APT

- El software en los sistemas Linux se distribuye en forma de **paquetes**.
- Un paquete es un archivo comprimido que contiene todo lo necesario para instalar un software.
- Existen muchos tipos distintos de paquetes en Linux. La mayoría de ellos están asociados con gestores de paquetes de distribuciones Linux específicas.
- Siempre es preferible buscar las aplicaciones en el formato de paquete nativo de cada distribución. No obstante, existen métodos para convertir entre distintos formatos de paquetes.
- Los paquetes se pueden buscar e instalar a través de la web u otros soportes de información, o se pueden descargar desde un gestor de paquetes.
- Un gestor de paquetes es capaz de **buscar, actualizar, instalar y manejar automáticamente las dependencias** de estos.

5.2.2 Gestor de paquetes

APT

- Ubuntu es una distribución de Linux basada en Debian. Es decir, comparte los servicios, la API del sistema y muchas herramientas con este. Por ello utiliza su sistema de empaquetado.
- El sistema de empaquetado de Debian tiene una gran cantidad de información asociada a cada paquete para asegurar una integración sencilla y limpia en el sistema. La característica más importante es el sistema de dependencias.
- El sistema de dependencias **permite el uso de elementos compartidos** en el sistema por programas individuales, como **librerías**.
- Una dependencia simple es que un paquete requiere que otro también esté instalado para poder funcionar correctamente.
- También **ocurren conflictos entre las dependencias**. Esto se produce cuando se instala un paquete junto a otro de forma que la combinación los hace inservibles. Incluso llegando a corromper el sistema. La razón de por qué ocurre esto es porque ambos paquetes utilizan el mismo **recurso** y entran en conflicto.

5.2.2 Gestor de paquetes

APT

- APT ofrece una forma sencilla de instalar paquetes a través de repositorios externos.
 - Lo primero que se hace es obtener las listas actualizadas de los paquetes de los repositorios.

```
sudo apt update
```

- **sudo apt upgrade:** actualiza todos los paquetes mostrando un listado de los que no ha podido actualizar.
- **sudo apt cache-search <nombre>:** se utiliza para buscar paquetes por el nombre en la lista local.
- **sudo apt install: <nombre>** se utiliza para instalar paquetes por el nombre.
- **sudo apt dist-upgrade:** actualiza el sistema operativo entre versiones. Utilizar con moderación.
- **sudo apt remove <nombre>:** elimina paquetes.

5.2.3 Instalación manual de paquetes

dpkg

- También se puede instalar manualmente paquetes descargados o importados.
- Los paquetes de las distribuciones basadas en debian poseen la extensión **.deb**.

```
dpkg -i <nombre_del_paquete>.deb
```

- Para desinstalar un archivo deb, se elimina con **apt remove**.

5.2.4 Estructura de directorios

Ubuntu

- Los sistemas de ficheros de Linux y Unix se organizan en una estructura **jerárquica**, de tipo árbol. **El nivel más alto** del sistema de ficheros es **/ o directorio raíz**.
- Por debajo del directorio raíz hay un importante grupo de directorios común a la mayoría de las distribuciones GNU/Linux:
 - **/bin** - aplicaciones binarias importantes.
 - **/boot** - ficheros de configuración y necesarios para el arranque, además de los núcleos.
 - **/dev** - los ficheros de dispositivo.
 - **/etc** - los ficheros de configuración y arranque de servicios.
 - **/home** - directorios personales de los usuarios.

5.2.4 Estructura de directorios

Ubuntu

- **/initrd** - usado cuando se crea un proceso de arranque personalizado.
- **/lib** - librerías del sistema
- **/lost+found** - ficheros que no se pueden asociar después de fallos del sistema.
- **/media** - particiones y medios montados automáticamente.
- **/mnt** - particiones montadas manualmente en el disco duro.
- **/opt** - directorio donde instalar aplicaciones de terceros.
- **/proc** - directorio dinámico que mantiene información de los procesos en ejecución actualmente.
- **/root** - directorio personal del usuario root

5.2.4 Estructura de directorios

Ubuntu

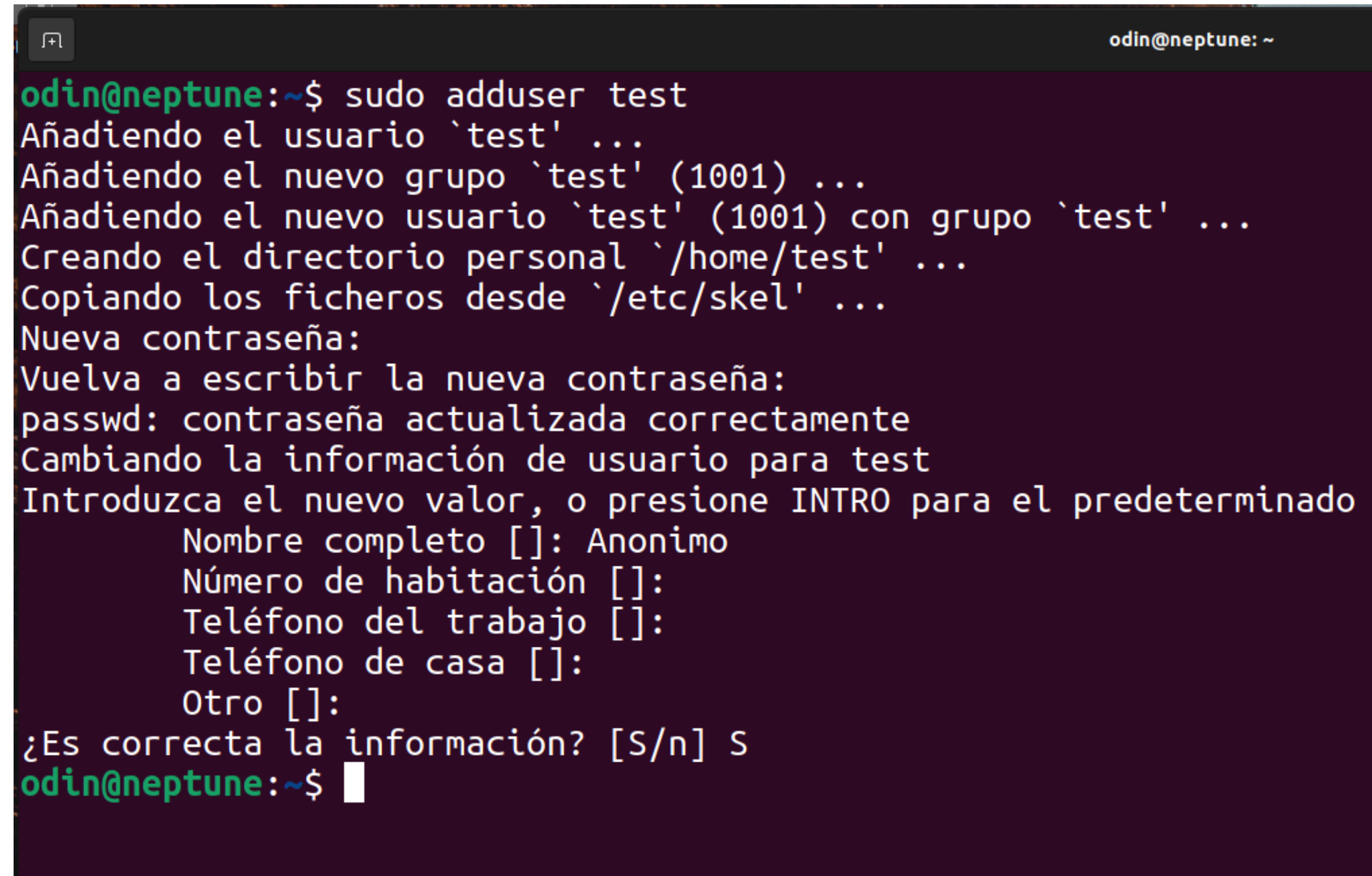
- **/sbin** - binarios importantes del sistema.
- **/srv** - puede contener archivos que se sirven mediante servicios a otro sistemas.
- **/sys** - archivos del sistema.
- **/tmp** - temporal files.
- **/usr** - aplicaciones y archivos a los que puede acceder la mayoría de usuarios.
- **/var** - archivos variables como archivos de registros y bases de datos.

5.2.5 Usuarios y grupos

Usuarios

- Para crear un usuario:

```
adduser <nombre_de_usuario>
```

A terminal window with a dark background and light-colored text. The prompt is 'odin@neptune: ~'. The user enters 'sudo adduser test'. The terminal shows the following output: 'Añadiendo el usuario `test' ...', 'Añadiendo el nuevo grupo `test` (1001) ...', 'Añadiendo el nuevo usuario `test` (1001) con grupo `test` ...', 'Creando el directorio personal `/home/test` ...', 'Copiando los ficheros desde `/etc/skel` ...', 'Nueva contraseña:', 'Vuelva a escribir la nueva contraseña:', 'passwd: contraseña actualizada correctamente', 'Cambiando la información de usuario para test', 'Introduzca el nuevo valor, o presione INTRO para el predeterminado'. Then it asks for 'Nombre completo []: Anonimo', 'Número de habitación []:', 'Teléfono del trabajo []:', 'Teléfono de casa []:', 'Otro []:', and '¿Es correcta la información? [S/n] S'. Finally, it shows the prompt 'odin@neptune:~\$' with a cursor.

```
odin@neptune:~$ sudo adduser test
Añadiendo el usuario `test' ...
Añadiendo el nuevo grupo `test' (1001) ...
Añadiendo el nuevo usuario `test' (1001) con grupo `test' ...
Creando el directorio personal `/home/test' ...
Copiando los ficheros desde `/etc/skel' ...
Nueva contraseña:
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
Cambiando la información de usuario para test
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []: Anonimo
Número de habitación []:
Teléfono del trabajo []:
Teléfono de casa []:
Otro []:
¿Es correcta la información? [S/n] S
odin@neptune:~$
```

5.2.5 Usuarios y grupos

Usuarios

- Para eliminar el usuario:

```
userdel -r <nombre_de_usuario>
```

- Para cambiar la contraseña:

```
passwd <nombre_de_usuario>
```

- Para modificar el usuario:

```
usermod [opciones] <nombre_de_usuario>
```

5.2.5 Usuarios y grupos

Usuarios

```
odin@neptune: ~  
odin@neptune:~$ usermod --help  
Modo de uso: usermod [opciones] USUARIO  
  
Opciones:  
-b, --badnames          allow bad names  
-c, --comment COMENTARIO nuevo valor del campo GECOS  
-d, --home DIR_PERSONAL nuevo directorio personal del nuevo usuario  
-e, --expiredate FECHA_EXPIR establece la fecha de caducidad de la  
                             cuenta a FECHA_EXPIR  
-f, --inactive INACTIVO establece el tiempo de inactividad después  
                             de que caduque la cuenta a INACTIVO  
-g, --gid GRUPO          fuerza el uso de GRUPO para la nueva cuenta  
                             de usuario  
-G, --groups GRUPOS      lista de grupos suplementarios  
-a, --append             append the user to the supplemental GROUPS  
                             mentioned by the -G option without removing  
                             the user from other groups  
-h, --help              muestra este mensaje de ayuda y termina  
-l, --login NOMBRE       nuevo nombre para el usuario  
-L, --lock              bloquea la cuenta de usuario  
-m, --move-home          mueve los contenidos del directorio  
                             personal al directorio nuevo (usar sólo  
                             junto con -d)  
-o, --non-unique         permite usar UID duplicados (no únicos)  
-p, --password CONTRASEÑA usar la contraseña cifrada para la nueva cuenta  
-R, --root CHROOT_DIR    directorio en el que hacer chroot  
-P, --prefix PREFIX_DIR  prefix directory where are located the /etc/* files  
-s, --shell CONSOLA      nueva consola de acceso para la cuenta del  
                             usuario  
-u, --uid UID            fuerza el uso del UID para la nueva cuenta  
                             de usuario  
-U, --unlock             desbloquea la cuenta de usuario  
-v, --add-subuids FIRST-LAST add range of subordinate uids  
-V, --del-subuids FIRST-LAST remove range of subordinate uids  
-w, --add-subgids FIRST-LAST add range of subordinate gids  
-W, --del-subgids FIRST-LAST remove range of subordinate gids  
-Z, --selinux-user SEUSER new SELinux user mapping for the user account
```


5.2.5 Usuarios y grupos

Usuarios

- Podemos listar todos los usuarios del sistema con:

```
cat /etc/passwd
```

- Al listarlos, observaremos que no sólo aparecen usuarios normales, sino que también aparecen usuarios del sistema. Estos usuarios se encargan de gestionar los servicios del mismo. Y se diferencian en que no tienen un shell asignado ni tampoco pueden hacer login.

```
odin@neptune:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

5.2.5 Usuarios y grupos

Usuarios

- La información de usuario que aparece es la siguiente:

```
odin@neptune:~$ cat /etc/passwd | grep odin
odin:x:1000:1000:odin,,,:/home/odin:/bin/bash
```

- nombre de usuario : contraseña cifrada en /etc/shadow : UID : GID : información adicional: home del usuario : shell
- Los usuarios del sistema no tienen shell y aparecen con **/usr/sbin/nologin** o **/bin/false**, por ellos podemos filtrar la búsqueda por esos criterios:

```
odin@neptune:~$ cat /etc/passwd | grep -v /bin/false | grep -v /usr/sbin/nologin
root:x:0:0:root:/root:/bin/bash
sync:x:4:65534:sync:/bin:/bin/sync
odin:x:1000:1000:odin,,,:/home/odin:/bin/bash
test:x:1001:1001:Anonimo,,,:/home/test:/bin/bash
```


5.2.5 Usuarios y grupos

Usuarios

- Podemos mostrar solamente los campos que nos interesan. Por ejemplo, si nos interesa mostrar los campos uno y seis podemos utilizar el comando **cut**:

```
cut -f1,6 -d: /etc/passwd
```

- En este caso la opción **-f** sirve para indicar los campos que queremos y con la opción **-d** indicamos el separador de los campos.

5.2.5 Usuarios y grupos

Grupos

- Podemos agrupar permisos de usuario por grupos de usuario. De esta forma podemos manipular más fácilmente un conjunto de usuarios del sistema.
- Seguro que en más de una ocasión has utilizado ya la herramienta **sudo** (**super user do**), que te permite ejecutar aplicaciones como si fueses usuario administrador. Para poder utilizarla, debes pertenecer al **grupo sudo**.
- También podemos asignar permisos de lectura a un grupo de usuarios sobre unos directorios y a otro, de lectura y escritura. De esta forma se puede mantener un control sobre los mismos.
- Para conocer toda la información relativa a grupos que hay en el sistema tienes que echar un vistazo al fichero **/etc/group**.

5.2.5 Usuarios y grupos

Grupos

```
sambashare:x:129:lorenzo,pepe,juan
```

- nombre del grupo : contraseña cifrada : GID : listado de usuarios pertenecientes al grupo

- Para crear un grupo de usuarios utilizaremos el comando:

```
groupadd <nombreDelGrupo>
```

- Para borrar un grupo lo haremos con:

```
groupdel <nombreDelGrupo>
```

- Hay que llevar **especial cuidado al borrar grupos**, ya que si se borra algún grupo de usuarios del sistema, se puede romper el sistema.

5.2.5 Usuarios y grupos

Grupos

- Para conocer a qué grupos pertenece un usuario se puede hacer cómodamente con:

```
groups <nombreDeUsuario>
```

- Los usuarios pertenecen a dos tipos de grupos:
 - **El grupo primario** es el que asigna el sistema operativo a los ficheros que crea el usuario. Este grupo primario normalmente, tiene el mismo nombre que el usuario.
 - **Grupos secundarios.** Constan de uno o más grupos a los cuales el usuario pertenece. Los usuarios pueden pertenecer hasta 15 grupos de usuarios.

```
odin@neptune:~$ groups odin
odin : odin adm cdrom sudo dip plugdev lpadmin lxd sambashare vboxsf
```

- usuario : grupo primario + resto de grupos

5.2.5 Usuarios y grupos

Grupos

- Para añadir un usuario a un grupo de usuarios utilizaremos:

```
usermod -aG <grupos> <usuario>
```

- **-a** indica que vamos a añadir grupos secundarios a un usuario y **-G** sirve para definirlos. Si queremos añadirlo a más de un grupo los separamos por comas.
- Si queremos borrar a un usuario de un grupo:

```
deluser <usuario> <grupo>
```

5.2.6 Permisos

En Linux se establece para cada directorio y fichero:

1. El propietario del archivo.
2. Quién es el grupo propietario.
3. Qué permisos tiene el propietario.
4. Los permisos que tiene el grupo propietario del archivo.
5. Los permisos del resto de usuarios.

5.2.6 Permisos

Para comprobar los permisos que tiene un archivo o directorio se puede utilizar:

```
ls -l o ls -lah
```

La diferencia entre los dos comandos, es que el segundo muestra también los archivos ocultos (comienzan con un .), así como los tamaños de los ficheros y directorios.

```
odin@neptune:~$ ls -lah
total 92K
drwxr-x--- 15 odin odin 4,0K mar 22 10:48 .
drwxr-xr-x  4 root root 4,0K mar  8 20:23 ..
-rw-----  1 odin odin  510 mar  8 20:53 .bash_history
-rw-r--r--  1 odin odin  220 feb 28 10:59 .bash_logout
-rw-r--r--  1 odin odin 3,7K feb 28 10:59 .bashrc
```

5.2.6 Permisos

En Linux **todo es un fichero**. Así, un directorio es un fichero, un periférico es un fichero, etc.

Hay tres tipos de ficheros:

- Los **ficheros normales** que son los que contienen texto plano o datos binarios.
- Los **directorios**, que tienen el mismo concepto que en cualquier otro sistema operativo.
- Los **especiales**, que se dividen en:
 - **Ficheros de bloque**. Estos hacen referencia a los dispositivos físicos. Se encuentran en **/dev** y manipulan **E/S en bloques de datos**.
 - **Los ficheros de carácter** manipulan **E/S byte a byte**. Por ejemplo la E/S del terminal, o **/dev/null** que sirve para desechar la salida de datos.
 - **Ficheros de tubería o pipe files**. Es similar al concepto de pipe (la entrada de un comando es la salida de otro) pero en este caso se genera un fichero que se puede acceder a él.
 - **Enlaces simbólicos**. Es un puntero que señala a otro archivo o directorio que funciona como un acceso directo vitaminado ya que permite acceder a la posición de memoria real donde está la información. Lo que nos permite manipular la información desde ubicaciones diferentes, expandir nuestro sistema de ficheros etc.
 - **Archivos de tipo socket**. Nos permiten pasar información entre diferentes aplicaciones.

5.2.6 Permisos

```
drwxr-xr-x  4 root root 4,0K mar  8 20:23 ..
-rw-----  1 odin odin  510 mar  8 20:53 .bash_history
```

La salida del comando ls -lah nos informa de:

-	r	w	x	r	-	x	r	-	x
Tipo de fichero	Permisos para el dueño del fichero			Permisos para el grupo al que pertenece el usuario			Permisos para el resto de usuarios		

r	Permiso de lectura
w	Permiso de escritura
x	Permiso de ejecución

5.2.6 Permisos

Para cambiar los permisos de un fichero en linux utilizaremos el comando:

```
chmod [opciones] <permisos> <fichero>
```

Donde en los permisos:

- 4 significa lectura.
- 2 significa escritura.
- 1 significa ejecución.
- 0 sin permiso.
- <permisos> es una combinación de 3 números, dónde cada número es una suma de los permisos para su correspondiente ámbito (dueño, grupo del dueño, resto de usuarios).

5.2.6 Permisos

Para cambiar los propietarios de un fichero en Linux usaremos el comando:

```
chown [opciones] <usuario:grupo> <fichero>
```

Donde:

- <usuario:grupo> es el nombre de usuario y/o grupo o los UID o GID respectivamente.