



# BIG DATA

**2025/26**

Ciclo	Especialización IA & BIG DATA
Nombre	Rodrigo Medina
Correo	YMQ06518@educastur.es
Nº Unidad Didáctica	01

# Tabla de Contenido

Tabla de Contenido .....	1
2. Obtener Ayuda y Localizar Archivos .....	4
3. Navegación y Listado de Archivos .....	6
4. Manipulación de Archivos y Directorios .....	8
5. Archivado y Compresión .....	10
6. Redirección, Tuberías y Filtros .....	14
7. Scripts Básicos .....	17
8. Ejercicios Avanzados .....	20

# 1. Conceptos Básicos y Variables

1. Muestra el contenido de tu variable de entorno `cd` junto con esa variable para navegar a dicho directorio y verifica con `HOME` . Luego, usa `pwd` en la ubicación correcta.

```
echo $HOME
cd &HOME
pwd
```

```
rm@servidor:~$ echo $HOME
/home/rm
rm@servidor:~$ cd $HOME
rm@servidor:~$ pwd
/home/rm
```

2. Ejecuta el comando `pwd` que te `whoami` . Ahora, crea una variable local llamada `USUARIO_ACTUAL` que contenga el resultado del comando anterior y muéstrala en la terminal.

```
whoami
USUARIO_ACTUAL=`whoami`
echo $USUARIO_ACTUAL
```

```
rm@servidor:~$ whoami
rm
rm@servidor:~$ USUARIO_ACTUAL=`whoami`
rm@servidor:~$ echo $USUARIO_ACTUAL
rm
rm@servidor:~$ _
```

3. Intenta crear un archivo llamado `dos palabras.txt` sin usar comillas. Observa el resultado con `ls` . ¿Qué ha ocurrido y por qué? Ahora, bórralo(s) y créalo correctamente.

```
touch dos palabras.txt
rm dos palabras.txt
```

```
rm@servidor:~$ touch dos palabras.txt
rm@servidor:~$ ls
dos palabras.txt
rm@servidor:~$ rm dos palabras.txt
rm@servidor:~$ touch dos_palabras.txt
rm@servidor:~$ ls
dos_palabras.txt
rm@servidor:~$
```

4. Usa el comando `type` para averiguar si `ls` y `cd` son internos o externos al shell. ¿Qué diferencia práctica crees que implica esto?

```
type ls cd
```

```
rm@servidor:~$ type ls cd
ls is aliased to `ls --color=auto'
cd is a shell builtin
```

5. Muestra tu `PATH` actual. Crea un directorio `~/mi_bin` y añádelo temporalmente al principio de tu `PATH`. Verifica que el cambio se ha realizado correctamente.

```
echo $PATH
mkdir mi_bin
PATH=~/mi_bin:$PATH
```

```
echo $PATH
```

```
rm@servidor:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
rm@servidor:~$ mkdir mi_bin
rm@servidor:~$ PATH=~/mi_bin:$PATH
rm@servidor:~$ echo $PATH
/home/rm/mi_bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
rm@servidor:~$ _
```

## 2. Obtener Ayuda y Localizar Archivos

1. Abre la página del manual para el comando `chmod` . ¿En qué sección del manual se encuentra? ¿Qué indica ese número de sección sobre el tipo de comando?

```
man chmod
```

```
CHMOD(1) User Commands
NAME
  chmod - change file mode bits
SYNOPSIS
  chmod [OPTION]... MODE[,MODE]... FILE...
  chmod [OPTION]... OCTAL-MODE FILE...
  chmod [OPTION]... --reference=RFILE FILE...
```

2. Usando la función de búsqueda dentro de la página del manual de encuentra la opción que ordena los archivos por tamaño.

```
man ls
/size
```

```
rm@servidor:~$ man ls
```

```
-S      sort by file size, largest first
```

3. Imagina que has olvidado dónde se guarda el archivo de configuración de usuarios. Sabiendo que se llama `passwd` , usa `find` para buscarlo desde el directorio raíz (/). Anota la ruta completa que has encontrado.

```
find / -name passwd
```

```
rm@servidor:~$ find / -name passwd
```

4. Crea un archivo vacío llamado `test_locate.txt` en tu directorio home. Inmediatamente después, búscalo con `locate` . ¿Aparece en los resultados? ¿Por qué sí o por qué no?

No aparece inmediatamente después porque la base de datos no se ha actualizado aún. (Hay que instalar `plocate`)

```
touch test_locate.txt
locate test_locate.txt
```

```
rm@servidor:~$ touch test_locate.txt
rm@servidor:~$ locate test_locate.txt
rm@servidor:~$
```

5. Basado en el ejercicio anterior, ¿qué comando (probablemente con `sudo` ) necesitas ejecutar para que `locate` sí encuentre tu archivo? Ejecútalo y verifica que ahora sí lo encuentras.

```
sudo updatedb
```

```
rm@servidor:~$ sudo updatedb
rm@servidor:~$ locate test_locate.txt
/home/rm/test_locate.txt
```

### 3. Navegación y Listado de Archivos

1. Navega al directorio `/etc` . Desde ahí, sin usar directorio `cd` , lista el contenido de tu home usando una ruta con el atajo `~` .

```
cd /etc
ls ~
```

```
rm@servidor:~$ cd /etc
rm@servidor:/etc$ ls ~
mi_bin  test_locate.txt
rm@servidor:/etc$ _
```

2. Desde tu directorio home , navega a `/` y luego a `var` y finalmente a `log` usando una sola línea de comando y rutas relativas.

```
cd ../../var/log
```

```
rm@servidor:~$ cd ../../var/log
rm@servidor:/var/log$ _
```

3. Lista el contenido de `/etc` en formato largo. En la salida, identifica el propietario, el grupo y los permisos del archivo `passwd` .

```
ls -l /etc/passwd
```

```
rm@servidor:/var/log$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1725 oct 15 16:53 /etc/passwd
```

**Propietario:** root

**Grupo:** root

**Permisos:** (usuario) rw ; (grupo) r; (otros) r

4. Compara la salida de `ls -l /etc` y `ls -lh /etc` . ¿Qué hace la opción útil para las personas?

```
rm@servidor:/var/log$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1725 oct 15 16:53 /etc/passwd
rm@servidor:/var/log$ ls -lh /etc/passwd
-rw-r--r-- 1 root root 1,7K oct 15 16:53 /etc/passwd
```

A través de la opción `-h` (**human readable**) se muestra el peso del archivo de forma mas entendible para una persona

**5. Ejecuta `ls -R ~` . ¿Qué hace la opción `h` y por qué es `R` ? ¿Por qué podría ser peligroso usarla en el directorio raíz (`/`)?**

Muestra un listado, de forma recursiva, del contenido de todos los directorios y subdirectorios desde donde se realiza la búsqueda.

Si se realiza una búsqueda de este tipo desde raíz (`ls -R /`) realizará un listado del contenido de todos los directorios del sistema.

```
rm@servidor:/var/log$ ls -R ~  
/home/rm:  
mi_bin  test_locate.txt  
  
/home/rm/mi_bin:  
rm@servidor:/var/log$ _
```



## 4. Manipulación de Archivos y Directorios

1. Crea la estructura de directorios `proyecto/src` , `proyecto/doc` y un único comando `mkdir` .

```
mkdir proyectos proyectos/src proyectos/doc proyectos/bin
```

```
rm@servidor:~$ mkdir proyectos proyectos/src proyectos/doc proyectos/bin
rm@servidor:~$ ls -R
.:
mi_bin  proyectos  test_locate.txt

./mi_bin:

./proyectos:
bin  doc  src

./proyectos/bin:

./proyectos/doc:

./proyectos/src:
rm@servidor:~$
```

2. Crea un archivo `~/notas.txt` . Muévelo a `proyecto/bin` usando `~/proyecto/doc` y, en el mismo comando, renómbalo a `README.md` .

```
touch notas.txt
mv notas.txt proyectos/doc/README.md
```

```
rm@servidor:~$ touch notas.txt
rm@servidor:~$ mv notas.txt proyectos/doc/README.md
rm@servidor:~$ ls -R
.:
mi_bin  proyectos  test_locate.txt

./mi_bin:

./proyectos:
bin  doc  src

./proyectos/bin:

./proyectos/doc:
README.md

./proyectos/src:
```

3. Copia el archivo README.md de proyecto/doc a proyecto/bin . Luego, borra el archivo original de la carpeta doc .

```
cp proyectos/doc/README.md proyectos/bin
rm proyectos/doc/README.md
```

```
rm@servidor:~$ cp proyectos/doc/README.md proyectos/bin/
rm@servidor:~$ rm proyectos/doc/README.md
rm@servidor:~$ ls -R
.:
mi_bin  proyectos  test_locate.txt

./mi_bin:

./proyectos:
bin  doc  src

./proyectos/bin:
README.md

./proyectos/doc:

./proyectos/src:
```

4. Intenta borrar el directorio proyecto con usa rmdir . ¿Qué error obtienes? Ahora, rm con la opción correcta para borrar el directorio y todo lo que contiene.

Con `rmdir` no se puede borrar directorios que no estén vacíos. Para ellos hay que usar `rm -r`

```
rm@servidor:~$ rmdir proyectos/
rmdir: failed to remove 'proyectos/': Directory not empty
rm@servidor:~$ rm -r proyectos/
rm@servidor:~$ ls -R
.:
mi_bin  test_locate.txt

./mi_bin:
```

5. Navega a /etc . Usando un solo comando ls con globbing , lista todos los archivos que empiecen con la letra s y terminen con .conf .

```
ls s*.conf
```

```
rm@servidor:~$ cd /etc
rm@servidor:/etc$ ls s*.conf
sensors3.conf  sudo.conf  sudo_logsrvd.conf  sysctl.conf
```

## 5. Archivado y Compresión

1. Crea un archivo tar llamado `log_backup.tar` que contenga todos los archivos del directorio `/var/log` . ¿Qué advertencias de “permiso denegado” aparecen y por qué?

```
tar cf log_backup.tar /var/log
```

```
rm@servidor:~$ tar cf log_backup.tar /var/log
tar: Removing leading `/' from member names
tar: /var/log/btmp: Cannot open: Permission denied
tar: /var/log/private: Cannot open: Permission denied
tar: /var/log/installer/subiquity-server-debug.log.1503: Cannot open: Permission denied
tar: /var/log/installer/installer-journal.txt: Cannot open: Permission denied
tar: /var/log/installer/autoinstall-user-data: Cannot open: Permission denied
tar: /var/log/installer/cloud-init-output.log: Cannot open: Permission denied
tar: /var/log/installer/cloud-init.log: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-curtin-apt.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-extract.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-cunthooks.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-partitioning.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-initial.conf: Cannot open: Permission denied
tar: /var/log/installer/subiquity-client-info.log.1447: Cannot open: Permission denied
tar: /var/log/installer/subiquity-server-info.log.1503: Cannot open: Permission denied
tar: /var/log/installer/subiquity-client-debug.log.1447: Cannot open: Permission denied
tar: Exiting with failure status due to previous errors
rm@servidor:~$ _
```

Se necesita permisos de administrador (`sudo`) para acceder a ciertos archivos

2. Comprime el archivo `log_backup.tar` con original y el comprimido usando `gzip` . Compara el tamaño del archivo `ls -lh` .

```
tar czf log_backup.tar.gz /var/log
ls -lh
```

```
rm@servidor:~$ tar czf log_backup.tar.gz /var/log
tar: Removing leading `/' from member names
tar: /var/log/btmp: Cannot open: Permission denied
tar: /var/log/private: Cannot open: Permission denied
tar: /var/log/installer/subiquity-server-debug.log.1503: Cannot open: Permission denied
tar: /var/log/installer/installer-journal.txt: Cannot open: Permission denied
tar: /var/log/installer/autoinstall-user-data: Cannot open: Permission denied
tar: /var/log/installer/cloud-init-output.log: Cannot open: Permission denied
tar: /var/log/installer/cloud-init.log: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-curtin-apt.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-extract.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-cunthooks.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-partitioning.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-initial.conf: Cannot open: Permission denied
tar: /var/log/installer/subiquity-client-info.log.1447: Cannot open: Permission denied
tar: /var/log/installer/subiquity-server-info.log.1503: Cannot open: Permission denied
tar: /var/log/installer/subiquity-client-debug.log.1447: Cannot open: Permission denied
tar: Exiting with failure status due to previous errors
rm@servidor:~$ ls -lh
total 165M
-rw-rw-r-- 1 rm rm 163M oct 18 17:36 log_backup.tar
-rw-rw-r-- 1 rm rm 2,2M oct 18 17:41 log_backup.tar.gz
drwxrwxr-x 2 rm rm 4,0K oct 17 17:34 mi_bin
-rw-rw-r-- 1 rm rm 0 oct 16 17:58 test_locate.txt
rm@servidor:~$ _
```

### 3. Lista el contenido del archivo log\_backup.tar.gz sin extraerlo para verificar que los archivos están dentro.

```
tar tf log_backup.tar.gz
```

```
rm@servidor:~$ tar tf log_backup.tar.gz _
```

```
var/log/apt/
var/log/apt/history.log
var/log/apt/eipp.log.xz
var/log/apt/term.log
var/log/cloud-init.log
var/log/alternatives.log
var/log/faillog
var/log/dmccg.3.gz
var/log/installer/
var/log/installer/subiquity-server-debug.log
var/log/installer/subiquity-server-info.log
var/log/installer/curtin-install.log
var/log/installer/casper-md5check.json
var/log/installer/media-info
var/log/installer/device-map.json
var/log/installer/block/
var/log/installer/block/discover.log
var/log/installer/block/probe-data.json
var/log/installer/curtin-install/
var/log/installer/subiquity-client-info.log
var/log/installer/subiquity-client-debug.log
var/log/journal/
var/log/journal/361e0a35958145e696e416a7ffc1c89f/
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@0006414903466dfd-906aaaa33ae6124c.journal~
var/log/journal/361e0a35958145e696e416a7ffc1c89f/user-100001c7fb1936b74e3c9699c869d509ba05-00000000000152e-000641491d5ddfe9.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/user-100004d2e81ff0bc046eca277d9c057c5add3-000000000001d81-0006416bc038590c.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/user-10000000641491d5de592-5ea43e929c8f66c8.journal~
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@0006416bbf06bc45-13dc85c6c33c3a33.journal~
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@1c7fb1936b74e3c9699c869d509ba05-0000000000011b8-000641491c648e34.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/user-1000052aa2bd0bc2e44dd96449c5e6f691362-000000000000da2-00064147e553bc96.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@723e8903679843e18a178d5dae79ad8d-000000000000e18-00064149033ed022.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@00064147e45bc392-9c3c9dcaef8a073e.journal~
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@000641491c8b6f09-77f88ab1310951e9.journal~
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@0514e7935fea4a53878775e3bb1a0bb1-0000000000015e7-0006415dabaa0a1b.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@0c1ae60c33d36485a89e325ac1b5c8d2c-00000000000022e4-00064185003d84fc.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/user-100000006413562cf3916-07f356baf735271b.journal~
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@96f9c7fc0f9d49d49e064252c54fc4c5-000000000001dd6-00064171d9a505e0.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@4d2e81ff0bc046eca277d9c057c5add3-000000000001a03-0006416bbf009561.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@00064185004337d2-67beb6d5aee36636.journal~
var/log/journal/361e0a35958145e696e416a7ffc1c89f/user-1000.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/user-1000096f9c7fc0f9d49d49e064252c54fc4c5-00000000000214f-00064171da82f5a7.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@4d2e81ff0bc046eca277d9c057c5add3-000000000001d82-0006416bc03a8d39.journal
var/log/journal/361e0a35958145e696e416a7ffc1c89f/user-1000000064147e553c403-09fe66b4d1f5aac3.journal~
var/log/journal/361e0a35958145e696e416a7ffc1c89f/system@0006413561ce0bee-0b0961e57c0bab8e.journal~
var/log/journal/361e0a35958145e696e416a7ffc1c89f/user-100000514e7935fea4a53878775e3bb1a0bb1-00000000000196a-0006415dad4b1ec3.journal
var/log/auth.log
var/log/syslog
var/log/dmccg.0
rm@servidor:~$
```

### 4. Extrae únicamente el archivo syslog (o messages) de log\_backup.tar.gz a tu directorio /tmp .

```
tar xf log_backup.tar.gz -C /tmp var/log/syslog
```

```
rm@servidor:~$ tar xf log_backup.tar.gz -C /tmp/ var/log/syslog
rm@servidor:~$ ls -R /tmp/var/
/tmp/var/:
log

/tmp/var/log:
syslog
rm@servidor:~$ _
```

## 5. Crea tres archivos ( a.txt , b.log , c.jpg ) y luego crea un archivo contenga.

```
touch a.txt b.log c.jpg
zip comprimidos.zip a.txt b.log c.jpg
```

```
rm@servidor:~$ touch a.txt b.log c.jpg
rm@servidor:~$ zip comprimidos.zip a.txt b.log c.jpg
  adding: a.txt (stored 0%)
  adding: b.log (stored 0%)
  adding: c.jpg (stored 0%)
rm@servidor:~$ ls -l
total 193872
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 a.txt
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 b.log
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 c.jpg
-rw-rw-r-- 1 rm rm    436 oct 19 17:30 comprimidos.zip
-rw-rw-r-- 1 rm rm 196065280 oct 19 16:24 log_backup.tar
-rw-rw-r-- 1 rm rm 2448876 oct 19 16:25 log_backup.tar.gz
drwxrwxr-x 2 rm rm    4096 oct 17 17:34 mi_bin
-rw-rw-r-- 1 rm rm      0 oct 16 17:58 test_locate.txt
rm@servidor:~$ _
```

## 6. Elimina los tres archivos originales y luego recupéralos desde el archivo

```
rm a.txt b.log c.jpg
unzip comprimidos.zip
```

```
rm@servidor:~$ rm a.txt b.log c.jpg
rm@servidor:~$ unzip comprimidos.zip
Archive:  comprimidos.zip
  extracting: a.txt
  extracting: b.log
  extracting: c.jpg
rm@servidor:~$ ls -l
total 193872
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 a.txt
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 b.log
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 c.jpg
-rw-rw-r-- 1 rm rm    436 oct 19 17:30 comprimidos.zip
-rw-rw-r-- 1 rm rm 196065280 oct 19 16:24 log_backup.tar
-rw-rw-r-- 1 rm rm 2448876 oct 19 16:25 log_backup.tar.gz
drwxrwxr-x 2 rm rm    4096 oct 17 17:34 mi_bin
-rw-rw-r-- 1 rm rm      0 oct 16 17:58 test_locate.txt
rm@servidor:~$
```

7. Usa `zcat` (o `gzcat`) para leer el contenido de un archivo de log comprimido (ej: en `/var/log`, busca uno que termine en descomprimido).

```
ls -l /var/log/*.gz  
zcat archivo.gz
```

```
rm@servidor:~$ ls -l /var/log/*.gz  
-rw-r----- 1 root adm 15751 oct 18 17:05 /var/log/dmesg.1.gz  
-rw-r----- 1 root adm 15824 oct 18 09:48 /var/log/dmesg.2.gz  
-rw-r----- 1 root adm 15880 oct 17 17:00 /var/log/dmesg.3.gz  
-rw-r----- 1 root adm 15764 oct 16 16:28 /var/log/dmesg.4.gz  
rm@servidor:~$ zcat /var/log/dmesg.1.gz _
```

## 6.Redirección, Tuberías y Filtros

1. Guarda la lista de archivos de tu directorio home (formato largo) en un archivo `mis_archivos.txt` .

```
ls -l ~ > mis_archivos.txt
```

```
rm@servidor:~$ ls -l ~ > mis_archivos.txt
rm@servidor:~$ cat mis_archivos.txt
total 193872
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 a.txt
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 b.log
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 c.jpg
-rw-rw-r-- 1 rm rm    436 oct 19 17:30 comprimidos.zip
-rw-rw-r-- 1 rm rm 196065280 oct 19 16:24 log_backup.tar
-rw-rw-r-- 1 rm rm  2448876 oct 19 16:25 log_backup.tar.gz
drwxrwxr-x 2 rm rm    4096 oct 17 17:34 mi_bin
-rw-rw-r-- 1 rm rm      0 oct 19 18:00 mis_archivos.txt
-rw-rw-r-- 1 rm rm      0 oct 16 17:58 test_locate.txt
rm@servidor:~$
```

2. Sin borrar el contenido anterior, añade la fecha y hora actual al final del archivo `mis_archivos.txt` .

```
date >> mis_archivos.txt
```

```
rm@servidor:~$ date >> mis_archivos.txt
rm@servidor:~$ cat mis_archivos.txt
total 193872
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 a.txt
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 b.log
-rw-rw-r-- 1 rm rm      0 oct 19 17:30 c.jpg
-rw-rw-r-- 1 rm rm    436 oct 19 17:30 comprimidos.zip
-rw-rw-r-- 1 rm rm 196065280 oct 19 16:24 log_backup.tar
-rw-rw-r-- 1 rm rm  2448876 oct 19 16:25 log_backup.tar.gz
drwxrwxr-x 2 rm rm    4096 oct 17 17:34 mi_bin
-rw-rw-r-- 1 rm rm      0 oct 19 18:00 mis_archivos.txt
-rw-rw-r-- 1 rm rm      0 oct 16 17:58 test_locate.txt
dom 19 oct 2025 18:02:35 UTC
rm@servidor:~$
```

3. Usa **grep** y una tubería (**|**) para contar el número de directorios que hay en **/etc** . (Pista: **ls -l | grep '^d'** )

```
ls -l /etc | grep ^d | wc -l
```

```
rm@servidor:~$ ls -l /etc | grep ^d | wc -l
106
```

4. Muestra las 10 últimas líneas del archivo **/etc/passwd** y, usando otra tubería, extrae solo los nombres de usuario (el primer campo).

```
cat /etc/passwd | tail | cut -f 1 -d ":"
```

```
rm@servidor:~$ cat /etc/passwd | tail | cut -f 1 -d ":"
pollinate
polkitd
syslog
uidd
tcpdump
tss
landscape
fwupd-refresh
usbmux
rm
```

5. Muestra una lista de todos los procesos del sistema ( **ps aux** ), ordénala por uso de CPU (tercera columna) y muestra solo las 5 líneas superiores.

```
ps aux | sort -k 3 -n | head -n 5
```

```
rm@servidor:~$ ps aux | sort -k 3 -n | head -n 5
message+  698  0.0  0.0  9780  5120 ?        Ss   15:56   0:00 @dbus-daemon --system --address=systemd:
nly
polkitd    707  0.0  0.0 308164  7680 ?        Ssl  15:56   0:00 /usr/lib/polkit-1/polkitd --no-debug
rm        1041  0.0  0.1  20812 11264 ?        Ss   15:56   0:00 /usr/lib/systemd/systemd --user
rm        1042  0.0  0.0  21148  3516 ?        S    15:56   0:00 (sd-pam)
rm        1052  0.0  0.0   8756  5504 tty1    S    15:56   0:00 -bash
```

6. ¿Cuál es la diferencia entre usar **>** y **>>** para redirigir la salida de un comando a un archivo? Demuéstralo con un ejemplo

- > Crear nuevo archivo de redirección y sobrescribe el contenido, en caso de que ya exista.
- >> Añade el nuevo contenido al final del documento, sin sobrescribir. Si no existe, también lo crea.



**7. Ejecuta `find /etc -name "*.conf"` . Redirige la salida estándar a un archivo `config_files.txt` y los errores (si los hay) a `errors.txt`**

```
find /etc -name *.conf > config_files.txt 2> errors.txt
```

```
rm@servidor:~$ find /etc -name *.conf > config_files.txt 2> errors.txt
rm@servidor:~$ ls -l
total 12
-rw-rw-r-- 1 rm rm 4965 oct 20 09:46 config_files.txt
-rw-rw-r-- 1 rm rm    0 oct 16 10:08 dos_palabras.txt
-rw-rw-r-- 1 rm rm  249 oct 20 09:46 errors.txt
```

## 7. Scripts Básicos

1. Crea un script que imprima tu nombre de usuario y el directorio de trabajo actual usando las variables de entorno correspondientes.

```
rm@servidor:~$ nano script.sh_
GNU nano 7.2
#!/bin/bash
echo "usuario: $USER | directorio: $PWD"

rm@servidor:~$ chmod +x script.sh
rm@servidor:~$ ./script.sh
usuario: rm | directorio: /home/rm
rm@servidor:~$
```

2. Haz el script anterior ejecutable solo para ti ( `chmod u+x ...` ) y ejecútalo. Luego, intenta ejecutarlo como otro usuario (si es posible) o explica qué pasaría.

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

```
rm@servidor:~$ chmod u+x script.sh
```

En caso de intentar ejecutarlo con otro usuario distinto al que ha sido creado, si los permisos de ejecución no están permitidos para otros usuarios, se mostraría permiso denegado y no se podrá ejecutar.

La opción `u+x` de `chmod` solo da permisos de ejecución al usuario que lo ha creado.

3. Modifica el script para que acepte un argumento. Si el argumento es “hola”, debe imprimir “mundo”. Si es cualquier otra cosa, no debe imprimir nada.

```
GNU nano 7.2
#!/bin/bash
argumento=$1

if [ $argumento == "hola" ]
then
    echo "mundo"
fi

rm@servidor:~$ ./script.sh hola
mundo
rm@servidor:~$ ./script.sh pepe
rm@servidor:~$
```

4. Mejora el script anterior para que, si no se proporciona ningún argumento, muestre un mensaje de uso: "Error: Debes proporcionar un argumento."

```
GNU nano 7.2
#!/bin/bash

argumento=$1

if [ $# -eq 0 ]
then
    echo "Error: Debes proporcionar un argumento"
elif [ $argumento == "hola" ]
then
    echo "mundo"
fi
```

```
rm@servidor:~$ ./script.sh
Error: Debes proporcionar un argumento
rm@servidor:~$ ./script.sh hola
mundo
rm@servidor:~$ ./script.sh pepe
rm@servidor:~$ _
```

5. Escribe un script que reciba dos números. Debe imprimir "iguales" si son iguales y "diferentes" si no lo son.

```
GNU nano 7.2
#!/bin/bash

argumento1=$1
argumento2=$2

if [ $argumento1 -eq $argumento2 ]
then
    echo "iguales"
else
    echo "diferentes"
fi
```

```
rm@servidor:~$ ./script2.sh 225 225
iguales
rm@servidor:~$ ./script2.sh 225 273
diferentes
rm@servidor:~$
```

6. Escribe un script que, dado un directorio como argumento, use un bucle for para iterar sobre su contenido ( ls \$1 ) y añada la extensión archivo.

```
GNU nano 7.2
#!/bin/bash

for file in $(ls $1)
do
    mv $1/$file $1/$file.archivo
done
```

```
rm@servidor:~$ mkdir dir
rm@servidor:~$ touch dir/pepe
rm@servidor:~$ chmod +x scriptExt.sh
rm@servidor:~$ ./scriptExt.sh dir
rm@servidor:~$ ls -l dir
total 0
-rw-rw-r-- 1 rm rm 0 oct 22 20:09 pepe.archivo
rm@servidor:~$ _
```

## 8. Ejercicios Avanzados

1. Muestra los shells de los usuarios listados en `/etc/passwd` , elimina las líneas duplicadas y ordénalos alfabéticamente. Pista: `cut` , `sort` , `uniq` )

```
cat /etc/passwd | cut -f 1 -d ":" | sort | uniq
```

```
rm@servidor:~$ cat /etc/passwd | cut -f 1 -d ":" | sort | uniq
_lapt
backup
bin
daemon
dhcpcd
fwupd-refresh
games
irc
landscape
list
lp
mail
man
messagebus
news
nobody
polkitd
pollinate
proxy
rm
root
sync
sys
syslog
systemd-network
systemd-resolve
systemd-timesync
tcpdump
tss
usbmux
uucp
uudd
www-data
rm@servidor:~$
```

2. Usando `ps` , `grep` y `wc` , crea un comando de una sola línea que te diga cuántos procesos está ejecutando el usuario `root` actualmente.

```
ps aux | cut -f 1 -d " " | grep root | wc -l
```

```
rm@servidor:~$ ps aux | cut -f 1 -d " " | grep root | wc -l
85
```

3. Lista todos los archivos en /etc , filtra los resultados para mostrar solo aquellos que han sido modificados en “Oct” (octubre) y guarda esa lista en october\_files.txt

```
ls -l /etc | grep oct > october_files.txt
```

```
rm@servidor:~$ ls -l /etc/ | grep oct > october_files.txt
rm@servidor:~$
```

4. Usando globbing , lista todos los archivos en en su nombre.

```
ls -l /etc/*[0-9]*
```

```
rm@servidor:~$ ls -l /etc/*[0-9]*
```

5. Usando find , busca en /usr/bin todos los archivos que sean ejecutables, pero que no sean propiedad del usuario root .

```
find /usr/bin -executable -not -user root
```

```
rm@servidor:~$ find /usr/bin/ -executable -not -user root
rm@servidor:~$ _
```

6. Compara la diferencia de tamaño y velocidad al comprimir un archivo grande (puedes usar /var/log/syslog ) con gzip y con bzip2

```
Ls -l /var/log/syslog/*
```

```
rm@servidor:~$ ls -l /var/log/syslog.*
-rw-r----- 1 syslog adm 75646 oct 18 17:42 /var/log/syslog.bz2
-rw-r----- 1 syslog adm 100026 oct 21 18:05 /var/log/syslog.gz
```

7. Crea un archivo tar de tu directorio home, pero esta vez, usa la opción para seguir enlaces simbólicos. Antes, crea un enlace simbólico en tu home para que puedas ver la diferencia.

```
ln -s /etc/passwd ~/enlace
tar -chf home.tar ~
```

```
rm@servidor:~$ ln -s /etc/passwd ~/enlace
rm@servidor:~$ tar -chf home.tar ~
tar: Removing leading `/' from member names
tar: /home/rm/home.tar: archive cannot contain itself; not dumped
tar: Removing leading `/' from hard link targets
tar: /home/rm/script.sh: Cannot open: Permission denied
tar: Exiting with failure status due to previous errors
```

8. Escribe un script que reciba una ruta a un archivo. Debe verificar si es un archivo regular, un directorio o si no existe, mostrando un mensaje diferente en cada caso. □Pista: `if [ -f ... ]`, `if [ -d ... ]` )

```
GNU nano 7.2
#!/bin/bash

ruta=$1

if [ -f $ruta ]
then
    echo "Es un archivo"
elif [ -d $ruta ]
then
    echo "Es un directorio"
else
    echo "No existe"
fi_
```

```
rm@servidor:~$ chmod +x script4.sh
rm@servidor:~$ ./script4.sh a.txt
Es un archivo
rm@servidor:~$ ./script4.sh p/
Es un directorio
rm@servidor:~$ ./script4.sh pepe
No existe
```

9. Crea un script que intente crear un directorio llamado el código de salida ( `test_dir` en `/`. Usando `$?` ), el script debe informar si tuvo éxito o si falló por un problema de permisos

```
GNU nano 7.2
#!/bin/bash

mkdir /test_dir

if [ $? -eq 0 ]
then
    echo "Directorio creado"
else
    echo "Error: no se pudo crear el directorio por permisos"
fi
```

```
rm@servidor:~$ chmod +x script5.sh
rm@servidor:~$ ./script5.sh
mkdir: cannot create directory '/test_dir': Permission denied
Error: no se pudo crear el directorio por permisos
rm@servidor:~$ sudo ./script5.sh
[sudo] password for rm:
Directorio creado
```

Al ejecutar el script, se intenta crear el directorio en la ruta raíz, por lo que se necesita permisos de superusuario. Si no se ejecuta con `sudo`, no se crea el directorio

10. Escribe un script que reciba cualquier número de argumentos. El script debe iterar sobre ellos y solo imprimir aquellos que sean números mayores que 10.

```
GNU nano 7.2
#!/bin/bash

for arg in $@
do
    if [ $arg -gt 10 ]
    then
        echo $arg
    fi
done
```

```
rm@servidor:~$ chmod +x script6.sh
rm@servidor:~$ ./script6.sh 2 3 4 5 10 12 15 22 853
12
15
22
853
```