

PR_01.1

1- Conceptos Básicos y Variables

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

```
aleix@aleix:~/BigData2526/PR_01.1$ echo $HOME  
/home/aleix  
aleix@aleix:~/BigData2526/PR_01.1$ cd $HOME  
aleix@aleix:~$ pwd  
/home/aleix
```

2. Ejecuta el comando **whoami**. Ahora, crea una variable local llamada **USUARIO_ACTUAL** que contenga el resultado del comando anterior y muéstralala en la terminal.

```
aleix@aleix:~$ whoami  
aleix  
aleix@aleix:~$ USUARIO_ACTUAL=$(whoami)  
aleix@aleix:~$ echo $USUARIO_ACTUAL  
aleix
```

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.

Se han creado dos archivos porque, al haber un espacio entre ellos, el sistema los interpreta como dos nombres distintos.

```
aleix@aleix:~/BigData2526/PR_01.1$ touch dos palabras.txt  
aleix@aleix:~/BigData2526/PR_01.1$ ls  
dos palabras.txt  
aleix@aleix:~/BigData2526/PR_01.1$ rm dos palabras.txt  
aleix@aleix:~/BigData2526/PR_01.1$ touch "dos palabras.txt"  
aleix@aleix:~/BigData2526/PR_01.1$ ls  
'dos palabras.txt'  
aleix@aleix:~/BigData2526/PR_01.1$ _
```

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

ls es externo y cd es interno → cd depende del shell, mientras que ls es un binario en /bin.

Los comandos internos son más rápidos y solo existen dentro del shell, mientras que los externos están en el sistema y pueden usarse desde cualquier script o shell distinto.

```
aleix@aleix:~/BigData2526/PR_01.1$ type ls
ls is aliased to `ls --color=auto'
aleix@aleix:~/BigData2526/PR_01.1$ type cd
cd is a shell builtin
aleix@aleix:~/BigData2526/PR_01.1$
```

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.

```
aleix@aleix:~/BigData2526/PR_01.1$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/usr/games:/usr/local/games:/snap/bin
aleix@aleix:~/BigData2526/PR_01.1$ mkdir ~/mi_bin
aleix@aleix:~/BigData2526/PR_01.1$ export PATH=~/mi_bin:$PATH
aleix@aleix:~/BigData2526/PR_01.1$ echo $PATH
/home/aleix/mi_bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/bin:/usr/games:/usr/local/games:/snap/bin
```

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?

El número 1 indica que contiene comandos de usuario (programas ejecutables)

```
aleix@aleix:~$ man chmod
CHMOD(1)
NAME
      chmod - change file mode bits
```

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

```
aleix@aleix:~$ man ls
...
size
  -S, --size
    print the allocated size of each file, in blocks
  -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.

```
aleix@aleix:~/BigData2526/PR_01.1$ sudo find / -name passwd 2>/dev/null
[sudo] password for aleix:
/usr/share/bash-completion/completions/passwd
/usr/share/doc/passwd
/usr/share/lintian/overrides/passwd
/usr/bin/passwd
/etc/passwd
/etc/pam.d/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 tenía instalado el paquete locate, así que tuve que instalarlo y sí que me funcionó.

```
aleix@aleix:~/BigData2526/PR_01.1$ touch ~/test_locate.txt
aleix@aleix:~/BigData2526/PR_01.1$ locate test_locate.txt
Command 'locate' not found, but can be installed with:
sudo apt install plocate
aleix@aleix:~/BigData2526/PR_01.1$ sudo apt install plocate
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  liburing2
The following NEW packages will be installed:
  liburing2 plocate
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 160 kB of archives.
After this operation, 623 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://es.archive.ubuntu.com/ubuntu noble amd64 liburing2 amd64 2.5-1build1 [21.1 kB]
Get:2 http://es.archive.ubuntu.com/ubuntu noble/universe amd64 plocate amd64 1.1.19-2ubuntu2 [139 kB]
Fetched 160 kB in 1s (226 kB/s)
Selecting previously unselected package liburing2:amd64.
(Reading database ... 84404 files and directories currently installed.)
Preparing to unpack .../liburing2_2.5-1build1_amd64.deb ...
Unpacking liburing2:amd64 (2.5-1build1) ...
Selecting previously unselected package plocate.
Preparing to unpack .../plocate_1.1.19-2ubuntu2_amd64.deb ...
Unpacking plocate (1.1.19-2ubuntu2) ...
Setting up liburing2:amd64 (2.5-1build1) ...
Setting up plocate (1.1.19-2ubuntu2) ...
update-alternatives: using /usr/bin/plocate to provide /usr/bin/locate (locate) in auto mode
info: Selecting GID from range 100 to 999 ...
info: Adding group `plocate' (GID 110) ...
Initializing plocate database; this may take some time... done
Created symlink /etc/systemd/system/timers.target.wants/plocate-updatedb.timer → /usr/lib/systemd/sys...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
aleix@aleix:~/BigData2526/PR_01.1$ locate test_locate.txt
/home/aleix/test_locate.txt
aleix@aleix:~/BigData2526/PR_01.1$
```

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.

Tendría que actualizar la BBDD del paquete locate para que pudiera encontrar el archivo recién añadido.

```
aleix@aleix:~/BigData2526/PR_01.1$ sudo updatedb
aleix@aleix:~/BigData2526/PR_01.1$ locate test_locate.txt
/home/aleix/test_locate.txt
```

3- Navegación y Listado de Archivos

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

```
aleix@aleix:~$ cd /etc
aleix@aleix:/etc$ ls ~
BigData2526  mi_bin  test_locate.txt
```

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

```
aleix@aleix:/etc$ cd ~
aleix@aleix:~$ cd ../../var/log
aleix@aleix:/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** .

```
lrwxrwxrwx 1 root root          21 Aug  1 14:21 os-release
-rw-r--r-- 1 root root        6920 Aug  5 17:14 overlayroot
drwxr-xr-x 2 root root        4096 Aug  5 17:02 PackageKit
-rw-r--r-- 1 root root         552 Oct 13 2022 pam.conf
drwxr-xr-x 2 root root        4096 Oct 15 10:02 pam.d
-rw-r--r-- 1 root root        1734 Oct 15 09:31 passwd
-rw-r--r-- 1 root root       1688 Oct 15 09:28 passwd-
drwxr-xr-x 3 root root        4096 Aug  5 17:02 perl
drwxr-xr-x 4 root root        4096 Aug  5 17:14 pki
```

4. Compara la salida de **ls -l /etc** y **ls -lh /etc** . ¿Qué hace la opción **h** y por qué es útil para las personas?

Es útil ya que lo que hace es ponerlo más humano en los tamaños de archivo en KB, MB o GB en lugar de bytes:

```
drwxr-xr-x 2 root root        4.0K Aug  5 17:14 sysstat
drwxr-xr-x 6 root root        4.0K Oct 15 10:02 systemd
drwxr-xr-x 2 root root        4.0K Aug  5 17:00 terminfo
drwxr-xr-x 2 root root        4.0K Oct 15 09:28 thermald
-rw-r--r-- 1 root root         8 Aug  5 17:02 timezone
drwxr-xr-x 2 root root        4.0K Aug  5 17:14 tmpfiles.d
drwxr-xr-x 2 root root        4.0K Aug  5 17:14 ubuntu-adantage
-rw-r--r-- 1 root root       1.3K Jan 27 2023 ucf.conf
```

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

La opción ***-R*** (de Recursive) le dice a ***ls*** que muestre también el contenido de **todos los subdirectorios**, y dentro de ellos, los suyos, y así sucesivamente. Puede ser peligroso ya que si abrimos un directorio base con miles de archivos podría llegar a petarse la máquina o que fuera tan lenta que no sea útil.

```
aleix@aleix:/var/log$ ls -R ~
/home/aleix:
BigData2526  mi_bin  test_locate.txt

/home/aleix/BigData2526:
PR_01.1

/home/aleix/BigData2526/PR_01.1:
'dos palabras.txt'

/home/aleix/mi_bin:
```

4- Manipulación de Archivos y Directorios

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

```
aleix@aleix:~/BigData2526/PR_01.1$ mkdir -p proyecto/{src,doc,bin}
aleix@aleix:~/BigData2526/PR_01.1$ ls
'dos palabras.txt'  proyecto
aleix@aleix:~/BigData2526/PR_01.1$ ls proyecto/
bin  doc  src
```

2. Crea un archivo ***~/notas.txt*** . Muévelo a ***~/proyecto/doc*** y, en el mismo comando, renómbralo a ***README.md*** .

```
aleix@aleix:~/BigData2526/PR_01.1$ touch ~/notas.txt
aleix@aleix:~/BigData2526/PR_01.1$ mv ~/notas.txt ~/.
.bash_history          BigData2526/                .local/           .profile      test_locate.
.bash_logout            .cache/                  mi_bin/          .ssh/         .
.bashrc                .lesshtst               notas.txt       .sudo_as_admin_successful
aleix@aleix:~/BigData2526/PR_01.1$ mv ~/notas.txt ~/BigData2526/
.git/                 PR_01.1/
aleix@aleix:~/BigData2526/PR_01.1$ mv ~/notas.txt ~/BigData2526/PR_01.1/
dos palabras.txt  proyecto/
aleix@aleix:~/BigData2526/PR_01.1$ mv ~/notas.txt ~/BigData2526/PR_01.1/proyecto/doc/README.md
aleix@aleix:~/BigData2526/PR_01.1$ ls proyecto/doc/
README.md
aleix@aleix:~/BigData2526/PR_01.1$
```

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

```
aleix@aleix:~/BigData2526/PR_01.1/proyecto$ cp doc/README.md bin/
aleix@aleix:~/BigData2526/PR_01.1/proyecto$ rm doc/README.md
```

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

```
aleix@aleix:~/BigData2526/PR_01.1$ rmdir proyecto/
rmdir: failed to remove 'proyecto/': Directory not empty
aleix@aleix:~/BigData2526/PR_01.1$ rm -rf proyecto/
aleix@aleix:~/BigData2526/PR_01.1$ ls
'dos palabras.txt'
```

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** .

```
aleix@aleix:~/BigData2526/PR_01.1$ ls /etc/s*.conf
/etc/sensors3.conf  /etc/sudo.conf  /etc/sudo_logsrvd.conf  /etc/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?

Me dice que faltan permisos:

```
tar: /var/log/installer/installer-journal.txt: Cannot open: Permission denied
tar: /var/log/installer/curtin-install.log
tar: /var/log/installer/casper-md5check.json
tar: /var/log/installer/subiquity-server-info.log.1408: Cannot open: Permission denied
tar: /var/log/installer/subiquity-server-debug.log.1408: Cannot open: Permission denied
tar: /var/log/installer/media-info
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/subiquity-server-info.log
tar: /var/log/installer/subiquity-server-debug.log
tar: /var/log/installer/device-map.json
tar: /var/log/installer/curtin-install/
tar: /var/log/installer/curtin-install/subiquity-curtin-apt.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-extract.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-initial.conf: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-curthooks.conf: Cannot open: Permission denied
tar: /var/log/installer/subiquity-client-info.log
tar: /var/log/installer/subiquity-client-debug.log
tar: /var/log/installer/subiquity-client-debug.log.1372: Cannot open: Permission denied
tar: /var/log/installer/block/
tar: /var/log/installer/block/probe-data.json
tar: /var/log/installer/block/discover.log
tar: /var/log/installer/subiquity-client-info.log.1372: Cannot open: Permission denied
tar: /var/log/private: Cannot open: Permission denied
tar: /var/log/apport.log
tar: Exiting with failure status due to previous errors
```

Con el sudo, ya no hay problema de permisos

```
/var/log/installer/subiquity-client-debug.log.1372
/var/log/installer/block/
/var/log/installer/block/probe-data.json
/var/log/installer/block/discover.log
/var/log/installer/subiquity-client-info.log.1372
/var/log/private/
/var/log/apport.log
aleix@aleix:~/BigData2526/PR_01.1$ ls
'dos palabras.txt'  log_backup.tar
```

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

```
aleix@aleix:~/BigData2526/PR_01.1$ ls -lh log_backup.*
-rw-r--r-- 1 root root 91M Oct 20 07:14 log_backup.tar
-rw-rw-r-- 1 aleix aleix 1.5M Oct 20 07:12 log_backup.tar.gz
```

3. Lista el contenido del archivo ***log_backup.tar.gz*** sin extraerlo para verificar que los archivos están dentro.

Con este comando se muestra todo lo que hay dentro sin extraer:

```
var/log/installer/curtin-install/subiquity-curtin-apt.conf
var/log/installer/curtin-install/subiquity-partitioning.conf
var/log/installer/curtin-install/subiquity-extract.conf
var/log/installer/curtin-install/subiquity-initial.conf
var/log/installer/curtin-install/subiquity-curthooks.conf
var/log/installer/subiquity-client-info.log
var/log/installer/subiquity-client-debug.log
var/log/installer/subiquity-client-debug.log.1372
var/log/installer/block/
var/log/installer/block/probe-data.json
var/log/installer/block/discover.log
var/log/installer/subiquity-client-info.log.1372
var/log/private/
var/log/apport.log
aleix@aleix:~/BigData2526/PR_01.1$ tar -tzf log_backup.tar.gz _
```

4. Extrae únicamente el archivo ***syslog*** (o ***messages***) de ***log_backup.tar.gz*** a tu directorio ***/tmp***.

```
aleix@aleix:~/BigData2526/PR_01.1$ tar -tzf log_backup.tar.gz | grep syslog
var/log/syslog
aleix@aleix:~/BigData2526/PR_01.1$ tar -xzf log_backup.tar.gz var/log/sys -C /tmp
var/log/syslog      var/log/sysstat/      var/log/sysstat/sa15  var/log/sysstat/sa16  var/log/sysstat/sa20
aleix@aleix:~/BigData2526/PR_01.1$ tar -xzf log_backup.tar.gz var/log/syslog -C /tmp
```

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

```
aleix@aleix:~/BigData2526/PR_01.1$ touch a.txt b.log c.jpg
aleix@aleix:~/BigData2526/PR_01.1$ zip mis_archivos a.txt b.log c.jpg
Command 'zip' not found, but can be installed with:
sudo apt install zip
aleix@aleix:~/BigData2526/PR_01.1$ sudo apt install zip
```

No tenía el zip instalado así que lo instalé

```
aleix@aleix:~/BigData2526/PR_01.1$ zip mis_archivos a.txt b.log c.jpg
adding: a.txt (stored 0%)
adding: b.log (stored 0%)
adding: c.jpg (stored 0%)
aleix@aleix:~/BigData2526/PR_01.1$ ls
a.txt  b.log  c.jpg  'dos palabras.txt'  log_backup.tar  log_backup.tar.gz  mis_archivos.zip  var
```

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

```
aleix@aleix:~/BigData2526/PR_01.1$ rm a.txt b.log c.jpg
aleix@aleix:~/BigData2526/PR_01.1$ unzip mis_archivos.zip
Archive: mis_archivos.zip
extracting: a.txt
extracting: b.log
extracting: c.jpg
```

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

Aquí lo que hago es poner **HEAD** para que solo muestre las primeras 10 líneas de archivo y que no me salgan las posibles miles de líneas que puede tener.

```
aleix@aleix:~/BigData2526/PR_01.1$ zcat /var/log/*.gz | head
[ 0.00000] kernel: Linux version 6.8.0-85-generic (buildd@lcy02-amd64-106) (x86_64-linux-gnu-gcc-13 (Ubuntu 13.5
ils for Ubuntu) 2.42) #85-Ubuntu SMP PREEMPT_DYNAMIC Thu Sep 18 15:26:59 UTC 2025 (Ubuntu 6.8.0-85.85-generic 6.8.12
[ 0.00000] kernel: Command line: BOOT_IMAGE=/vmlinuz-6.8.0-85-generic root=/dev/mapper/ubuntu--vg-ubuntu--lv ro
[ 0.00000] kernel: KERNEL supported cpus:
[ 0.00000] kernel: Intel GenuineIntel
[ 0.00000] kernel: AMD AuthenticAMD
[ 0.00000] kernel: Hygon HygonGenuine
[ 0.00000] kernel: Centaur CentaurHauls
[ 0.00000] kernel: zhaoxin Shanghai
[ 0.00000] kernel: BIOS-provided physical RAM map:
[ 0.00000] kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000009fbff] usable
```

6- Redirección,Tuberías y Filtros

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

```
aleix@aleix:~/BigData2526/PR_01.1$ ls -l ~ > mis_archivos.txt
```

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

```
aleix@aleix:~/BigData2526/PR_01.1$ date >> mis_archivos.txt
```

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

```
aleix@aleix:~/BigData2526/PR_01.1$ ls -l /etc | grep '^d' | wc -l
106
```

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

```
aleix@aleix:~/BigData2526/PR_01.1$ tail -n 10 /etc/passwd | cut -d: -f1
pollinate
polkitd
syslog
uuidd
tcpdump
tss
landscape
fwupd-refresh
usbmux
aleix
```

- 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.

```
aleix@aleix:~/BigData2526/PR_01.1$ ps aux | sort -k3 -r | head -n 5
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root      2057  0.8  2.1 667984 42656 ?        Ssl  07:32  0:00 /usr/libexec/fwupd/fwupd
root        8  0.3  0.0     0     0 ?        I   06:35  0:12 [kworker/0:0-events]
root      344  0.0  1.3 289116 27392 ?        SLs1 06:35  0:00 /sbin/multipathd -d -s
root      738  0.0  1.1 109644 22912 ?        Ssl  06:35  0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unatt
```

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

Si pones solo un “`>`” lo que hace es reemplazar todo lo que tenga el archivo por lo que escribiste, y si pones “`>>`” lo que haces es adjuntarlo al archivo, no se reemplaza

```
aleix@aleix:~/BigData2526/PR_01.1$ echo "Hola" > ejemplo.txt
aleix@aleix:~/BigData2526/PR_01.1$ cat ejemplo.txt
Hola
aleix@aleix:~/BigData2526/PR_01.1$ echo "Mundo" >> ejemplo.txt
aleix@aleix:~/BigData2526/PR_01.1$ cat ejemplo.txt
Hola
Mundo
aleix@aleix:~/BigData2526/PR_01.1$ echo "Prueba" > ejemplo.txt
aleix@aleix:~/BigData2526/PR_01.1$ cat ejemplo.txt
Prueba
```

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`.

```
aleix@aleix:~/BigData2526/PR_01.1$ find /etc -name "*.conf" > config_files.txt 2> 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.

```
aleix@aleix:~/BigData2526/PR_01.1$ cat mi_script.sh
echo "Usuario: $USER"
echo "Directorio actual: $PWD"
```

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.

No tendría los permisos necesarios para acceder al script que se ha puesto solo para mi usuario.

```
aleix@aleix:~/BigData2526/PR_01.1$ chmod u+x mi_script.sh
aleix@aleix:~/BigData2526/PR_01.1$ ./mi_script.sh
Usuario: aleix
Directorio actual: /home/aleix/BigData2526/PR_01.1
```

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
if [ "$1" = "hola" ]; then
    echo "mundo"
fi
```

```
aleix@aleix:~/BigData2526/PR_01.1$ ./mi_script.sh hola
mundo
aleix@aleix:~/BigData2526/PR_01.1$ ./mi_script.sh adios
```

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

if [ -z "$1" ]; then
    echo "Error: Debes proporcionar un argumento."
    exit 1
fi

if [ "$1" = "hola" ]; then
    echo "mundo"
fi
```

```
aleix@aleix:~/BigData2526/PR_01.1$ ./mi_script.sh
Error: Debes proporcionar un argumento.
aleix@aleix:~/BigData2526/PR_01.1$ ./mi_script.sh hola
mundo
```

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

if [ "$1" -eq "$2" ]; then
    echo "iguales"
else
    echo "diferentes"
fi
```

```
aleix@aleix:~/BigData2526/PR_01.1$ ./script_num.sh 3 4
-bash: ./script_num.sh: Permission denied
aleix@aleix:~/BigData2526/PR_01.1$ chmod u+x script_num.sh
aleix@aleix:~/BigData2526/PR_01.1$ ./script_num.sh 3 4
diferentes
aleix@aleix:~/BigData2526/PR_01.1$ ./script_num.sh 5 5
iguales
aleix@aleix:~/BigData2526/PR_01.1$ _
```

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 **.bak** a cada archivo.

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

if [ -z "$1" ]; then
    echo "Error: Debes indicar un directorio."
    exit 1
fi

for file in $(ls "$1"); do
    mv "$1/$file" "$1/${file}.bak"
done
```

```
aleix@aleix:~/BigData2526/PR_01.1$ chmod u+x script_dic.sh
aleix@aleix:~/BigData2526/PR_01.1$ mkdir prueba
aleix@aleix:~/BigData2526/PR_01.1$ cd prueba/
aleix@aleix:~/BigData2526/PR_01.1/prueba$ touch p1.txt p2.txt
aleix@aleix:~/BigData2526/PR_01.1/prueba$ ls
p1.txt  p2.txt

aleix@aleix:~/BigData2526/PR_01.1$ ./script_dic.sh ~/BigData2526/PR_01.1/prueba/
aleix@aleix:~/BigData2526/PR_01.1$ ls prueba/
p1.txt.bak  p2.txt.bak
```

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**).

```
aleix@aleix:~/BigData2526/PR_01.1$ cut -d: -f7 /etc/passwd | sort | uniq
/bin/bash
/bin/false
/bin/sync
/usr/sbin/nologin
```

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

```
aleix@aleix:~/BigData2526/PR_01.1$ ps aux | grep ^root | wc -l
35
```

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** .

```
aleix@aleix:~/BigData2526/PR_01.1$ ls -l /etc | grep 'Oct' > october_files.txt
```

4. Usando **globbing**, lista todos los archivos en **/etc** que contengan un número en su nombre.

```
aleix@aleix:~/BigData2526/PR_01.1$ ls /etc/*[0-9]*
/etc/e2scrub.conf  /etc/mke2fs.conf  /etc/sensors3.conf

/etc/dbus-1:
session.d  system.d

/etc/iproute2:
bpf_pinning  ematch_map  group  nl_protos  rt_dsfld  rt_protos  rt_protos.d  rt_realms  rt_scopes  rt_tables  rt_tab

/etc/libnl-3:
classid  pktloc

/etc/polkit-1:
rules.d

/etc/python3:
debian_config

/etc/python3.12:
sitecustomize.py

/etc/rc0.d:
K01cryptdisks  K01cryptdisks-early  K01iscsid  K01open-iscsi  K01open-vm-tools  K01plymouth  K01unattended-upgrades  K

/etc/rc1.d:
K01iscsid  K01open-iscsi  K01open-vm-tools  K01ufw  K01uuidd

/etc/rc2.d:
S01apport  S01console-setup.sh  S01cron  S01dbus  S01grub-common  S01open-vm-tools  S01plymouth  S01rsync  S01sysstat

/etc/rc3.d:
S01apport  S01console-setup.sh  S01cron  S01dbus  S01grub-common  S01open-vm-tools  S01plymouth  S01rsync  S01sysstat

/etc/rc4.d:
S01apport  S01console-setup.sh  S01cron  S01dbus  S01grub-common  S01open-vm-tools  S01plymouth  S01rsync  S01sysstat

/etc/rc5.d:
S01apport  S01console-setup.sh  S01cron  S01dbus  S01grub-common  S01open-vm-tools  S01plymouth  S01rsync  S01sysstat

/etc/rc6.d:
K01cryptdisks  K01cryptdisks-early  K01iscsid  K01open-iscsi  K01open-vm-tools  K01plymouth  K01unattended-upgrades  K

/etc/udisks2:
mount_options.conf.example  udisks2.conf

/etc/X11:
xkb_Xsession.d
```

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

No encontré ninguno

```
aleix@aleix:~/BigData2526/PR_01.1$ find /usr/bin/ -type f -executable ! -user root
```

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

```
aleix@aleix:~/BigData2526/PR_01.1$ gzip -c /var/log/syslog > syslog.gz
aleix@aleix:~/BigData2526/PR_01.1$ bzip2 -c /var/log/syslog > syslog.bz2
Command 'bzip2' not found, but can be installed with:
sudo apt install bzip2
[sudo] password for aleix:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Suggested packages:
  bzip2-doc
The following NEW packages will be installed:
  bzip2
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 34.5 kB of archives.
After this operation, 112 kB of additional disk space will be used.
Get:1 http://es.archive.ubuntu.com/ubuntu noble-updates/main amd64 bzip2 amd64 1.0.8-5.1build0.1 [34.5 kB]
Fetched 34.5 kB in 6s (5,966 B/s)
Selecting previously unselected package bzip2.
(Reading database ... 84459 files and directories currently installed.)
Preparing to unpack .../bzip2_1.0.8-5.1build0.1_amd64.deb ...
Unpacking bzip2 (1.0.8-5.1build0.1) ...
Setting up bzip2 (1.0.8-5.1build0.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
aleix@aleix:~/BigData2526/PR_01.1$ bzip2 -c /var/log/syslog > syslog.bz2
aleix@aleix:~/BigData2526/PR_01.1$ ls -lh syslog.gz syslog.bz2
-rw-rw-r-- 1 aleix aleix 50K Oct 20 08:39 syslog.bz2
-rw-rw-r-- 1 aleix aleix 105K Oct 20 08:38 syslog.gz
aleix@aleix:~/BigData2526/PR_01.1$ time gzip -c /var/log/syslog > /dev/null

real    0m0.011s
user    0m0.010s
sys     0m0.000s
aleix@aleix:~/BigData2526/PR_01.1$ time bzip2 -c /var/log/syslog > /dev/null

real    0m0.033s
user    0m0.030s
sys     0m0.002s
```

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.

```
aleix@aleix:~/BigData2526/PR_01.1$ ln -s ~/BigData2526/PR_01.1/prueba/ ~/enlace_prueba
aleix@aleix:~/BigData2526/PR_01.1$ tar -chf home_backup.tar ~/enlace_prueba
tar: Removing leading `/' from member names
tar: Removing leading `/' from hard link targets
```

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

if [ -f "$1" ]; then
    echo "$1 es un archivo regular."
elif [ -d "$1" ]; then
    echo "$1 es un directorio."
else
    echo "$1 no existe."
fi
```

```
aleix@aleix:~/BigData2526/PR_01.1$ chmod u+x check_file.sh
aleix@aleix:~/BigData2526/PR_01.1$ ./check_file.sh prueba/
prueba/ es un directorio.
aleix@aleix:~/BigData2526/PR_01.1$ ./check_file.sh prueba/p
p1.txt.bak p2.txt.bak
aleix@aleix:~/BigData2526/PR_01.1$ ./check_file.sh prueba/p2.txt.bak
prueba/p2.txt.bak es un archivo regular.
aleix@aleix:~/BigData2526/PR_01.1$ ./check_file.sh prueba/p2.txt.bak4
prueba/p2.txt.bak4 no existe.
```

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

```
GNU nano 7.2                                         create_test_dir.sh
#!/bin/bash

mkdir /test_dir 2>/dev/null

if [ $? -eq 0 ]; then
    echo "Directorio creado con éxito."
else
    echo "No se puede crear el directorio. Probablemente por permisos."
fi
```

```
aleix@aleix:~/BigData2526/PR_01.1$ chmod u+x create_test_dir.sh
aleix@aleix:~/BigData2526/PR_01.1$ ./create_test_dir.sh
No se puede crear el directorio. Probablemente por permisos.
aleix@aleix:~/BigData2526/PR_01.1$ sudo ./create_test_dir.sh
[sudo] password for aleix:
Directorio creado con éxito.
```

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                                     filter_numbers.sh
#!/bin/bash

for arg in "$@"; do
    if [[ "$arg" =~ ^[0-9]+$ ]] && [ "$arg" -gt 10 ];then
        echo "$arg"
    fi
done_
aleix@aleix:~/BigData2526/PR_01.1$ chmod u+x filter_numbers.sh
aleix@aleix:~/BigData2526/PR_01.1$ ./filter_numbers.sh 5 12 7 20 abc 15
12
20
15
```