

PR_01.1 Dani Gayol Rodríguez

PR_01.1 Dani Gayol Rodríguez.....	1
1.) Conceptos Básicos y Variables.....	2
2.) Obtener Ayuda y Localizar Archivos	4
3.- Navegación y Listado de Archivos	7
4.- Manipulación de Archivos y Directorios.....	10
5.- Archivado y Compresión	12
6.- Redirección, Tuberías y Filtros.....	15
7.- Scripts Básicos	17
8.- Ejercicios Avanzados	20

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.

```
user@user-dani:/$ echo $HOME
/home/user
user@user-dani:/$ ls ~
new_script
user@user-dani:/$ cd ~
user@user-dani:~$ pwd
/home/user
```

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

```
user@user-dani:~$ whoami
user
user@user-dani:~$ echo 'whoami' > USUARIO_ACTUAL
user@user-dani:~$ chmod +x USUARIO_ACTUAL
user@user-dani:~$ ls
new_script  USUARIO_ACTUAL
user@user-dani:~$ ./USUARIO_ACTUAL
user
```

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.

```
user@user-dani:~$ touch dos palabras.txt
user@user-dani:~$ ls
dos  new_script  palabras.txt  USUARIO_ACTUAL
```

Lo que pasa es que al poner un espacio sin usar las comillas, intenta crear dos archivos, entonces por eso al poner “ls” muestra dos archivos

```
user@user-dani:~$ ls
dos  new_script  palabras.txt  USUARIO_ACTUAL
user@user-dani:~$ rm dos palabras.txt
user@user-dani:~$ ls
new_script  USUARIO_ACTUAL
user@user-dani:~$ touch "dos palabras.txt"
user@user-dani:~$ ls
'dos palabras.txt'  new_script  USUARIO_ACTUAL
```

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

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

El comando “ls” es un comando externo, en cambio, el comando “cd” es interno. Los comandos internos pueden cambiar el estado del shell y los comandos externos son programas independientes que se ejecutan y terminan

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.

```
user@user-dani:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
user@user-dani:~$ mkdir ~/mi_bin
user@user-dani:~$ export PATH=~/mi_bin:$PATH
user@user-dani:~$ echo $PATH
/home/user/mi_bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/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?

```
user@user-dani:~$ man chmod

CHMOD(1)                                User Commands                                CHMOD(1)
NAME
  chmod - change file mode bits

SYNOPSIS
  chmod [OPTION]... MODE[,MODE]... FILE...
  chmod [OPTION]... OCTAL-MODE FILE...
  chmod [OPTION]... --reference=RFILE FILE...

DESCRIPTION
  This manual page documents the GNU version of chmod.  chmod changes the file mode bits of each given file according to mode, which can be either a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.

  The format of a symbolic mode is [ugoa...][[-=]][perms...], where perms is either zero or more letters from the set rwXst, or a single letter from the set ugo. Multiple symbolic modes can be given, separated by commas.

  A combination of the letters ugo controls which users' access to the file will be changed: the user who owns it (u), other users in the file's group (g), other users not in the file's group (o), or all users (a). If none of these are given, the effect is as if (a) were given, but bits that are set in the umask are not affected.

  The operator + causes the selected file mode bits to be added to the existing file mode bits of each file; - causes them to be removed; and = causes them to be added and causes unmentioned bits to be removed except that a directory's unmentioned set user and group ID bits are not affected.

  The letters rwXst select file mode bits for the affected users: read (r), write (w), execute (or search for directories) (X), execute/search only if the file is a directory or already has execute permission for some user (x), set user or group ID on execution (s), restricted deletion flag or sticky bit (t). Instead of one or more of these letters, you can specify exactly one of the letters ugo: the permissions granted to the user who owns the file (u), the permissions granted to other users who are members of the file's group (g), and the permissions granted to users that are in neither of the two preceding categories (o).

  A numeric mode is from one to four octal digits (0-7), derived by adding up the bits with values 4, 2, and 1. Omitted digits are assumed to be leading zeros. The first digit selects the set user ID (4) and set group ID (2) and restricted deletion or sticky (1) attributes. The second digit selects permissions for the user who owns the file: read (4), write (2), and execute (1); the third selects permissions for other users in the file's group, with the same values; and the fourth for other users not in the file's group, with the same values.

  chmod never changes the permissions of symbolic links; the chmod system call cannot change their permissions. This is not a problem since the permissions of symbolic links are never used. However, for each symbolic link listed on the command line, chmod changes the permissions of the pointed-to file. In contrast, chmod ignores symbolic links encountered during recursive directory traversals.

SETUID AND SETGID BITS
  chmod clears the set-group-ID bit of a regular file if the file's group ID does not match the user's effective group ID or one of the user's supplementary group IDs, unless the user has appropriate privileges. Additional restrictions may cause the set-user-ID and set-group-ID bits of MODE or RFILE to be ignored. This behavior depends on the policy and functionality of the underlying chmod system call. When in doubt, check the underlying system behavior.

  For directories chmod preserves set-user-ID and set-group-ID bits unless you explicitly specify otherwise. You can set or clear the bits with symbolic modes like u+s and g-s. To clear these bits for directories with a numeric mode requires an additional leading zero like 00755, leading minus like -6000, or leading equals like =755.

Manual page chmod(1) line 1 (press h for help or q to quit)
```

Se encuentra en la sección del manual “1”, aparece arriba del todo entre paréntesis “`chmod(1)`”. El número 1 indica que es un comando de usuario ejecutable

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

```
user@user-dani:~$ man ls_
```

```

-S, --size
    print the allocated size of each file, in blocks

-S
    sort by file size, largest first

--sort=WORD
    sort by WORD instead of name: none (-U), size (-S), time (-t), version (-v), extension (-X), width

--time=WORD
    select which timestamp used to display or sort; access time (-u): atime, access, use; metadata change time (-c): ctime, status; modified time
    (default): mtime, modification; birth time: birth, creation;

    with -l, WORD determines which time to show; with --sort=time, sort by WORD (newest first)

--time-style=TIME_STYLE
    time/date format with -l; see TIME_STYLE below

-t
    sort by time, newest first; see --time

-T, --tabsize=COLS
    assume tab stops at each COLS instead of 8

-u
    with -lt: sort by, and show, access time; with -l: show access time and sort by name; otherwise: sort by access time, newest first

-U
    do not sort; list entries in directory order

-v
    natural sort of (version) numbers within text

-w, --width=COLS
    set output width to COLS. 0 means no limit

-x
    list entries by lines instead of by columns

-X
    sort alphabetically by entry extension

/size

```

Puedes buscar en el manual usando el “/”, por ejemplo, para buscar el comando para ordenar los archivos por tamaño, puse “/size”

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.

```

user@user-dani:/$ sudo find / -name passwd
/usr/share/bash-completion/completions/passwd
/usr/share/doc/passwd
/usr/share/lintian/overrides/passwd
/usr/bin/passwd
/etc/passwd
/etc/pam.d/passwd

```

La ruta completa es “/etc/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?

```

user@user-dani:/$ touch ~/test_locate.txt
user@user-dani:/$ ls ~
'dos palabras.txt'  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL
user@user-dani:/$ locate test_locate.txt
/home/user/test_locate.txt

```

Si aparece ya que estoy usando “plocate” el cual es más rápido, por lo tanto, aparece antes en la base de datos

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.

```
user@user-dani:/$ sudo updatedb
user@user-dani:/$ locate test_locate.txt
/home/user/test_locate.txt
```

El primer comando sirve para actualizar la base de datos, por lo tanto, al ejecutar el “locate”, si lo muestra ya que la base de datos se actualizó correctamente

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 ~.

```
user@user-dani:/$ cd /etc
user@user-dani:/etc$ ls ~
new_script  USUARIO_ACTUAL
user@user-dani:/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.

```
user@user-dani:~$ cd / && cd var && cd log
user@user-dani:/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.

```
user@user-dani:~$ ls -l /etc
```

```

drwxr-xr-x 2 root root 4096 ago 5 17:14 sensors.d
-rw-r--r-- 1 root root 12813 mar 27 2021 services
drwxr-xr-x 3 root root 4096 oct 16 06:55 sgml
-rw-r----- 1 root shadow 966 oct 15 09:50 shadow
-rw-r----- 1 root shadow 966 oct 15 09:50 shadow-
-rw-r--r-- 1 root root 148 ago 5 17:14 shells
drwxr-xr-x 2 root root 4096 ago 5 16:55 skel
drwxr-xr-x 6 root root 4096 ago 5 17:14 sos
drwxr-xr-x 4 root root 4096 oct 15 09:50 ssh
drwxr-xr-x 4 root root 4096 oct 15 09:49 ssl
-rw-r--r-- 1 root root 18 oct 15 09:50 subgid
-rw-r--r-- 1 root root 0 ago 5 16:54 subgid-
-rw-r--r-- 1 root root 18 oct 15 09:50 subuid
-rw-r--r-- 1 root root 0 ago 5 16:54 subuid-
-rw-r--r-- 1 root root 4343 jun 25 12:42 sudo.conf
-r--r----- 1 root root 1800 ene 29 2024 sudoers
drwxr-xr-x 2 root root 4096 ago 5 17:02 sudoers.d
-rw-r--r-- 1 root root 9804 jun 25 12:42 sudo_logsrvd.conf
drwxr-xr-x 2 root root 4096 ago 5 17:14 supercat
-rw-r--r-- 1 root root 2209 mar 24 2024 sysctl.conf
drwxr-xr-x 2 root root 4096 ago 5 17:02 sysctl.d
drwxr-xr-x 2 root root 4096 ago 5 17:14 sysstat
drwxr-xr-x 6 root root 4096 ago 5 16:49 systemd
drwxr-xr-x 2 root root 4096 ago 5 17:00 terminfo
drwxr-xr-x 2 root root 4096 oct 15 09:48 thermald
-rw-r--r-- 1 root root 8 ago 5 17:02 timezone
drwxr-xr-x 2 root root 4096 ago 5 17:14 tmpfiles.d
drwxr-xr-x 2 root root 4096 ago 5 17:14 ubuntu-advantage
-rw-r--r-- 1 root root 1260 ene 27 2023 ucf.conf
drwxr-xr-x 4 root root 4096 ago 5 17:02 udev
drwxr-xr-x 2 root root 4096 oct 15 09:49 udisks2
drwxr-xr-x 3 root root 4096 ago 5 17:14 ufw
drwxr-xr-x 3 root root 4096 ago 5 17:02 update-manager
drwxr-xr-x 2 root root 4096 ago 5 17:14 update-motd.d
drwxr-xr-x 2 root root 4096 ago 5 17:14 update-notifier
drwxr-xr-x 2 root root 4096 oct 15 09:48 UPower
-rw-r--r-- 1 root root 1523 ago 5 17:14 usb_modeswitch.conf
drwxr-xr-x 2 root root 4096 ago 5 17:14 usb_modeswitch.d
lrwxrwxrwx 1 root root 16 ago 5 17:02 vconsole.conf -> default/keyboard
drwxr-xr-x 2 root root 4096 oct 15 09:49 vim
drwxr-xr-x 4 root root 4096 oct 15 09:49 vmware-tools
lrwxrwxrwx 1 root root 23 feb 26 2024 vtrgb -> /etc/alternatives/vtrgb
drwxr-xr-x 5 root root 4096 oct 16 06:55 vulkan
-rw-r--r-- 1 root root 4942 ago 5 17:14 wgetrc
drwxr-xr-x 7 root root 4096 oct 16 06:55 X11
-rw-r--r-- 1 root root 681 abr 8 2024 xattr.conf
drwxr-xr-x 5 root root 4096 oct 16 06:55 xdg
drwxr-xr-x 2 root root 4096 oct 16 06:55 xml
-rw-r--r-- 1 root root 460 ago 5 17:14 zsh_command_not_found
user@user-dani:~$

```

```

user@user-dani:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1777 oct 15 09:50 /etc/passwd
user@user-dani:~$

```

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?

```

user@user-dani:~$ ls -l /etc && ls -lh /etc

```

La diferencia entre “`-l`” y “`-lh`” es que si solo pone la “`l`” aparece el tamaño de los archivos en bytes y si le añades la “`h`” el tamaño aparece el formato KB, MB y GB el cual es más fácil de entender.

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

```
user@user-dani:~$ ls -R ~  
/home/user:  
new_script  USUARIO_ACTUAL  
user@user-dani:~$ _
```

La opción “-R” sirve para listar todos los subdirectorios dentro de ese directorio, podría ser peligroso hacerlo en el directorio raíz (/) ya que este contiene muchos datos y crearía una lista muy larga en la terminal.

4.- Manipulación de Archivos y Directorios

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

```
user@user-dani:/$ mkdir -p ~/proyecto/{src,doc,bin}
user@user-dani:/$ ls ~
'dos palabras.txt'  mi_bin  new_script  proyecto  test_locate.txt  USUARIO_ACTUAL
user@user-dani:/$ ls ~/proyecto
bin  doc  src
```

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

```
user@user-dani:/$ touch ~/notas.txt
user@user-dani:/$ ls ~
'dos palabras.txt'  mi_bin  new_script  notas.txt  proyecto  test_locate.txt  USUARIO_ACTUAL
user@user-dani:/$ mv ~/notas.txt ~/proyecto/doc/README.md
user@user-dani:/$ ls ~/proyecto/doc
README.md
```

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

```
user@user-dani:/$ cp ~/proyecto/doc/README.md ~/proyecto/bin/
user@user-dani:/$ ls ~/proyecto/bin
README.md
user@user-dani:/$ rm ~/proyecto/doc/README.md
user@user-dani:/$ ls ~/proyecto/doc
user@user-dani:/$ _
```

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.

```
user@user-dani:/$ rmdir ~/proyecto
rmdir: failed to remove '/home/user/proyecto': Directory not empty
```

Da un error al intentar borrarlo ya que este directorio contiene archivos, para poder borrarlo necesitas borrar todos los archivos que contiene o si no forzar el borrado para que se borre directamente

```
user@user-dani:/$ rm -r ~/proyecto
user@user-dani:/$ ls ~
'dos palabras.txt'  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL
```

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.

```
user@user-dani:/$ cd /etc
user@user-dani:/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?

```
user@user-dani:/$ 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-client-debug.log.1412: Cannot open: Permission denied
tar: /var/log/installer/subiquity-server-info.log.1462: Cannot open: Permission denied
tar: /var/log/installer/cloud-init-output.log: Cannot open: Permission denied
tar: /var/log/installer/subiquity-client-info.log.1412: Cannot open: Permission denied
tar: /var/log/installer/curtin-install/subiquity-initial.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-curthooks.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-curtin-apt.conf: Cannot open: Permission denied
tar: /var/log/installer/subiquity-server-debug.log.1462: Cannot open: Permission denied
tar: /var/log/installer/cloud-init.log: Cannot open: Permission denied
tar: /var/log/installer/autoinstall-user-data: Cannot open: Permission denied
tar: /var/log/installer/installer-journal.txt: Cannot open: Permission denied
tar: Exiting with failure status due to previous errors
user@user-dani:/$ ls ~
'dos palabras.txt'  log_backup.tar  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL
```

Aparece un aviso de permiso denegado para leer algunos archivos o directorios sin privilegios de administrador

```
user@user-dani:/$ sudo tar -cf ~/log_backup.tar /var/log
[sudo] password for user:
tar: Removing leading '/' from member names
tar: /var/log/journal/ad6156e490d94c0ab5934b5da846462c/user-1000.journal: file changed as we read it
user@user-dani:/$ ls ~
'dos palabras.txt'  log_backup.tar  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL
```

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

```
user@user-dani:/$ ls -lh ~/log_backup.tar
-rw-r--r-- 1 root root 19M oct 20 09:17 /home/user/log_backup.tar
user@user-dani:/$ gzip ~/log_backup.tar
user@user-dani:/$ ls -lh ~/log_backup.tar.gz
-rw-r--r-- 1 user user 622K oct 20 09:17 /home/user/log_backup.tar.gz
```

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

```
user@user-dani:/$ tar -tzf ~/log_backup.tar.gz_
```

```
var/log/unattended-upgrades/unattended-upgrades.log
var/log/unattended-upgrades/unattended-upgrades-dpkg.log
var/log/dist-upgrade/
var/log/cloud-init-output.log
var/log/sysstat/
var/log/sysstat/sa15
var/log/sysstat/sa15
var/log/sysstat/sa16
var/log/sysstat/sa20
var/log/dpkg.log
var/log/bootstrap.log
var/log/landscape/
var/log/landscape/sysinfo.log
var/log/cloud-init.log
var/log/faillog
var/log/fontconfig.log
var/log/apt/
var/log/apt/history.log
var/log/apt/term.log
var/log/apt/eipp.log.xz
var/log/appport.log
var/log/README
var/log/installer/
var/log/installer/block/
var/log/installer/block/probe-data.json
var/log/installer/block/discover.log
var/log/installer/subiquity-server-debug.log
var/log/installer/curtin-install.log
var/log/installer/subiquity-client-debug.log.1412
var/log/installer/subiquity-server-info.log.1462
var/log/installer/subiquity-client-info.log
var/log/installer/cloud-init-output.log
var/log/installer/subiquity-client-info.log.1412
var/log/installer/curtin-install/
var/log/installer/curtin-install/subiquity-initial.conf
var/log/installer/curtin-install/subiquity-partitioning.conf
var/log/installer/curtin-install/subiquity-curthooks.conf
var/log/installer/curtin-install/subiquity-extract.conf
var/log/installer/curtin-install/subiquity-curtin-apt.conf
var/log/installer/media-info
var/log/installer/subiquity-server-debug.log.1462
var/log/installer/subiquity-client-debug.log
var/log/installer/casper-md5check.json
var/log/installer/cloud-init.log
var/log/installer/autoinstall-user-data
var/log/installer/device-map.json
var/log/installer/subiquity-server-info.log
var/log/installer/installer-journal.txt
var/log/dmesg
```

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

```
user@user-dani:/$ tar -xzf ~/log_backup.tar.gz -C /tmp syslog
tar: syslog: Not found in archive
tar: Exiting with failure status due to previous errors
user@user-dani:/$ tar -xzf ~/log_backup.tar.gz -C /tmp messages
tar: messages: Not found in archive
tar: Exiting with failure status due to previous errors
```

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

```

user@user-dani:~$ touch a.txt b.log c.jpg
user@user-dani:~$ ls
a.txt  b.log  c.jpg  'dos palabras.txt'  log_backup.tar.gz  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL
user@user-dani:~$ zip archivos.zip a.txt b.log c.jpg
  adding: a.txt (stored 0%)
  adding: b.log (stored 0%)
  adding: c.jpg (stored 0%)
user@user-dani:~$ ls
archivos.zip  a.txt  b.log  c.jpg  'dos palabras.txt'  log_backup.tar.gz  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL

```

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

```

user@user-dani:~$ ls
archivos.zip  a.txt  b.log  c.jpg  'dos palabras.txt'  log_backup.tar.gz  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL
user@user-dani:~$ rm a.txt b.log c.jpg
user@user-dani:~$ ls
archivos.zip  'dos palabras.txt'  log_backup.tar.gz  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL
user@user-dani:~$ unzip archivos.zip
Archive:  archivos.zip
  extracting: a.txt
  extracting: b.log
  extracting: c.jpg
user@user-dani:~$ ls
archivos.zip  a.txt  b.log  c.jpg  'dos palabras.txt'  log_backup.tar.gz  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL

```

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.

Tuve que crear un archivo ya que en “/var/log” no habia ningún archivo comprimido, además, lo cree en “/var/tmp” ya que en “/var/log” no tenía permisos para crear ni editar

```

user@user-dani:/$ ls /var/tmp
prueba.log.gz
systemd-private-054c1b60687c4d5b8f18ca63c7fad9f3-fwupd.service-ToHg8u
systemd-private-054c1b60687c4d5b8f18ca63c7fad9f3-ModemManager.service-46CsJ0
systemd-private-054c1b60687c4d5b8f18ca63c7fad9f3-polkit.service-5zFnL4
user@user-dani:/$ zcat /var/tmp/prueba.log.gz
Este es un log de prueba para zcat

```

6.- Redirección, Tuberías y Filtros

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

```
user@user-dani:~$ ls
archivos.zip  a.txt  b.log  c.jpg  'dos palabras.txt'  log_backup.tar.gz  mi_bin  new_script  test_locate.txt  USUARIO_ACTUAL
user@user-dani:~$ ls -l ~ > mis_archivos.txt
user@user-dani:~$ ls
archivos.zip  a.txt  b.log  c.jpg  'dos palabras.txt'  log_backup.tar.gz  mi_bin  mis_archivos.txt  new_script  test_locate.txt  USUARIO_ACTUAL
```

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

```
user@user-dani:~$ date >> mis_archivos.txt
```

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

```
user@user-dani:~$ ls -l /etc | grep '^d' | wc -l
114
```

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

```
user@user-dani:~$ tail -n 10 /etc/passwd | cut -d':' -f1
polkitd
syslog
uuid
tcpdump
tss
landscape
fwupd-refresh
usbmux
sshd
user
```

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.

```
user@user-dani:~$ ps aux --sort=-%cpu | head -n 5
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root      6815  0.3  0.0      0     0 ?        I    07:00    0:03 [kworker/1:1-ata_sff]
root      5404  0.2  0.0      0     0 ?        I    05:05    0:16 [kworker/0:0-events]
root      6771  0.1  0.0      0     0 ?        I    06:50    0:02 [kworker/1:0-cgroup_destroy]
root       377  0.0  0.6 289116 27264 ?        Slsl oct20    0:06 /sbin/multipathd -d -s
```

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

El ">" sirve para sobrescribir el archivo y el ">>" sirve para añadir al final del archivo

```
user@user-dani:~$ echo "Ejemplo 1" > ejemplo.txt
user@user-dani:~$ cat ejemplo.txt
Ejemplo 1
user@user-dani:~$ echo "Ejemplo 2" >> ejemplo.txt
user@user-dani:~$ cat ejemplo.txt
Ejemplo 1
Ejemplo 2
user@user-dani:~$ echo "Ejemplo 3" > ejemplo.txt
user@user-dani:~$ cat ejemplo.txt
Ejemplo 3
```

Como se puede ver, usando únicamente ">" sobrescribe el archivo entero, en cambio, si usas ">>" lo añade sin sobrescribir todo el archivo entero

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.

```
user@user-dani:~$ find /etc -name "*.conf" > config_files.txt 2> errors.txt
user@user-dani:~$ ls
archivos.zip  b.log  config_files.txt  ejemplo.txt  log_backup.tar.gz  mis_archivos.txt  test_locate.txt
a.txt        c.jpg  'dos palabras.txt'  errors.txt  mi_bin  new_script  USUARIO_ACTUAL
user@user-dani:~$ cat errors.txt
find: '/etc/credstore.encrypted': Permission denied
find: '/etc/ssl/private': Permission denied
find: '/etc/polkit-1/rules.d': Permission denied
find: '/etc/credstore': Permission denied
find: '/etc/multipath': Permission denied
```


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.

```
user@user-dani:~$ nano mi_script.sh
```

```
GNU nano 7.2
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.

```
user@user-dani:~$ chmod u+x mi_script.sh
user@user-dani:~$ ./mi_script.sh
Usuario: user
Directorio actual: /home/user
```

Si intentas ejecutarlo como otro usuario que no tenga permisos, no podrá ejecutarlo a no ser que tenga permisos especiales ya que usando “u+x” solo da permiso de ejecución al propietario del archivo

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
```

```
user@user-dani:~$ ./mi_script.sh
user@user-dani:~$ ./mi_script.sh Hola
mundo
```

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

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

```
user@user-dani:~$ ./mi_script.sh Hola
mundo
user@user-dani:~$ ./mi_script.sh
Error: debes proporcionar un argumento.
user@user-dani:~$ ./mi_script.sh prueba
user@user-dani:~$
```

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

```
user@user-dani:~$ nano mi_script2.sh_
```

```
GNU nano 7.2
if [ "$1" = "$2" ]; then
    echo "iguales"
else
    echo "diferentes"
fi
```

```
user@user-dani:~$ ./mi_script2.sh 1 1
iguales
user@user-dani:~$ ./mi_script2.sh 1 2
diferentes
```

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.

```
user@user-dani:~$ nano mi_script3.sh_
```

```
GNU nano 7.2
if [ -z "$1" ]; then
    echo "Error: debes proporcionar un Directorio."
    exit 1
fi

for file in "$1"/*; do
    if [ -f "$file" ]; then
        mv "$file" "$file.bak"
    fi
done
```

Ahora voy a crear un directorio de prueba para este ejercicio

```
user@user-dani:~$ mkdir directorio_prueba
user@user-dani:~$ cd directorio_prueba
user@user-dani:~/directorio_prueba$ echo "Archivo 1" > archivo1.txt
user@user-dani:~/directorio_prueba$ echo "Archivo 2" > archivo2.txt
user@user-dani:~/directorio_prueba$ echo "Archivo 3" > archivo3.txt
user@user-dani:~/directorio_prueba$ ls
archivo1.txt  archivo2.txt  archivo3.txt
```

```
user@user-dani:~$ ./mi_script3.sh directorio_prueba
user@user-dani:~$ ls directorio_prueba
archivo1.txt.bak  archivo2.txt.bak  archivo3.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`).

```
user@user-dani:/$ 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.

```
user@user-dani:/$ ps aux | grep '^root' | wc -l
103
```

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`.

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

```
user@user-dani:/$ ls /etc/*[0-9]*_
```

```

/etc/e2scrub.conf /etc/mke2fs.conf /etc/sensors3.conf

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

/etc/gtk-3.0:
im-multipress.conf settings.ini

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

/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 K01uuidd

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

/etc/rc2.d:
S01appport S01console-setup.sh S01cron S01dbus S01grub-common S01open-vm-tools S01plymouth S01rsync S01ssh S01sysstat

/etc/rc3.d:
S01appport S01console-setup.sh S01cron S01dbus S01grub-common S01open-vm-tools S01plymouth S01rsync S01ssh S01sysstat

/etc/rc4.d:
S01appport S01console-setup.sh S01cron S01dbus S01grub-common S01open-vm-tools S01plymouth S01rsync S01ssh S01sysstat

/etc/rc5.d:
S01appport S01console-setup.sh S01cron S01dbus S01grub-common S01open-vm-tools S01plymouth S01rsync S01ssh S01sysstat

/etc/rc6.d:
K01cryptdisks K01cryptdisks-early K01iscsid K01open-iscsi K01open-vm-tools K01plymouth K01unattended-upgrades K01uuidd

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

/etc/X11:
rgb.txt xkb xorg.conf.d Xreset Xreset.d Xresources Xsession Xsession.d Xsession.options

```

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

```

user@user-dani:/$ find /usr/bin -type f -executable ! -user root

```

No aparece ningún archivo ya que de momento no cree ningun archivo ejecutable en ese directorio

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

Usando gzip:

```

user@user-dani:/$ sudo time gzip -k /var/log/syslog
0.00user 0.00system 0:00.00elapsed 100%CPU (0avgtext+0avgdata 1664maxresident)k
0inputs+72outputs (0major+144minor)pagefaults 0swaps
user@user-dani:/$ ls -lh /var/log/syslog.gz
-rw-r----- 1 syslog adm 34K oct 21 08:10 /var/log/syslog.gz

```

Usando bzip2:

```

user@user-dani:/$ sudo time bzip2 -k /var/log/syslog
0.01user 0.00system 0:00.01elapsed 100%CPU (0avgtext+0avgdata 2944maxresident)k
0inputs+64outputs (0major+480minor)pagefaults 0swaps
user@user-dani:/$ ls -lh /var/log/syslog.bz2
-rw-r----- 1 syslog adm 30K oct 21 08:14 /var/log/syslog.bz2

```

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.

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 ...]).

```

user@user-dani:~$ nano mi_script4.sh

```

```

GNU nano 7.2
if [ -f "$1" ];then
    echo "$1 es un archivo regular."
elif [ -d "$1" ]; then
    echo "$1 es un directorio."
else
    echo "$1 no existe o no es archivo ni directorio"
fi

```

```

user@user-dani:~$ ./mi_script4.sh /etc/passwd
/etc/passwd es un archivo regular.
user@user-dani:~$ ./mi_script4.sh /etc
/etc es un directorio.
user@user-dani:~$ ./mi_script4.sh /etc/no_existe_prueba
/etc/no_existe_prueba no existe o no es archivo ni directorio

```

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.

```

user@user-dani:~$ nano mi_script5.sh_

```

```

GNU nano 7.2
mkdir /test_dir 2>/dev/null

if [ $? -eq 0 ]; then
    echo "Directorio creado con exito."
else
    echo "Error: No se pudo crear el directorio."
fi

user@user-dani:~$ ./mi_script5.sh
Error: No se pudo crear 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.

```

user@user-dani:~$ nano mi_script6.sh_

GNU nano 7.2
for arg in "$@"; do
    if [[ "$arg" =~ ^[0-9]+$ ]] && [ "$arg" -gt 10 ]; then
        echo "$arg"
    fi
done

user@user-dani:~$ ./mi_script6.sh 1
user@user-dani:~$ ./mi_script6.sh 1 2 3
user@user-dani:~$ ./mi_script6.sh 1 2 3 10 11 12
11
12

```