Práctica 2 - ISO

1. Editor de textos:

(a) Nombre al menos 3 editores de texto que puede utilizar desde la línea de comandos.

vi, vim, neovim (nvim), nano y emacs.

(b) ¿En qué se diferencia un editor de texto de los comandos cat, more o less? Enumere los modos de operación que posee el editor de textos vi.

Los editores de texto permiten editar los archivos de texto. cat, more, less solo permiten visualizarlos.

vi tiene tres modos:

- insert mode: este modo se usa para la edición de texto normal. El modo reemplazar es una variación del modo insertar que reemplaza texto en lugar de insertarlo.
- command mode: este modo se usa para la exploración de archivos, copiar y pegar, y ejecutar comandos simples. Con este modo, también se realizan funciones como deshacer, rehacer y otras.
- ex mode: este modo se usa para guardar, cerrar y abrir archivos, así como también para buscar y reemplazar, y otras operaciones más complejas. Desde este modo, es posible insertar el resultado de programas en el archivo actual, configurar vim, etc. Todo lo que es posible usando ex se puede hacer desde este modo.

(c) Nombre los comandos más comunes que se le pueden enviar al editor de textos vi.

En ex mode:

Comando	Resultado
`:wq`	Guarda y cierra el archivo actual.
`:x`	Guarda el archivo actual si hay cambios sin guardar, y luego lo cierra.
`:w`	Guarda el archivo actual y permanece en el editor.
`:w <filename>`	Guarda el archivo actual bajo un nombre de archivo diferente.
`:q`	Cierra el archivo actual (solo si no hay cambios sin guardar).
`:q!`	Cierra el archivo actual, e ignora los cambios no guardados.

En command mode:

Tecla	Resultado
`i`	Cambia al modo *insertar* y comienza a insertar *antes* de la posición actual del cursor (insertar).
`a`	Cambia al modo *insertar* y comienza a insertar *luego* de la posición actual del cursor (anexar).
`I`	Mueve el cursor hasta el *inicio* de la línea actual y cambia al modo *insertar*.
`A`	Mueve el cursor hasta el *final* de la línea actual y cambia al modo *insertar*.
`R`	Cambia al modo *replace*, y comienza en el carácter bajo el cursor. En el modo *replace*, no se inserta texto, sino que cada carácter que ingresa reemplaza a un carácter del documento actual. (`vim` y `vi` también vienen con comandos de reemplazo más potentes; estos se analizan en otra sección.)
`o`	Abra una nueva línea *debajo* de la actual y cambie inmediatamente al modo *insertar*.
`O`	Abra una nueva línea *arriba* de la actual y cambie al modo *insertar*.

2. Proceso de Arranque SystemV (https://github.com/systeminit/si):

(a) Enumere los pasos del proceso de inicio de un sistema GNU/Linux, desde que se prende la PC hasta que se logra obtener el login en el sistema.

1. Se empieza a ejecutar el código del BIOS
2. El BIOS ejecuta el POST
3. El BIOS lee el sector de arranque (MBR)
4. Se carga el gestor de arranque (MBC)
5. El bootloader carga el kernel y el initrd
6. Se monta el initrd como sistema de archivos raíz y se inicializan componentes esenciales (ej.: scheduler)
7. El Kernel ejecuta el proceso init y se desmonta el initrd
8. Se lee el /etc/inittab
9. Se ejecutan los scripts apuntados por el runlevel 1
10. El final del runlevel 1 le indica que vaya al runlevel por defecto
11. Se ejecutan los scripts apuntados por el runlevel por defecto
12. El sistema está listo para usarse

(b) Proceso INIT. ¿Quién lo ejecuta? ¿Cuál es su objetivo?

El proceso INIT es ejecutado por el kernel, una vez que este ha sido cargado en memoria, y su función es cargar todos los subprocesos necesarios para el correcto funcionamiento del SO. Es el encargado de montar los filesystems y de hacer disponible los demás dispositivos.

El proceso init posee el PID 1 y se encuentra en /sbin/init (en SysV se lo configura a través del archivo /etc/inittab). No tiene padre y es el padre de todos los procesos (pstree).

(c) RunLevels. ¿Qué son? ¿Cuál es su objetivo?

Los RunLevels son los modos en que arranca Linux, y cada uno de estos es responsable de levantar (iniciar) o bajar (parar) una serie de servicios.

(d) ¿A qué hace referencia cada nivel de ejecución según el estándar? ¿Dónde se define qué Runlevel ejecutar al iniciar el sistema operativo? ¿Todas las distribuciones respetan estos estándares?

Según el estándar:

- 0: halt (parada)
- 1: single user mode (monousuario)
- 2: multiuser, without NFS (modo multiusuario sin soporte de red)
- 3: full multiuser mode console (modo multiusuario completo por consola)
- 4: no se utiliza
- 5: X11 (modo multiusuario completo con login gráfico basado en X)
- 6: reboot

En sistemas que utilizan el esquema SysVinit, el nivel de ejecución por defecto se especifica en el archivo /etc/inittab. No todas las distribuciones siguen estos estándares de runlevel de manera estricta. En los sistemas modernos, muchas distribuciones han reemplazado SysVinit con systemd, que utiliza "targets" en lugar de runlevels para definir los estados del sistema.

(e) Archivo /etc/inittab. ¿Cuál es su finalidad? ¿Qué tipo de información se almacena en él? ¿Cuál es la estructura de la información que en él se almacena?

El archivo /etc/inittab es un archivo de configuración cuyo como propósito principal es definir el nivel de ejecución (runlevel) por defecto al que el sistema debe arrancar, y especificar cómo se deben gestionar ciertos procesos del sistema durante el arranque y apagado.

Entonces:

- Define el runlevel por defecto que el sistema debe usar al iniciar.
- Especifica qué procesos se deben iniciar, gestionar y reiniciar en ciertos niveles de ejecución.
- Configura el comportamiento del sistema cuando cambian los runlevels.

El archivo contiene una lista de procesos que deben ejecutarse para diferentes runlevels y eventos. Cada línea en el archivo define un proceso o acción en el sistema.

Su estructura es: id:nivelesEjecución:acción:proceso

- Id: identifica la entrada en inittab (1 a 4 caracteres)
- NivelesEjecución: el/los niveles de ejecución en los que se realiza la acción
- Acción: describe la acción a realizar:
    - wait: inicia cuando entra al runlevel e init espera a que termine
    - initdefault
    - ctrlaltdel: se ejecutará cuando init reciba la señal SIGINT
    - off, respawn, once, sysinit, boot, bootwait, powerwait, etc.
- Proceso: el proceso exacto que será ejecutado

(f) Suponga que se encuentra en el runlevel <X>. Indique qué comando(s) ejecutaría para cambiar al runlevel <Y>. ¿Este cambio es permanente? ¿Por qué?

Para cambiar de un runlevel a otro se utiliza el comando init N, donde N es el número de runlevel al que se desea cambiar: *$ init <Y>*

Este cambio es temporal y solo afecta al estado actual del sistema. El cambio no es permanente porque el sistema volverá a su runlevel predeterminado en el próximo reinicio. Para hacer un cambio permanente en el runlevel predeterminado, se debe modificar el archivo de configuración /etc/inittab.

(g) Scripts RC. ¿Cuál es su finalidad? ¿Dónde se almacenan? Cuando un sistema GNU/Linux arranca o se detiene se ejecutan scripts, indique cómo determina qué script ejecutar ante cada acción. ¿Existe un orden para llamarlos? Justifique.

Cuando init entra en un runlevel, llama al script rc con un argumento numérico especificando el nivel de ejecución al que ir. Entonces, el script rc inicia y detiene servicios en el sistema según sea necesario para llevar el sistema a ese nivel de ejecución.

Todos los scripts RC se encuentran en el directorio /etc/rc.d/, que contiene un subdirectorio para cada nivel de ejecución (rc0.d, …, rc6.d). Dentro de cada uno de estos subdirectorios hay enlaces simbólicos a los scripts maestros almacenados en /etc/rc.d/init.d/.

Los enlaces simbólicos se nombran con el formato: [S|K]<orden><nombreScript>.

Los archivos que comienzan con S mayúscula representan scripts que se inician al entrar en ese nivel de ejcución, mientras que los archivos que comienzan con una K mayúscula representan scripts que se detienen. Los números especifican el orden en que deben ejecutarse los scripts.

Por ejemplo, un demonio puede tener un script llamado S35daemon en rc3.d/, y un script llamado K65daemon para detenerlo en rc2.d/. Tener los números al principio del nombre del archivo hace que se ordenen, y se procesen, en el orden deseado.

3. SystemD(https://github.com/systemd/systemd):

(a) ¿Qué es sytemd?

systemd es un sistema que centraliza la administración de daemons y librerias del sistema. El daemon systemd reemplaza al proceso init (systemd pasa a tener el PID 1).

systemd mejora el paralelismo de booteo y es compatible con SystemV, si es llamado como init, los runlevels son reemplazados por targets, y al igual que con upstart, el archivo /etc/inittab no existe más.

(b) ¿A qué hace referencia el concepto de Unit en SystemD?

Las unidades de trabajo son denominadas units de tipo:

- Service: controla un servicio particular (.service).
- Socket: encapsula IPC, un sockect del sistema o file system FIFO (.socket) → sockect-based activation.
- Target: agrupa units o establece puntos de sincronización durante el booteo (.target) → dependencia de unidades
- Snapshot: almacena el estado de un conjunto de unidades que puede ser restablecido más tarde (.snapshot), etc.

Estas units pueden tener dos estados: active o inactive.

(c) ¿Para qué sirve el comando systemctl en SystemD?

- Administrar servicios: se puede iniciar, detener, reiniciar, habilitar y deshabilitar servicios del sistema.
- Ver el estado de un servicio: mostrar si un servicio está activo o inactivo, proporcionado información de errores o fallos.
- Cambiar el estado del sistema: sería como cambiar el runlevel.
- Gestionar el arranque y apagado del sistema.

(d) ¿A qué hace referencia el concepto de target en SystemD?

Un target en systemd es una forma flexible de agrupar unidades y definir el estado del sistema. Reemplazan los runlevels tradicionales de SysVinit y permiten a los administradores configurar con precisión qué servicios o unidades se deben iniciar, detener o reiniciar cuando se cambia el estado del sistema.

(e) Ejecute el comando pstree. ¿Qué es lo que se puede observar a partir de la ejecución de este comando?

pstree muestra los procesos en ejecución en forma de árbol. El árbol tiene su raíz en init o systemd, depende del sistema de inicio que utilice el sistema.

4. Usuarios:

(a) ¿Qué archivos son utilizados en un sistema GNU/Linux para guardar la información de los usuarios?

El archivo /etc/passwd se utiliza para almacenar información sobre los usuarios locales y para cada usuario hay siete campos separados por dos puntos: username:password:uid:gid:GECOS:/home/dir:shell

1. username es el nombre del usuario (login).

2. password es donde se guardaban las contraseñas en formato cifrado tradicionalmente. Actualmente, se guardan cifradas en un archivo aparte con el nombre /etc/shadow (esto se indica colocando una x en el campo).

3. UID es un ID único de usuario, un número que identifica al usuario en el nivel más básico de forma unívoca.

4. GID es el número de ID de grupo principal del usuario.

5. GECOS es un texto arbitrario que, por lo general, incluye el nombre real del usuario y otra información adicional (mail, teléfono, etc.).

6. /home/dir es la ubicación donde se encuentran los datos personales del usuario y los archivos de configuración.

7. shell es la shell por defecto de los procesos y usuarios. La shell /sbin/nologin se utiliza para bloquear el inicio de sesión en el sistema de forma interactiva y es muy común utilizarla en cuentas de usuarios que representan procesos o servicios en lugar de usuarios humanos.

(b) ¿A qué hacen referencia las siglas UID y GID? ¿Pueden coexistir UIDs iguales en un sistema GNU/Linux? Justifique.

- UID (User ID): es un identificador único de usuario. Cada usuario tiene su propio UID, que es un número que lo identifica de manera unívoca. Por convención, muchas distribuciones de GNU/Linux asignan por defecto el UID 1000 al primer usuario creado en el sistema y luego asignan a los usuarios nuevos el primer número de UID disponible en el rango, a partir de la UID 1000 en adelante (a menos que se especifique uno explícitamente).
- GID (Group ID): es un identificador único de grupo. Cada grupo de usuarios tiene su propio GID, para identificarlo de manera unívoca.
    - Los grupos locales están definidos en /etc/group.
    - Cada usuario tiene exactamente un grupo principal y pueden ser miembros de ninguno o más grupos adicionales.
    - Para los usuarios locales, el grupo principal está definido por el número de GID del grupo indicado en el cuarto campo de /etc/passwd.
    - Generalmente, el grupo principal es propietario de los nuevos archivos creados por el usuario.
    - Normalmente, el grupo principal de un usuario creado recientemente es un grupo creado con el mismo nombre que el del usuario. El usuario es el único miembro de este grupo privado de usuarios (UPG).

(c) ¿Qué es el usuario root? ¿Puede existir más de un usuario con este perfil en GNU/Linux? ¿Cuál es la UID del root?

root es un superusuario, un usuario que tiene todo el poder sobre el sistema. Este usuario tiene el poder de anular los privilegios normales del sistema de archivos y se utiliza para manejar y administrar el sistema. UID 0 siempre se asigna a la cuenta de root.

Es necesario contar con privilegios de superusuario (root) para poder realizar tareas, como la instalación o eliminación de software, y para administrar los directorios y los archivos del sistema. La mayoría de los dispositivos solo pueden ser controlados por el usuario root, pero existen algunas excepciones (por ejemplo, los dispositivos desmontables, como los dispositivos USB). El usuario root tiene poder ilimitado para dañar el sistema: eliminar archivos y directorios, eliminar cuentas de usuarios, agregar puertas traseras, etc.

(d) Agregue un nuevo usuario llamado iso2017 a su instalación de GNU/Linux, especifique que su home sea creada en /home/iso_2017, y hágalo miembro del grupo catedra (si no existe, deberá crearlo). Luego, sin iniciar sesión como este usuario cree un archivo en su home personal que le pertenezca. Luego de todo esto,

borre el usuario y verifique que no queden registros de él en los archivos de información de los usuarios y grupos.

```
root@c04685c6d964:/ISO# groupadd catedra
root@c04685c6d964:/ISO# sudo useradd -d "/home/iso/2017" -m iso2017
root@c04685c6d964:/ISO# sudo usermod -a -G catedra iso2017
root@c04685c6d964:/ISO#
root@c04685c6d964:/ISO# cat /etc/passwd | grep "iso2017"
Dockerfile          Practica-1/         docker-compose.yaml
root@c04685c6d964:/ISO# cat /etc/passwd | grep "iso2017" && cat /etc/group | grep "catedra"
iso2017:x:1000:1001::/home/iso/2017:/bin/sh
catedra:x:1000:iso2017
root@c04685c6d964:/ISO# cd /home/iso/2017:/bin/sh
bash: cd: /home/iso/2017:/bin/sh: No such file or directory
root@c04685c6d964:/ISO# cd /home/iso/2017:/bin/sh
bash: cd: /home/iso/2017:/bin/sh: No such file or directory
root@c04685c6d964:/ISO# cd /home/iso_2017
bash: cd: /home/iso_2017: No such file or directory
root@c04685c6d964:/ISO# cd /home/iso2017
bash: cd: /home/iso2017: No such file or directory
root@c04685c6d964:/ISO# cd
root@c04685c6d964:~#
What's next:
    Try Docker Debug for seamless, persistent debugging tools in any container or image → docker debug c04685c6d964
    Learn more at https://docs.docker.com/go/debug-cli/
PS C:\Users\matia\OneDrive\Escritorio\Matias\Facultad\4to Semestre\ISO\Practicas\Resoluciones> docker exec -it c04685c6d964 bash
root@c04685c6d964:/ISO# cd /home/iso/2017
root@c04685c6d964:/home/iso/2017# touch ejemplo.txt
root@c04685c6d964:/home/iso/2017# cat /etc/passwd | grep "iso2017"
iso2017:x:1000:1001::/home/iso/2017:/bin/sh
root@c04685c6d964:/home/iso/2017# userdel iso2017
root@c04685c6d964:/home/iso/2017# cat /etc/passwd | grep "iso2017"
root@c04685c6d964:/home/iso/2017#
```

(e) Investigue la funcionalidad y parámetros de los siguientes comandos:

- useradd ó adduser:

```
leo@leo:~$ useradd --help
Usage: useradd [options] LOGIN
       useradd -D
       useradd -D [options]

Options:
      --badname                 do not check for bad names
  -b, --base-dir BASE_DIR       base directory for the home directory of the
                                new account
      --btrfs-subvolume-home    use BTRFS subvolume for home directory
  -c, --comment COMMENT         GECOS field of the new account
  -d, --home-dir HOME_DIR       home directory of the new account
  -D, --defaults                print or change default useradd configuration
  -e, --expiredate EXPIRE_DATE  expiration date of the new account
  -f, --inactive INACTIVE       password inactivity period of the new account
  -F, --add-subids-for-system   add entries to sub[ud]id even when adding a system user
  -g, --gid GROUP               name or ID of the primary group of the new
                                account
  -G, --groups GROUPS           list of supplementary groups of the new
                                account
  -h, --help                    display this help message and exit
  -k, --skel SKEL_DIR           use this alternative skeleton directory
  -K, --key KEY=VALUE           override /etc/login.defs defaults
  -l, --no-log-init             do not add the user to the lastlog and
                                faillog databases
  -m, --create-home             create the user's home directory
  -M, --no-create-home          do not create the user's home directory
  -N, --no-user-group           do not create a group with the same name as
                                the user
  -o, --non-unique              allow to create users with duplicate
                                (non-unique) UID
  -p, --password PASSWORD       encrypted password of the new account
  -r, --system                  create a system account
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -s, --shell SHELL             login shell of the new account
  -u, --uid UID                 user ID of the new account
  -U, --user-group              create a group with the same name as the user
  -Z, --selinux-user SEUSER     use a specific SEUSER for the SELinux user mapping
      --extrausers              Use the extra users database
```

- usermod:

```
leo@leo:~$ usermod --help
Usage: usermod [options] LOGIN

Options:
  -a, --append                  append the user to the supplemental GROUPS
                                mentioned by the -G option without removing
                                the user from other groups
  -b, --badname                 allow bad names
  -c, --comment COMMENT         new value of the GECOS field
  -d, --home HOME_DIR           new home directory for the user account
  -e, --expiredate EXPIRE_DATE  set account expiration date to EXPIRE_DATE
  -f, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -g, --gid GROUP               force use GROUP as new primary group
  -G, --groups GROUPS           new list of supplementary GROUPS
  -h, --help                    display this help message and exit
  -l, --login NEW_LOGIN         new value of the login name
  -L, --lock                    lock the user account
  -m, --move-home               move contents of the home directory to the
                                new location (use only with -d)
  -o, --non-unique              allow using duplicate (non-unique) UID
  -p, --password PASSWORD       use encrypted password for the new password
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -r, --remove                  remove the user from only the supplemental GROUPS
                                mentioned by the -G option without removing
                                the user from other groups
  -R, --root CHROOT_DIR         directory to chroot into
  -s, --shell SHELL             new login shell for the user account
  -u, --uid UID                 new UID for the user account
  -U, --unlock                  unlock the user account
  -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
```

- userdel:

```
leo@leo:~$ userdel --help
Usage: userdel [options] LOGIN

Options:
  -f, --force                   force some actions that would fail otherwise
                                e.g. removal of user still logged in
                                or files, even if not owned by the user
  -h, --help                    display this help message and exit
  -r, --remove                  remove home directory and mail spool
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
      --extrausers              Use the extra users database
  -Z, --selinux-user            remove any SELinux user mapping for the user
```

- su:

```
leo@leo:~$ su --help

Usage:
 su [options] [-] [<user> [<argument>...]]

Change the effective user ID and group ID to that of <user>.
A mere - implies -l.  If <user> is not given, root is assumed.

Options:
 -m, -p, --preserve-environment      do not reset environment variables
 -w, --whitelist-environment <list>  don't reset specified variables

 -g, --group <group>            specify the primary group
 -G, --supp-group <group>       specify a supplemental group

 -, -l, --login                 make the shell a login shell
 -c, --command <command>        pass a single command to the shell with -c
 --session-command <command>    pass a single command to the shell with -c
                                  and do not create a new session
 -f, --fast                     pass -f to the shell (for csh or tcsh)
 -s, --shell <shell>            run <shell> if /etc/shells allows it
 -P, --pty                      create a new pseudo-terminal

 -h, --help                     display this help
 -V, --version                  display version

For more details see su(1).
```

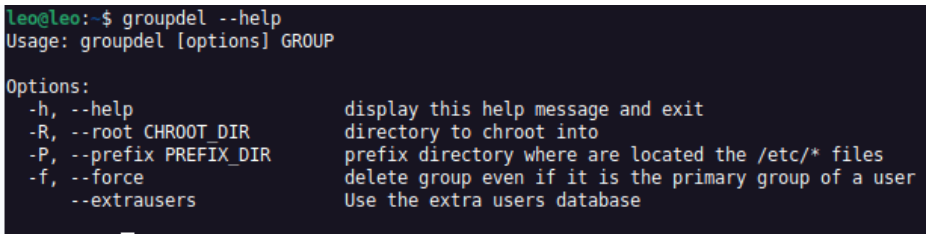
- groupadd:

```
leo@leo:~$ groupadd --help
Usage: groupadd [options] GROUP

Options:
  -f, --force                   exit successfully if the group already exists,
                                and cancel -g if the GID is already used
  -g, --gid GID                 use GID for the new group
  -h, --help                    display this help message and exit
  -K, --key KEY=VALUE           override /etc/login.defs defaults
  -o, --non-unique              allow to create groups with duplicate
                                (non-unique) GID
  -p, --password PASSWORD       use this encrypted password for the new group
  -r, --system                  create a system account
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DI        directory prefix
  -U, --users USERS             list of user members of this group
      --extrausers              Use the extra users database
```

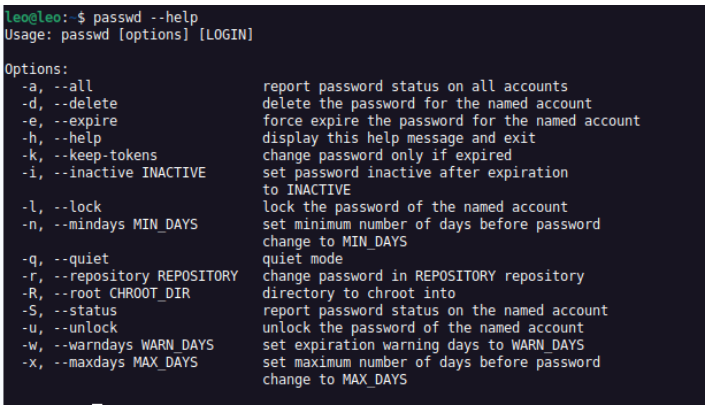
- who:

```
leo@leo:~$ who --help
Usage: who [OPTION]... [ FILE | ARG1 ARG2 ]
Print information about users who are currently logged in.

  -a, --all         same as -b -d --login -p -r -t -T -u
  -b, --boot        time of last system boot
  -d, --dead        print dead processes
  -H, --heading     print line of column headings
  -l, --login       print system login processes
      --lookup      attempt to canonicalize hostnames via DNS
  -m                only hostname and user associated with stdin
  -p, --process     print active processes spawned by init
  -q, --count       all login names and number of users logged on
  -r, --runlevel    print current runlevel
  -s, --short       print only name, line, and time (default)
  -t, --time        print last system clock change
  -T, -w, --mesg    add user's message status as +, - or ?
  -u, --users       list users logged in
      --message     same as -T
      --writable    same as -T
      --help        display this help and exit
      --version     output version information and exit

If FILE is not specified, use /var/run/utmp.  /var/log/wtmp as FILE is common.
If ARG1 ARG2 given, -m presumed: 'am i' or 'mom likes' are usual.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation <https://www.gnu.org/software/coreutils/who>
or available locally via: info '(coreutils) who invocation'
```

- groupdel:



- passwd:



5. FileSystem:

(a) ¿Cómo son definidos los permisos sobre archivos en un sistema GNU/Linux?

Los archivos tienen 3 categorías de usuario a las que se le aplican permisos. El archivo pertenece a un usuario, que generalmente es quien creó el archivo. El archivo también pertenece a un solo grupo, generalmente el grupo primario del usuario que creó el archivo, pero esto se puede cambiar. Se pueden establecer diferentes permisos para el usuario propietario y el grupo propietario, así como para todos los otros usuarios en el sistema que no sean el usuario o un miembro del grupo propietario. Se aplicarán los permisos más específicos. Por lo tanto, los permisos de usuario anulan los permisos de grupo, que anulan otros permisos.

Categoría de permisos:

<u>- Lectura/Read (r):</u>

- En archivos: Pueden leerse los contenidos del archivo.
- En directorios: Permite detallar los contenidos del directorio.

<u>- Escritura/Write (w):</u>

- En archivos: Pueden modificarse los contenidos del archivo.
- En directorios: Pueden crearse o eliminar archivos en el directorio.

<u>- Ejecución/Execute (x):</u>

- En archivos: Se pueden ejecutar archivos como comandos.
- En directorios: Es posible acceder al contenido del directorio. El primer carácter indica el tipo del archivo:

```
leo@leo:~/Documents/GitHub/Facultad/Segundo/Segundo semestre/ISO/Prácticas/Práctica 2$ ls -l
total 1232
-rw-rw-r-- 1 leo leo 314335 Aug 30 18:38 'Explicación Práctica 2.pdf'
-rw-rw-r-- 1 leo leo 254686 Aug 21 19:27 'Practica 2.pdf'
-rw-rw-r-- 1 leo leo 685922 Sep  9 17:41 'Resolución ISO Práctica 2.pdf'
```

El primer carácter indica el tipo del archivo:

- "-" para archivos normales.
- "d" para directorios.
- "l" para enlaces simbólicos (symlinks), entre otros.

Los siguientes 9 caracteres indican los permisos de los grupos:

- Primer grupo: Permisos del propietario.
- Segundo grupo: Permisos del grupo.
- Tercer grupo: Permisos para otros usuarios.

En el caso del archivo "Practica 2.pdf":

- El primer carácter "-" indica que es un archivo normal.
- Los siguientes 3 caracteres (rw-) indican que el propietario tiene permiso de lectura y escritura.
- Los siguientes 3, (rw-) indican que el grupo tiene permiso de lectura y escritura.
- Los últimos 3 (r--) indican que los otros usuarios tienen permiso de lectura.
- El "1" indica el número de enlaces al archivo o directorio.
- "leo" indica el usuario propietario.
- "leo" indica el grupo al que pertenece el archivo o directorio.
  ↳ Mismo nombre debido al "User Private Group".
- "254686" indica el peso en bytes.
- "Aug 21 19:27" indica la fecha y hora de la última modificación.
- "Practica 2.pdf", finalmente, indica el nombre.

(b) Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con los permisos en GNU/Linux:

- chmod
- chown
- chgrp

(c) Al utilizar el comando chmod generalmente se utiliza una notación octal asociada para definir permisos. ¿Qué significa esto? ¿A qué hace referencia cada valor?

* chmod (Change Mode): Se utiliza para cambiar los permisos de un archivo o directorio. En GNU/Linux, los permisos definen qué usuarios pueden leer, escribir o ejecutar un archivo o directorio.

- chmod [OPTIONS] [PERMISSIONS] FILE}

Modo simbólico: Se utilizan letras para representar los permisos:

- r para lectura (read).

- w para escritura (write).

- x para ejecución (execute).

Y utiliza la siguiente sintaxis:

- u para el usuario.

- g para el grupo.

- o para otros.

Modo numérico: Los permisos se representan con números. El sistema de permisos se basa en valores octales:

- 4 para la lectura.

- 2 para la escritura.

- 1 para la ejecución.

Estos valores se suman para asignar un permiso, por ejemplo:

Lectura+Escritura > 2+4 = 6

Lectura+Escritura+Ejecución > 2+4+1 = 7

Lectura > 2 = 4

* chown (Change Ownership): Se utiliza para cambiar el propietario (usuario) y/o el grupo asociado a un archivo o directorio.

- chown [OPTIONS] [USER][:[GROUP]] FILE

Opciones más comunes:

-R, cambia el propietario y grupo de manera recursiva en todos los subdirectorios y archivos.

--reference=archivo, cambia el propietario y grupo de un archivo utilizando los mismos que tiene otro archivo de referencia.

* chgrp (Change Group): Se utiliza para cambiar el grupo asociado a un archivo o directorio sin cambiar el propietario.

- chgrp [OPTIONS] [GROUP] FILE

(d) ¿Existe la posibilidad de que algún usuario del sistema pueda acceder a determinado archivo para el cual no posee permisos? Nombrelo, y realice las pruebas correspondientes.

El superusuario (root) tiene privilegios elevados y puede acceder a cualquier archivo o directorio en el sistema, independientemente de sus permisos.

(e) Explique los conceptos de “full path name” y “relative path name”. De ejemplos claros de cada uno de ellos.

Rutas absolutas (full path name):

Una ruta absoluta es un nombre completamente calificado que comienza en el directorio (/) raíz y especifica cada subdirectorio que se atraviesa para llegar y que representa en forma exclusiva un solo archivo. Cada archivo del sistema de archivos tiene un único nombre de ruta absoluta, reconocido con una regla simple: un nombre de archivo con una barra (/) como primer carácter es el nombre de la ruta absoluta. Por ejemplo, el nombre de ruta absoluta para el archivo de registro de mensajes del sistema es /var/log/messages.

Rutas relativas (relative path name):

Al igual que una ruta absoluta, una ruta relativa identifica un archivo único y especifica solo la ruta necesaria para llegar al archivo desde el directorio de trabajo. Para reconocer nombres de ruta relativos, se sigue una regla simple: un nombre de ruta que no tenga otro carácter más que una barra (/) como primer carácter es un nombre de ruta relativo. Un usuario en el directorio /var podría referirse en forma relativa al archivo de registro del mensaje como log/messages.

./ representa al directorio de trabajo y ../ representa al directorio padre del directorio de trabajo actual. Por ejemplo, un usuario en el directorio /home/user/Documets/Foo/ puede hacer referencia al archivo /home/user/file.txt con la ruta relativa ./../../file.txt.

(f) ¿Con qué comando puede determinar en qué directorio se encuentra actualmente? ¿Existe alguna forma de ingresar a su directorio personal sin necesidad de escribir todo el path completo? ¿Podría utilizar la misma idea para acceder a otros directorios? ¿Cómo? Explique con un ejemplo.

El comando pwd (print working directory) imprime la ruta absoluta del directorio de trabajo actual. Para ingresar al directorio personal sin necesidad de escribir todo el path completo se puede utilizar “cd”.

También se puede utilizar “cd ~” para acceder al directorio personal o acortar las rutas. Utilizando “~” también se puede acceder al directorio personal de otro usuario, por ejemplo:

- cd ~pepe/Downloads

↳ Accede al directorio “Downloads” del usuario “pepe”

(g) Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el uso del FileSystem:

- cd
- umount
- mkdir
- du
- rmdir
- df
- mount
- ln
- ls
- pwd
- cp
- mv

cd	Cambiar el directorio de trabajo actual.

umount	Desmontar sistemas de archivos o dispositivos.

mkdir	Crear un directorio, si no existe.	
	-m, --mode=*MODE*	Establece los permisos del directorio MODE es un octal.
	-p, --parents	Crea directorios padres según sea necesario, con sus modos de archivo no afectados por ninguna opción -m.
	-v, --verbose	Imprime cada directorio creado.

du	Muestra el uso del espacio en disco por archivos y directorios.	
	-a, --all	Muestra todos los archivos, no solo directorios.
	-d, --max-depth=*N*	Imprime los valores hasta *N* niveles.
	-h, --human-readable	Imprime los tamaños en formato legible (1K, 234M, 2G).
	-c, --total	Imprime el peso total
	`$ du -a -d -h 1 dir | sort -h`	
rmdir	Elimina un directorio, si está vacío.	
	-p, --parents	Elimina el directorio y sus ancestros. "rmdir -p a/b/c" es equivalente a "rmdir a/b/c a/b a".
	-v, --verbose	Muestra un diagnóstico por cada directorio procesado

df	Muestra información sobre el espacio disponible y utilizado en los sistemas de archivos montados.	
	-h, --human-readable	Imprime los tamaños en formato legible (1K, 234M, 2G).

mount	Montar sistemas de archivos en un directorio específico

ln	Crea enlaces duros (por defecto) o simbólicos a archivos y directorios.	
	-s, --symbolic	Crea enlaces simbólicos `ln -s file link`

ls	Lista los archivos y directorios de un directorio	
	-a, --all	No ignora las entradas que empiezan con "."
	-A, --almost-all	Como -a pero no lista ./ y ../
	-d, --directory	Lista el directorio en sí, no su contenido
	-h, --human-readable	Imprime los tamaños en formato legible.
	-l	Utilizar un formato de listado largo
	-R, --recursive	Lista los subdirectorios recursivamente

pwd	Muestra la ruta completa del directorio de trabajo actual

cp	Copia archivos y directorios de una ubicación a otra.	
	-f, --force	Si no se puede abrir un archivo de destino existente, elimínelo e inténtelo de nuevo.
	-i, --interactive	Preguntar antes de sobrescribir.
	-p	Igual a `--preserve=mode,ownership,timestamps`.
	-R, -r, --recursive	Copia directorios recursivamente.
	-v, --verbose	Explicar lo que se está haciendo.
mv	Mueve o renombra archivos y directorios.	
	-f, --force	No preguntar antes de sobrescribir
	-i, --interactive	Preguntar antes de sobrescribir.
	-v, --verbose	Explicar lo que se está haciendo.

6. Procesos:

(a) ¿Qué es un proceso? ¿A qué hacen referencia las siglas PID y PPID? ¿Todos los procesos tienen estos atributos en GNU/Linux? Justifique. Indique qué otros atributos tiene un proceso.

Un proceso es un programa que se está ejecutando.

El PID es la identificación del proceso y el PPID es la identificación de un proceso padre. En Linux todos los procesos tienen un PPID excepto por el proceso systemd o init, que son el primer proceso en ejecutarse y es de donde que derivan todos los demás procesos.

Además del PID y el PPID los procesos tienen otros atributos cómo:

- Estado: en ejecución, en espera, suspendido, zombie.
- Espacio de memoria: cada proceso tiene su propio espacio de memoria virtual.
- Información sobre la propiedad del proceso: usuario y grupo propietario del proceso.
- Prioridad de planificación: Indica la prioridad del proceso en relación con otros procesos.

(b) Indique qué comandos se podrían utilizar para ver qué procesos están en ejecución en un sistema GNU/Linux.

El comando ps se utiliza para elaborar una lista de los procesos actuales. El comando puede proporcionar información detallada de los procesos, que incluye:

- La identificación del usuario (UID) que determina los privilegios del proceso.
- La identificación del proceso (PID) única.
- La CPU y el tiempo real empleado.
- La cantidad de memoria que el proceso ha asignado en diversas ubicaciones.
- La ubicación del proceso STDOUT, conocido como terminal de control.
- El estado del proceso actual.

De manera predeterminada, el comando ps sin opciones selecciona todos los procesos que tienen la misma identificación de usuario efectivo (EUID) que el usuario actual y que están asociados con la misma terminal en la que se invocó ps. Una lista de visualización común es ps aux (no es lo mismo que ps -aux) que muestra todos los procesos, con columnas que serán de interés para los usuarios, e incluye procesos sin un terminal de control.

(c) ¿Qué significa que un proceso se está ejecutando en Background? ¿Y en Foreground?

En sistemas GNU/Linux y Unix, los conceptos de foreground (primer plano) y background (segundo plano) describen cómo se ejecuta un proceso con respecto al usuario y la terminal.

Foreground:

- Un proceso que se ejecuta en primer plano está activo directamente en la terminal que lo inició. El proceso ocupa la terminal y recibe directamente las entradas del usuario (teclado), además de mostrar la salida (información) directamente en la misma.

- Mientras un proceso está en primer plano, el usuario no puede interactuar con la terminal para realizar otras acciones hasta que el proceso termine.

Ejemplo: Al utilizar “cat archivo.txt” el comando está en foreground y la terminal está ocupada con la ejecución de ese comando, no se puede utilizar hasta que el proceso finalice.

Background:

- Un proceso que se ejecuta en segundo plano se está ejecutando sin bloquear la terminal, lo que permite que el usuario continúe utilizando la terminal para otros comandos mientras el proceso sigue funcionando en segundo plano.

- Un proceso en segundo plano no recibe entradas del teclado directamente, pero sigue ejecutándose y produciendo resultados.

Ejemplo: Al utilizar un comando con el símbolo “&” al final, el proceso se ejecutará en segundo plano.

(d) ¿Cómo puedo hacer para ejecutar un proceso en Background? ¿Como puedo hacer para pasar un proceso de background a foreground y viceversa?

Para ejecutar un proceso en Background se puede agregar “&” al final:

- cat archivo.txt &

Para enviar un proceso de Foreground a Background se puede utilizar Ctrl + Z para suspenderlo temporalmente y después ejecutando “bg” para enviarlo al segundo plano. Para traerlo del Background se puede utilizar “fg”.

(e) Pipe (|). ¿Cuál es su finalidad? Cite ejemplos de su utilización.

Su función es encadenar comandos, permite usar la salida de un comando como entrada de otro, y procesar datos en etapas, facilita el procesamiento de datos en varias etapas, aplicando diferentes comandos a los datos sucesivamente.

Ejemplo: ls -l | wc -l # Lista detalladamente el contenido del directorio actual y cuenta las líneas.

(f) Redirección. ¿Qué tipo de redirecciones existen? ¿Cuál es su finalidad? Cite ejemplos de utilización.

Entrada estándar, salida estándar y error estándar:

Un proceso puede necesitar leer entradas desde alguna parte y escribir salidas en la pantalla o en archivos. Un comando ejecutado desde el aviso de shell normalmente lee su entrada desde el teclado y envía su salida a su ventana de terminal.

Un proceso utiliza canales numerados denominados descriptores de archivos para obtener entradas y enviar salidas. Todos los procesos tendrán al menos tres descriptores de archivo para comenzar.

- Entrada estándar (canal 0) lee entradas desde el teclado.
- Salida estándar (canal 1) envía una salida normal al terminal.
- Error estándar (canal 2) envía mensajes de error al terminal.

Si un programa abre conexiones independientes para otros archivos, puede usar descriptores de archivo con números superiores.

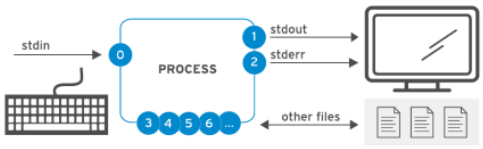


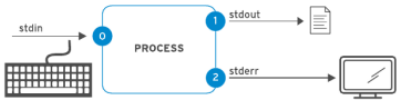
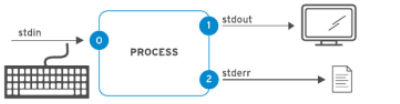
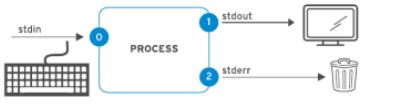
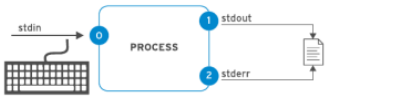
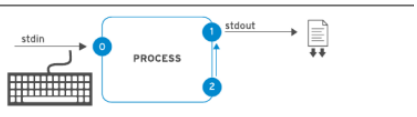
Figura 4.1: Canales de E/S de un proceso (descriptores de archivo)

#	Nombre	Descripción	Conexión predeterminada	Uso
0	**stdin**	Entrada estándar	Teclado	Solo lectura
1	**stdout**	Salida estándar	Terminal	Solo escritura
2	**stderr**	Error estándar	Terminal	Solo escritura
3+	**filename**	Otros archivos	Ninguno	Lectura y escritura

Redireccionamiento de la salida a un archivo:

El redireccionamiento de E/S reemplaza los destinos de canales predeterminados con nombres de archivos que representan dispositivos o archivos de salida. Con el uso del redireccionamiento, los mensajes de error y salida de un proceso que se envían generalmente a la ventana de terminal pueden capturarse como contenido de archivo, enviarse a un dispositivo o descartarse.

El redireccionamiento de stdout evita que la salida de un proceso aparezca en el terminal. Como se puede ver en la siguiente tabla, el redireccionamiento de únicamente stdout no evita que los mensajes de error stderr aparezcan en el terminal. Si el archivo no existe, se creará. Si el archivo existe y el redireccionamiento no es uno que se agregue al archivo, el contenido del archivo se sobrescribirá. El archivo especial /dev/null descarta discretamente la salida del canal redirigido a él y es siempre un archivo vacío.

Uso	Explicación	Ayuda visual
>file	redirigir **stdout** para sobrescribir un archivo	
>>file	redirigir **stdout** para agregar a un archivo	
2>file	redirigir **stderr** para sobrescribir un archivo	
2>/dev/null	descartar mensajes de error **stderr** mediante el redireccionamiento a **/dev/null**	
>file 2>&1 *&>file*	redirigir **stdout** y **stderr** para sobrescribir el mismo archivo	
>>file 2>&1 *&>>file*	redirigir **stdout** y **stderr** para agregar al mismo archivo	

(g) Comando kill. ¿Cuál es su funcionalidad? Cite ejemplos.

El comando kill en GNU/Linux y Unix se utiliza para enviar señales a procesos en ejecución, generalmente con el propósito de terminarlos o modificar su comportamiento. Aunque el nombre sugiere que solo sirve para “matar” procesos, puede enviar diversas señales para controlar los procesos.

Señales:

Señal	Número	Descripción	Uso común
`SIGHUP`	1	Hangup. Indica que el terminal fue cerrado o pide que se recargue la configuración.	Recargar configuraciones en procesos como `nginx` o `apache`.
`SIGINT`	2	Interrupt. Se envía cuando se presiona `Ctrl+C`.	Interrumpir un proceso en ejecución desde la terminal.
`SIGQUIT`	3	Quit. Se envía con `Ctrl+\`.	Terminar un proceso y generar un volcado de memoria (core dump).
`SIGKILL`	9	Kill. Mata un proceso inmediatamente. No puede ser ignorado.	Forzar la terminación de un proceso problemático.
`SIGTERM`	15	Terminate. Señal estándar para pedir que un proceso termine.	Terminar un proceso de manera suave y controlada.
`SIGCONT`	18	Continue. Restaura un proceso detenido con `SIGSTOP`.	Continuar un proceso que fue detenido.
`SIGSTOP`	19	Stop. Detiene un proceso sin terminarlo. No puede ser ignorado.	Pausar un proceso sin finalizarlo.
`SIGTSTP`	20	Stop signal. Se envía con `Ctrl+Z`.	Detener temporalmente un proceso en primer plano.
`SIGUSR1`	10	User-defined signal 1.	Señal definida por el usuario.
`SIGUSR2`	12	User-defined signal 2.	Señal definida por el usuario.
`SIGSEGV`	11	Segmentation fault. Indica un error de segmentación.	Generalmente usada por el sistema para indicar un error de memoria.

Ejemplos:

- kill 1234 # Envía SIGTERM al proceso con PID 1234.

- kill -9 1234 # Envía SIGKILL al proceso con PID 1234.

- kill -HUP 1234 # Envía SIGHUP al proceso con PID 1234.

- kill -u usuario # Envía SIGTERM a todos los procesos del usuario.

(h) Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el manejo de procesos en GNU/Linux. Además, compárelos entre ellos:

- ps
- kill
- pstree
- killall
- top
- nice

Comando	Funcionalidad	Parámetros principales	Comparación
`ps`	Muestra una instantánea de los procesos actuales en el sistema.	- `-e`: Muestra todos los procesos. - `-f`: Muestra una lista completa. - `-o`: Define las columnas que se muestran (como `pid`, `ppid`, `cmd`, etc.).	Proporciona una visión estática de los procesos en el momento de ejecución. Útil para listar y filtrar procesos según diferentes criterios. No muestra datos dinámicos como `top`.
`kill`	Envía señales a procesos, comúnmente para terminarlos.	- `-9`: Envía `SIGKILL`, termina un proceso inmediatamente. - `-15`: Envía `SIGTERM`, solicita terminar el proceso (por defecto). - `-l`: Lista todas las señales disponibles.	Mata procesos individualmente. Requiere el PID de un proceso, a diferencia de `killall`, que actúa sobre procesos por nombre.
`pstree`	Muestra los procesos en un formato de árbol, mostrando la jerarquía de procesos.	- `-p`: Muestra los PID junto con los nombres de los procesos. - `-u`: Muestra el propietario (usuario) de los procesos.	A diferencia de `ps`, muestra la relación padre-hijo entre los procesos, útil para visualizar cómo están relacionados jerárquicamente.
`killall`	Termina múltiples procesos que tienen el mismo nombre.	- `-9`: Mata todos los procesos con el nombre indicado. - `-I`: Ignora las diferencias entre mayúsculas y minúsculas al buscar procesos. - `-v`: Muestra más detalles sobre los procesos afectados.	Más efectivo que `kill` cuando hay varios procesos del mismo nombre que deben terminarse a la vez. No requiere conocer los PIDs específicos.
`top`	Muestra en tiempo real los procesos activos y el uso de recursos del sistema (CPU, memoria, etc.).	- `-d`: Define el tiempo de refresco (en segundos). - `-u [usuario]`: Muestra solo los procesos de un usuario específico. - `-n`: Define cuántos ciclos de actualización debe hacer antes de detenerse.	A diferencia de `ps`, que es estático, `top` proporciona una vista dinámica y actualizada en tiempo real. Es más útil para el monitoreo continuo.
`nice`	Cambia la prioridad de un proceso al iniciarlo (prioridad baja o alta).	- `-n [valor]`: Establece el valor de prioridad (`-20` es la mayor prioridad, `19` la más baja). - `-g [grupo]`: Cambia la prioridad de todos los procesos en un grupo.	Controla la prioridad de un proceso cuando se inicia, mientras que el comando `renice` puede cambiar la prioridad de procesos ya en ejecución.

7. Otros comandos de Linux (Indique funcionalidad y parámetros):

(a) ¿A qué hace referencia el concepto de empaquetar archivos en GNU/Linux?

El concepto de empaquetar archivos se refiere al proceso de agrupar varios archivos y directorios en un solo archivo comprimido o no comprimido. Esto se hace para facilitar la distribución, almacenamiento o transferencia de un conjunto de archivos. El archivo resultante, conocido como “paquete”, puede ser descomprimido o desempaquetado para recuperar los archivos originales.

(b) Seleccione 4 archivos dentro de algún directorio al que tenga permiso y sume el tamaño de cada uno de estos archivos. Cree un archivo empaquetado conteniendo estos 4 archivos y compare los tamaños de los mismos. ¿Qué característica nota?

```
leo@leo:~/Desktop/Ejemplo$ ls -l
total 12
-rw-rw-r-- 1 leo leo 10240 Sep 13 17:46 archivo1.txt
-rw-rw-r-- 1 leo leo     0 Sep 13 17:43 archivo2.txt
-rw-rw-r-- 1 leo leo     0 Sep 13 17:43 archivo3.txt
-rw-rw-r-- 1 leo leo     0 Sep 13 17:43 archivo4.txt
leo@leo:~/Desktop/Ejemplo$ du --bytes
10240   .
leo@leo:~/Desktop/Ejemplo$ tar cvf archivoEmpaquetado archivo1.txt archivo2.txt archivo3.txt archivo4.txt
archivo1.txt
archivo2.txt
archivo3.txt
archivo4.txt
leo@leo:~/Desktop/Ejemplo$ ls -l
total 32
-rw-rw-r-- 1 leo leo 10240 Sep 13 17:46 archivo1.txt
-rw-rw-r-- 1 leo leo     0 Sep 13 17:43 archivo2.txt
-rw-rw-r-- 1 leo leo     0 Sep 13 17:43 archivo3.txt
-rw-rw-r-- 1 leo leo     0 Sep 13 17:43 archivo4.txt
-rw-rw-r-- 1 leo leo 20480 Sep 13 17:50 archivoEmpaquetado
leo@leo:~/Desktop/Ejemplo$ du --bytes
30720   .
leo@leo:~/Desktop/Ejemplo$
```

(c) ¿Qué acciones debe llevar a cabo para comprimir 4 archivos en uno solo? Indique la secuencia de comandos ejecutados.

(d) ¿Pueden comprimirse un conjunto de archivos utilizando un único comando?

Si se desea empaquetar y comprimir un conjunto de archivos se puede utilizar:

- tar -czvf archivo.tar archivo1 archivo2 archivo3 archivoN

- tar -cjvf archivo.tar archivo1 archivo2 archivo3 archivoN

- tar -cJvf archivo.tar archivo1 archivo2 archivo3 archivoN

Nota: j utilizará gzip (.tar.gz), z usará bzip2 (.tar.bz2) y J utiliza xz (.tar.xz).

Si ya se disponen de archivos empaquetados .tar y se desean comprimir:

- gzip archivo1.tar archivo2.tar archivo3.tar archivoN.tar

- bzip2 archivo1.tar archivo2.tar archivo3.tar archivoN.tar

(e) Investigue la funcionalidad de los siguientes comandos:

- tar
- grep
- gzip
- zgrep
- wc

Comando	Funcionalidad
`tar`	Utilizado para empaquetar y descomprimir archivos. Puede combinar múltiples archivos en un solo archivo tar (`.tar`) o extraer archivos de un archivo tar. También puede comprimir archivos usando opciones adicionales como `-z` para gzip o `-j` para bzip2.
`grep`	Busca texto en archivos usando expresiones regulares. Muestra las líneas que coinciden con el patrón especificado.
`gzip`	Comprime archivos utilizando el algoritmo de compresión gzip. Cambia la extensión del archivo a `.gz` . También puede descomprimir archivos gzip usando la opción `-d` .
`zgrep`	Similar a `grep` , pero diseñado para trabajar con archivos comprimidos en formato gzip. Permite buscar texto dentro de archivos `.gz` sin necesidad de descomprimirlos primero.
`wc`	Cuenta líneas, palabras y caracteres en un archivo o en la entrada estándar. Puede mostrar diferentes estadísticas usando opciones como `-l` para contar líneas, `-w` para contar palabras y `-c` para contar caracteres.

8. Indique qué acción realiza cada uno de los comandos indicados a continuación considerando su orden. Suponga que se ejecutan desde un usuario que no es root ni pertenece al grupo de root. (Asuma que se encuentra posicionado en el directorio de trabajo del usuario con el que se logueó). En caso de no poder ejecutarse el comando, indique la razón:

ls –l > prueba
ps > PRUEBA
chmod 710 prueba
chown root : root PRUEBA
chmod 777 PRUEBA
chmod 700 / e t c /passwd
passwd root
rm PRUEBA
man / e t c /shadow
f i n d / −name ∗. conf
usermod root −d /home/newroot −L
cd / root
rm ∗
cd / e t c
cp ∗ /home −R
shutdown

- ls –l > prueba: Genera un listado de todos los directorios y archivos del directorio de inicio del usuario de forma detallada y lo guarda en un archivo llamado prueba.

- ps > PRUEBA: Guarda un listado con la información de los procesos que se están ejecutando actualmente en un nuevo archivo llamado PRUEBA.
- chmod 710 prueba: Cambia los permisos de acceso al archivo prueba, le da permisos de lectura, escritura y ejecución al propietario, permiso de ejecución a los participantes de su grupo y cero permisos a los demás usuarios.
- chown root : root PRUEBA: Intenta cambiar el propietario y el grupo al que pertenece el archivo PRUEBA pero no puede porque no es un usuario root.

```
mati@74706125dca4:~$ chown root:root PRUEBA
chown: changing ownership of 'PRUEBA': Operation not permitted
```

- chmod 777 PRUEBA: Cambia los permisos de acceso al archivo PRUEBA, le da permisos de lectura, escritura y ejecución al propietario, a los participantes de su grupo y a los demás usuarios.
- chmod 700 /etc/passwd: El usuario intenta cambiar los permisos de lectura, escritura y ejecución al directorio /etc/passwd pero esta operación no es posible porque solo un usuario root puede administrar los permisos de los directorios almacenados en /etc.

```
mati@74706125dca4:~$ chmod 700 /etc/passwd
chmod: changing permissions of '/etc/passwd': Operation not permitted
```

- passwd root: El usuario intenta ver o cambiar la contraseña del usuario root pero esta acción no está permitida porque solo un usuario root puede cambiar la contraseña de otros usuarios.

```
mati@74706125dca4:~$ passwd root
passwd: You may not view or modify password information for root.
```

- rm PRUEBA: Se elimina el archivo PRUEBA.
- man /etc/shadow: Esta acción no se puede realizar ya que el comando man no puede brindar un acceso a las páginas del manual del sistema ya que estas brindan información detallada sobre los comandos, utilidades, funciones del sistema y archivos del sistema operativo, no sobre directorios, además el permiso es denegado ya que intenta acceder al directorio shadow el cual contiene las contraseñas encriptadas de los usuarios.

```
mati@74706125dca4:~$ man /etc/shadow
man: can't open /etc/shadow: Permission denied
```

- find / -name *.conf: Busca y lista a partir del directorio raíz todos los archivos y directorios que contengan la extensión .conf.
- usermod root –d /home/newroot –L: El Usuario intenta cambiar el directorio de inicio del Usuario root a el directorio /home/newroot y además intenta bloquear su contraseña, esta acción no está permitida ya que el usuario no posee los permisos necesarios para ejecutar el comando.

```
mati@74706125dca4:~$ usermod root -d /home/newroot -L
bash: usermod: command not found
```

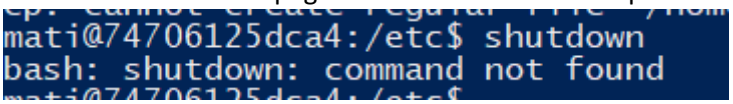
- cd /root: El usuario intenta acceder al directorio de inicio del usuario root, algo no posible ya que el usuario no posee los permisos necesarios.

```
mati@74706125dca4:~$ cd /root
bash: cd: /root: Permission denied
```

- rm *: Se eliminan todos los archivos en el directorio donde está posicionado el usuario actualmente.
- cd /etc: El usuario se sitúa en el directorio de configuraciones del sistema.
- cp * /home –R: El usuario intenta copiar todos los archivos y directorios del directorio actual /etc al directorio /home, incluyendo subdirectorios de manera recursiva, pero esto no es posible ya que solo los usuarios root tienen permisos para copiar el contenido del directorio /etc.

```
mati@74706125dca4:/etc$ cp * /home -R
cp: cannot create regular file '/home/adduser.conf': Permission denied
cp: cannot create directory '/home/alternatives': Permission denied
cp: cannot create directory '/home/apparmor.d': Permission denied
cp: cannot create directory '/home/apt': Permission denied
```

- shutdown: El usuario apaga el sistema. Se necesitan permisos sudo para hacerlo.



9. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones:

(a) Terminar el proceso con PID 23.

Se podría utilizar los comandos “kill 23” ó “kill -9 23”.

(b) Terminar el proceso llamado init o systemd. ¿Qué resultados obtuvo?

Se podría intentar usar el comando “kill -9 1” pero no se obtendría ningún resultado ya que el proceso init es de suma importancia para el arranque del sistema y este no puede ser terminado.

(c) Buscar todos los archivos de usuarios en los que su nombre contiene la cadena “.conf”

Se podría utilizar el comando “find /home -name *.conf*”.

(d) Guardar una lista de procesos en ejecución el archivo **/home/<su nombre de usuario>/procesos**

Se podría utilizar el comando “ps > /home/mati/procesos”.

(e) Cambiar los permisos del archivo **/home/<su nombre de usuario>/xxxx** a:

- Usuario: Lectura, escritura, ejecución
- Grupo: Lectura, ejecución
- Otros: ejecución

Se podría utilizar el comando “chmod 751 /home/suUsuario/archivoX”.

(f) Cambiar los permisos del archivo **/home/<su nombre de usuario>/yyyy** a:

- Usuario: Lectura, escritura.
- Grupo: Lectura, ejecución
- Otros: Ninguno

Se podría utilizar el comando “chmod 650 /home/suUsuario/archivoY”.

(g) Borrar todos los archivos del directorio **/tmp**

m /tmp/* # Solo borra archivos.

rm -r /tmp/* # Borra archivos, links y directorios.

(h) Cambiar el propietario del archivo **/opt/isodata** al usuario **iso2010**

Se podría utilizar el comando “sudo chown iso2010 /opt/isodata”.

(i) Guardar en el archivo **/home/<su nombre de usuario>/donde** el directorio donde me encuentro en este momento, en caso de que el archivo exista no se debe eliminar su contenido anterior.

Se podría utilizar el comando “pwd >> /home/usuario/donde”.

10. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones:

(a) Ingrese al sistema como usuario “root”

su - root

(b) Cree un usuario. Elija como nombre, por convención, la primera letra de su nombre seguida de su apellido. Asígnele una contraseña de acceso.

useradd -m -s /bin/bash MGuaymas # Sin contraseña porque debe ingresarse encriptada.

passwd Blacky # Se le asigna una contraseña.

(c) ¿Qué archivos fueron modificados luego de crear el usuario y qué directorios se crearon?

Se modificaron los archivos /etc/passwd, /etc/shadow, /etc/group, /etc/gshadow y se creó el directorio de inicio del nuevo usuario en /home/MGuaymas.

(d) Crear un directorio en /tmp llamado cursada2017

mkdir /tmp/cursada2017

(e) Copiar todos los archivos de /var/log al directorio antes creado.

sudo cp /var/log/* > /tmp/cursada2017.

(f) Para el directorio antes creado (y los archivos y subdirectorios contenidos en él) cambiar el propietario y grupo al usuario creado y grupo users.

chown -R MGuaymas:users cursada2017/

(g) Agregue permiso total al dueño, de escritura al grupo y escritura y ejecución a todos los demás usuarios para todos los archivos dentro de un directorio en forma recursiva.

chmod -R 723 cursada2017/

(h) Acceda a otra terminal virtual para loguearse con el usuario antes creado.

su - MGuaymas

(i) Una vez logueado con el usuario antes creado, averigüe cuál es el nombre de su terminal.

cat /etc/passwd | grep MGuaymas | cut -f7 -d:

cat /etc/passwd - Concateno el contenido del archivo.

grep MGuaymas - Filtro por el usuario que deseo.

cut -f7 -d: - Tomo la columna 7 y delimito el contenido por ":"

(j) Verifique la cantidad de procesos activos que hay en el sistema.

ps -e | wc -l

(k) Verifiqué la cantidad de usuarios conectados al sistema.

who

(l) Vuelva a la terminal del usuario root, y envíele un mensaje al usuario anteriormente creado, avisándole que el sistema va a ser apagado.

su.

wall "El sistema está por apagarse". # Envía un mensaje a todos los usuarios que están actualmente conectados al sistema, se necesita permisos sudo para usarlo.

(m) Apague el sistema.

shutdown -P now

11. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones:

(a) Cree un directorio cuyo nombre sea su número de legajo e ingrese a él.

mkdir 23061_0

(b) Cree un archivo utilizando el editor de textos vi, e introduzca su información personal: Nombre, Apellido, Número de alumno y dirección de correo electrónico. El archivo debe llamarse "LEAME".

vi LEAME

Dentro del editor cambiar a modo de inserción usando la tecla “i”, escribir toda la información en el archivo y luego volver al modo comando usando la tecla “Esc”.

Una vez en el modo comando escribir el comando “:wq”.

(c) Cambie los permisos del archivo LEAME, de manera que se puedan ver reflejados los siguientes permisos:

- Dueño: ningún permiso
- Grupo: permiso de ejecución
- Otros: todos los permisos

chmod 017 LEAME

(d) Vaya al directorio /etc y verifique su contenido. Cree un archivo dentro de su directorio personal cuyo nombre sea leame donde el contenido del mismo sea el listado de todos los archivos y directorios contenidos en /etc. ¿Cuál es la razón por la cual puede crear este archivo si ya existe un archivo llamado “LEAME” en este directorio?

“cd /etc”.

“ls > /home/leame”.

Esta acción se puede realizar debido a que GNU/Linux es case sensitive.

(e) ¿Qué comando utilizaría y de qué manera si tuviera que localizar un archivo dentro del filesystem? ¿Y si tuviera que localizar varios archivos con características similares? Explique el concepto teórico y ejemplifique.

Para buscar un solo archivo con find: “find / -name nombre_del_archivo”.

Para buscar archivos con características similares: “find / - name “*cadena_a_contener”.

(f) Utilizando los conceptos aprendidos en el punto e), busque todos los archivos cuya extensión sea .so y almacene el resultado de esta búsqueda en un archivo dentro del directorio creado en a). El archivo deberá llamarse .ejercicio_f".

find / -type f -name *.so > /home/suUsuario/legajo/.ejercicio_f # Solo archivos.

find / -name *.so > /home/suUsuario/legajo/.ejercicio_f # Archivos, directorios y links.

12. Indique qué acción realiza cada uno de los comandos indicados a continuación considerando su orden. Suponga que se ejecutan desde un usuario que no es root ni pertenece al grupo de root. (Asuma que se encuentra posicionado en el directorio de trabajo del usuario con el que se logueó). En caso de no poder ejecutarse el comando indique la razón:

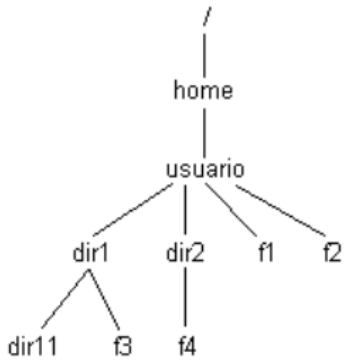
```
mkdir iso
cd ./iso; ps > f0
ls > f1
cd /
echo $HOME
ls -l &> $HOME/iso/ls
cd $HOME; mkdir f2
ls -ld f2
chmod 341 f2
touch dir
cd f2
cd ~/iso
pwd >f3
ps | grep 'ps' | wc -l >> ../f2/f3
chmod 700 ../f2; cd ..
find . -name etc/passwd
find / -name etc/passwd
mkdir ejercicio5
------------------------------------------------
------------------------------------------------
```

(a) Inicie 2 sesiones utilizando su nombre de usuario y contraseña. En una sesión vaya siguiendo paso a paso las órdenes que se encuentran escritas en el cuadro superior. En la otra sesión, cree utilizando algún editor de textos un archivo que se llame .ejercicio10_explicacion" dentro del directorio creado en el ejercicio 9.a) y, para cada una de las órdenes que ejecute en la otra sesión, realice una breve explicación de los resultados obtenidos.

(b) Complete en el cuadro superior los comandos 19 y 20, de manera tal que realicen la siguiente acción:

- 19: Copiar el directorio iso y todo su contenido al directorio creado en el inciso 9.a).
- 20: Copiar el resto de los archivos y directorios que se crearon en este ejercicio al directorio creado en el ejercicio 9.a).

(c) Ejecute las órdenes 19 y 20 y coméntelas en el archivo creado en el inciso a).



- "mkdir iso": Crea el directorio "iso" en su directorio inicial.
- "cd ./iso ; ps > f0": Se sitúa en el directorio "iso" y crea el archivo "f0" con un listado de todos los procesos en ejecución en el sistema actualmente.
- "ls > f1": Crea un archivo llamado "f1" en el directorio "iso" que contiene la lista de archivos y/o subdirectorios del directorio "iso".
- "cd /": Se sitúa en el directorio raíz.
- "echo $HOME": Muestra en la salida estándar la dirección del directorio inicial del usuario.

- “ls –l &> $HOME/iso/ls”: Se crea un nuevo archivo llamado “ls” en el directorio “iso” ubicado en el directorio inicial del usuario que va a tener un listado detallado de los archivos y/o directorios del directorio raíz y además también la salida de error del comando realizado “ls –l”, esto se logra sumando a la redirección el símbolo “&”.
- “cd $HOME ; mkdir f2”: Se sitúa en su directorio inicial y crea el directorio “f2”.
- “ls –ld f2”: Muestra información detallada del directorio “f2”. Se especifica que se trate al mismo como un directorio con el parámetro “d”.
- “chmod 341 f2”: Cambia los permisos de lectura, escritura y ejecución del directorio “f2”. Escritura y Ejecución para el dueño, Lectura para el grupo y Ejecución para los demás usuarios.
- “touch dir”: Crea un archivo llamado “dir” en el directorio inicial del usuario.
- “cd f2”: Se sitúa en el directorio “f2”.
- “cd ~/iso”: Se sitúa en el directorio “iso”.
- “pwd > f3”: Crea un archivo nuevo en el directorio “iso” que contiene la información de la ruta actual del usuario en ese momento “~/iso”.
- “ps | grep “ ps ” | wc –l >> ../f2/f3”: Se listan todos los procesos en ejecución y el “grep” recibe esa lista como entrada estándar, allí filtra los resultados por aquellos procesos que posean la cadena “ps”, ese nuevo listado filtrado es pasado como entrada estándar al comando “wc” el cuál cuenta la cantidad de líneas de ese listado, es decir, la cantidad de procesos que se encontraron con la cadena “ps” para luego redirigir esa salida al archivo “f3” añadiendo la cantidad al final del mismo.
- “chmod 700 ../f2 ; cd ..”: Se modifican los permisos de lectura, escritura y ejecución del directorio “f2”. Lectura, Escritura y Ejecución para el dueño y cero permisos para el grupo y otros usuarios. Luego al ejecutar “cd ..” escala un nivel hacia arriba en la estructura de directorios, ahora situándose nuevamente en su directorio inicial.
- “find . –name etc/passwd”: Se muestra un mensaje de warning por el mal uso del comando “find”.
- “find / -name etc/passwd”: Muestra en pantalla el listado de los archivos y/o directorios de “/etc/passwd” ordenados por nombre.
- “mkdir ejercicio5”: Crea el directorio “ejercicio5”.

13. Cree una estructura desde el directorio /home que incluya varios directorios, subdirectorios y archivos, según el esquema siguiente. Asuma que “usuario” indica cuál es su nombre de usuario. Además, deberá tener en cuenta que dirX hace referencia a directorios y fX hace referencia a archivos:

(a) Utilizando la estructura de directorios anteriormente creada, indique que comandos son necesarios para realizar las siguientes acciones:

- Mueva el archivo "f3.al directorio de trabajo /home/usuario.
- Copie el archivo "f4.en el directorio "dir11".
- Haga los mismo que en el inciso anterior pero el archivo de destino, se debe llamar "f7".
- Cree el directorio copia dentro del directorio usuario y copie en él, el contenido de "dir1".
- Renombre el archivo "f1"por el nombre archivo y vea los permisos del mismo.
- Cambie los permisos del archivo llamado archivo de manera de reflejar lo siguiente:
    - Usuario: Permisos de lectura y escritura
    - Grupo: Permisos de ejecución
    - Otros: Todos los permisos
- Renombre los archivos "f3 2 "f4"de manera que se llamen "f3.exe 2 "f4.exe respectivamente.
- Utilizando un único comando cambie los permisos de los dos archivos renombrados en el inciso anterior, de manera de reflejar lo siguiente:
    - Usuario: Ningún permiso
    - Grupo: Permisos de escritura

- Otros: Permisos de escritura y ejecución

- “mv f3 $HOME”.
- “cp f4 $HOME/dir1/dir11”.
- “cp f4 $HOME/dir1/dir11/f7”.
- “mkdir copia ; cp –a dir1 copia”.
- “mv f1 archivo ; ls –ld archivo”.
- “chmod 617 archivo”.
- “mv f3 f3.exe ; mv f4 f4.exe”.
- “chmod 023 f3.exe f4.exe”.

14. Indique qué comando/s es necesario para realizar cada una de las acciones de la siguiente secuencia de pasos (considerando su orden de aparición):

(a) Cree un directorio llamado logs en el directorio /tmp.

sudo mkdir /tmp/logs

(b) Copie todo el contenido del directorio /var/log en el directorio creado en el punto anterior.

sudo cp -a /var/log/* /tmp/logs # -r por si hay directorios.

(c) Empaquete el directorio creado en 1, el archivo resultante se debe llamar "misLogs.tar".

tar –cvf misLogs.tar logs

(d) Empaquete y comprima el directorio creado en 1, el archivo resultante se debe llamar "misLogs.tar.gz".

tar –cvfz misLogs.tar.gz logs

(e) Copie los archivos creados en 3 y 4 al directorio de trabajo de su usuario.

cp misLogs.tar misLogs.tar.gz /home/suUsuario

(f) Elimine el directorio creado en 1, logs.

rm -r logs

(g) Desempaquete los archivos creados en 3 y 4 en 2 directorios diferentes.

mkdir /home/suUsuario/1 /home/suUsuario/2

tar -xvf misLogs.tar -C /home/suUsuario/1 # -C para cambiar la ruta.

tar -xzvf misLogs.tar.gz -C /home/suUsuario/2