

1) Editor de textos:

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

VI/VIM, MCEdit y Nano.

- 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.

Cat, more y less solo muestran el contenido de un archivo en diferentes formatos, un editor de texto permite editar el contenido de los mismos.

i, a, o, O: Modo de inserción.

i: Insertar texto antes del cursor.

I: Insertar texto al inicio de la línea actual.

a: Insertar texto después del cursor.

A: Insertar texto al final de la línea actual.

o: Insertar texto en una nueva línea debajo del cursor.

O: Insertar texto en una nueva línea arriba del cursor.

Esc: Modo comandos.

Esc y : : Modo línea de comandos.

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

Comandos más comunes:

q!: Salir del editor sin guardar los cambios.

wq!: Guardar los cambios y salir del editor.

w: Guardar los cambios.

num: Mostrar el número de línea actual.

yy: Copiar una línea de texto.

p: Pegar una línea de texto previamente copiada con yy.

b: Ir al inicio de la palabra.

e: Ir al final de la palabra.

x: Borrar un solo carácter.

dd: Eliminar una línea entera.

Xdd: Borrar X número de líneas.

Xyy: Copiar X número de líneas.

G: Ir a la última línea del archivo.

XG: Ir a la línea X del archivo.

gg: Ir a la primera línea del archivo.


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.

Proceso de arranque System V

(<https://github.com/systeminit/si>)

- 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




Facultad de Informática
UNIVERSIDAD NACIONAL DE LA PLATA

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

Lo ejecuta el Kernel.

init

- 1 Su función es cargar todos los subprocesos necesarios para el correcto funcionamiento del SO
- 2 El proceso *init* posee el PID 1 y se encuentra en */sbin/init*
- 3 En SysV se lo configura a través del archivo */etc/inittab*
- 4 No tiene padre y es el padre de todos los procesos (*pstree*)
- 5 Es el encargado de montar los filesystems y de hacer disponible los demás dispositivos



Facultad de Informática
UNIVERSIDAD NACIONAL DE LA PLATA

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

Es el modo en que arranca Linux (3 en Redhat, 2 en Debian).

El proceso de arranque se divide en niveles, cada uno es responsable de iniciar o parar una serie de servicios.

Se encuentran definidos en /etc/inittab -> id:nivelesEjecucion: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.

```
$ cat /etc/inittab
id:2:initdefault:
si::sysinit:/etc/init.d/rcS
ca::ctrlaltdel:/sbin/shutdown -t3 -r
```

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?

- Existen 7, y permiten iniciar un conjunto de procesos al arranque o apagado del sistema
- 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

- Los scripts que se ejecutan están en /etc/init.d
- En /etc/rcX.d (donde X = 0..6) hay links a los archivos del /etc/init.d

- Formato de los links:

[S|K]<orden><nombreScript>

```
$ ls -l /etc/rcS.d/
S55urandom
S70x11-common
```

- **S**: lanza el script con el argument start
- **K**: lanza el script con el argument stop

Robado de <https://github.com/agusrnfr/>

Cuando un sistema GNU/Linux arranca, primero se carga el kernel del sistema, después se inicia el primer proceso, denominado init, que es el responsable de ejecutar y activar el resto del sistema, mediante la gestión de los niveles de ejecución (o runlevels).

En el caso del modelo runlevel de SystemV, cuando el proceso init arranca, utiliza un fichero de configuración llamado /etc/inittab para decidir el modo de ejecución en el que va a entrar. En este fichero se define el runlevel por defecto (initdefault) en arranque (por instalación en Fedora el 5, en Debian el 2), y una serie de servicios de terminal por activar para atender la entrada del usuario. Después, el sistema, según el runlevel escogido, consulta los ficheros contenidos en /etc/rcn.d, donde n es el número asociado al runlevel (nivel escogido), en el que se encuentra una lista de servicios por activar o parar en caso de que arranquemos en el runlevel, o lo abandonemos. Dentro del directorio encontraremos una serie de scripts o enlaces a los scripts que controlan el servicio. Cada script posee un nombre relacionado con el servicio, una S o K inicial que indica si es el script para iniciar (S) o matar (K) el servicio, y un número que refleja el orden en que se ejecutarán los servicios.

No todas las distribuciones respetan los estándares.

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

Robado de <https://github.com/agusrnfr/>

Es el archivo de configuración de `init`. Cuando el sistema se arranca, se verifica si existe un runlevel predeterminado en el archivo `/etc/inittab`, si no, se debe introducir por medio de la consola del sistema. Después se procede a ejecutar todos los scripts relativos al runlevel especificado.

- 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?

El comando necesario es `init`.

```
leo@leo:~$ init --help
init [OPTIONS...] COMMAND

Send control commands to the init daemon.

Commands:
  0          Power-off the machine
  6          Reboot the machine
  2, 3, 4, 5 Start runlevelX.target unit
  1, s, S    Enter rescue mode
  q, Q       Reload init daemon configuration
  u, U       Reexecute init daemon

Options:
  --help      Show this help
  --no-wall   Don't send wall message before halt/power-off/reboot

See the telinit(8) man page for details.
```

Los cambios realizados con `init` son temporales y afectan a la sesión actual del sistema. Si se cambia el nivel de ejecución, el sistema solo aplicará para esa sesión.

Si se reinicia el sistema, el nivel de ejecución predeterminado se restaurará al configurado en los archivos de configuración (`/etc/inittab` y `systemd`).

Los cambios permanentes se deben realizar directamente sobre los archivos de configuración del sistema.

- 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.

Los scripts RC (Run Commands) tienen la finalidad de iniciar o detener servicios cuando el sistema GNU/Linux arranca o se detiene. Su principal propósito es asegurarse de que los servicios (como redes, daemons (procesos que se ejecutan en segundo plano), montajes de FileSystem, etc) se inicien y se apaguen correctamente a medida que el sistema cambia de estado operativo o nivel de ejecución (RunLevel).

Los scripts RC se almacenan en directorios específicos que dependen del sistema de inicialización que utiliza la distribución Linux:

En sistemas basados en sysvinit:

- /etc/rc.d o /etc/init.d
- Para cada Runlevel existen subdirectorios llamados /etc/rcX.d, donde X es el número de RunLevel. Cada uno de estos directorios contiene enlaces simbólicos a los scripts en /etc/init.d.

Los scripts se ejecutan en un orden específico:

- El orden de los scripts en cada nivel de ejecución se determina por los números después de las letras "S" (start) y "K" (kill), por ejemplo, el script "S10cron" se ejecutará antes que "S20network" debido a que $10 < 20$.
- Este orden asegura que algunos servicios críticos se inicien o se detengan en un orden específico para evitar conflictos.

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

- a) ¿Qué es systemd?

SystemD es un conjunto de daemons de administración de sistema, bibliotecas y herramientas diseñados como una plataforma de administración y configuración central para interactuar con el núcleo del Sistema Operativo GNU/Linux.

Descrito como "bloque de construcción básico" para un sistema operativo, SystemD se puede utilizar tanto como un sistema de inicio de Linux (el proceso init llamado por el núcleo o kernel de Linux para inicializar el espacio de usuario durante el proceso de arranque de Linux y gestionar posteriormente todos los demás procesos). El nombre SystemD se adhiere a la convención de Unix de distinguir los daemons fácilmente por tener la letra "D" como última letra del nombre del archivo.

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

Se denomina Unit a las unidades de trabajo de tipo:

- **Service:** Controla un servicio particular (.service).
- **Socket:** Encapsula IPC, un socket del sistema o FileSystem FIFO (.socket) -> socket-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 reestablecido más tarde (.snapshot).
- Etc.

Las Units pueden tener dos estados: Active o Inactive.

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

```
leo@leo:~$ systemctl --help
systemctl [OPTIONS...] COMMAND ...

Query or send control commands to the system manager.

Unit Commands:
list-units [PATTERN...]          List units currently in memory
list-automounts [PATTERN...]     List automount units currently in memory,
                                ordered by path
list-paths [PATTERN...]          List path units currently in memory,
                                ordered by path
list-sockets [PATTERN...]        List socket units currently in memory,
                                ordered by address
list-timers [PATTERN...]         List timer units currently in memory,
                                ordered by next elapse
is-active PATTERN...             Check whether units are active
is-failed [PATTERN...]           Check whether units are failed or
                                system is in degraded state
status [PATTERN...|PID...]       Show runtime status of one or more units
show [PATTERN...|JOB...]         Show properties of one or more
                                units/jobs or the manager
cat PATTERN...                  Show files and drop-ins of specified units
help PATTERN...|PID...          Show manual for one or more units
list-dependencies [UNIT...]      Recursively show units which are required
                                or wanted by the units or by which those
                                units are required or wanted
start UNIT...                   Start (activate) one or more units
stop UNIT...                    Stop (deactivate) one or more units
lines 1-27...skipping...
```

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

En SystemD, un Target es una unidad que agrupa otras unidades para alcanzar un estado específico del sistema. Es un concepto que sirve para organizar y controlar las dependencias entre los diferentes servicios, sockets, dispositivos, y otros targets que deben estar activos o inactivos en un momento determinado.

Los Targets son una forma de abstraer y agrupar acciones comunes de arranque y apagado del sistema, permitiendo que SystemD gestione el inicio, detención y reinicio de servicios en el orden correcto. Un Target define un “estado del sistema”, que es un conjunto de servicios y dependencias que deben estar funcionando para cumplir un propósito particular.

Ejemplos de Targets:

- multi-user.target: Indica que el sistema está en un estado donde varios usuarios pueden acceder de manera no gráfica.
- graphical.target: Similar al ejemplo anterior, pero además habilita la interfaz gráfica.
- rescue.target: Proporciona un entorno de rescate con los servicios mínimos necesarios para reparaciones o diagnósticos.
- shutdown.target: Se utiliza cuando el sistema se apaga, asegurándose de que todos los servicios se detengan correctamente antes de apagar la máquina.

Los Targets pueden depender unos de otros o de otros servicios. Por ejemplo, graphical.target depende de multi-user.target porque se necesita una base de servicios multiusuario antes de que se inicie la interfaz gráfica.

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

El comando **pstree** muestra un árbol con los procesos que se están ejecutando.


```

leo@leo:~$ pstree
systemd--ModemManager--3*[{ModemManager}]
--NetworkManager--3*[{NetworkManager}]
--accounts-daemon--3*[{accounts-daemon}]
--atd
--avahi-daemon--avahi-daemon
--colord--3*[{colord}]
--containerd--14*[{containerd}]
--cron
--cups-browsed--3*[{cups-browsed}]
--cupsd
--dbus-daemon
--dockerd--16*[{dockerd}]
--fwupd--5*[{fwupd}]
--gdm3--gdm-session-wor--gdm-x-session--Xorg--11*[{Xorg}]
--gnome-session-b--3*[{gnome-session-b}]
--3*[{gdm-x-session}]
--3*[{gdm-session-wor}]
--3*[{gdm3}]
--gnome-remote-de--3*[{gnome-remote-de}]
--2*[{kerneloops}]
--mysqld--36*[{mysqld}]
--polkitd--3*[{polkitd}]
--postgres--5*[{postgres}]
--power-profiles--3*[{power-profiles-}]
--rsyslogd--3*[{rsyslogd}]
--rtkit-daemon--2*[{rtkit-daemon}]
--snapd--18*[{snapd}]
--switcheroo-cont--3*[{switcheroo-cont}]
systemd--(sd-pam)
--at-spi-bus-laun--dbus-daemon
--4*[{at-spi-bus-laun}]
--at-spi2-registr--3*[{at-spi2-registr}]
--2*[{chrome_crashpad}--2*[{chrome_crashpad}]]
--chrome_crashpad--(chrome_crashpad)
--com.docker.back--com.docker.back--Docker Desktop--Docker Desktop--Docker Desktop--28*[{Docker Desktop}]
--Docker Desktop--Docker Desktop
--Docker Desktop--9*[{Docker Desktop}]
--34*[{Docker Desktop}]
--com.docker.buil--15*[{com.docker.buil}]
--com.docker.dev--8*[{com.docker.dev-}]
--24*[{com.docker.back}]
--14*[{com.docker.back}]
--dbus-daemon
--dconf-service--3*[{dconf-service}]
--2*[{evince}--7*[{evince}]]
--evinced--3*[{evinced}]
--evolution-addre--6*[{evolution-addre}]
--evolution-calen--9*[{evolution-calen}]
--evolution-sourc--4*[{evolution-sourc}]
--gcr-ssh-agent--2*[{gcr-ssh-agent}]
--2*[{gjs}--11*[{gjs}]]
--gnome-keyring-d--4*[{gnome-keyring-d}]
--gnome-session-b--evolution-alarm--7*[{evolution-alarm}]
--gsd-disk-utilit--3*[{gsd-disk-utilit}]
--update-notifier--5*[{update-notifier}]
--4*[{gnome-session-b}]
--gnome-session-c--(gnome-session-c)
--gnome-shell--Discord--Discord--Discord--30*[{Discord}]
--Discord--Discord
--Discord--14*[{Discord}]
--Discord--5*[{Discord}]
--Discord--47*[{Discord}]
--45*[{Discord}]
--brave-browser-s--brave--brave--brave--30*[{brave}]

```

4) Usuarios:

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

/etc/passwd: Contiene una lista de los usuarios del sistema. Cada línea representa un usuario y contiene campos como el nombre de usuario, ID del usuario (UID), el ID del grupo (GID), la ruta de inicio (home directory), y la shell predeterminada.

/etc/shadow: Almacena las contraseñas cifradas y la información relacionada con la política de contraseñas, como la fecha de expiración y los períodos de validez. Solo los usuarios privilegiados (como root) pueden acceder a este archivo por razones de seguridad.

/etc/group: Tiene la información sobre los grupos del sistema y los usuarios asociados a cada grupo. Cada línea representa un grupo, con detalles como el nombre del grupo, el GID y los usuarios que son miembros del grupo.

/etc/gshadow: Similar a /etc/shadow, este archivo contiene las contraseñas cifradas de los grupos (si los grupos tienen contraseña) y otra información sobre los grupos.

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

UID hace referencia a la ID del usuario (**U**ser **I**Dentifier).

GID hace referencia a la ID del grupo (**G**roup **I**Dentifier).

La UID es la forma que tiene el sistema operativo de identificar al usuario, por lo tanto si 2 o más usuarios tienen la misma UID, el sistema no podría diferenciar entre ellos y ambos tendrían acceso a los mismos archivos y permisos, lo que sería un problema de seguridad.

En la práctica sí es posible que esto pase, se pueden mantener varios usuarios con la misma UID. A su vez, esto va en contra de las prácticas recomendadas y puede causar problemas.

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

El usuario root es el administrador del sistema o superusuario que tiene control total sobre todo el sistema. Puede ejecutar cualquier comando y tiene acceso a todos los archivos y recursos, sin restricciones de permisos.

En la práctica, se podrían tener múltiples usuarios root, cualquier usuario al que se le asigne la UID 0 tendrá permisos de administrador.

```
leo@leo:~$ cat /etc/passwd | grep "root"
root:x:0:0:root:/root:/bin/bash
```

- 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@8319eb6f9136:/# groupadd catedra
root@8319eb6f9136:/# sudo useradd -d "/home/iso_2017" -m iso2017
root@8319eb6f9136:/# sudo usermod -a -G catedra iso2017
root@8319eb6f9136:/# cat /etc/passwd | grep "iso2017" && cat /etc/group | grep "catedra"
iso2017:x:1000:1001::/home/iso_2017:/bin/sh
catedra:x:1000:iso2017
root@8319eb6f9136:/# cd /home/iso_2017
root@8319eb6f9136:/home/iso_2017# touch ejemplo.txt
root@8319eb6f9136:/home/iso_2017# userdel iso2017
root@8319eb6f9136:/home/iso_2017# cat /etc/passwd | grep "iso2017"
root@8319eb6f9136:/home/iso_2017#
```

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

- useradd o 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 subuidid 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_DIR    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

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

Options:
-h, --help          display this help message and exit
-R, --root CHROOT_DIR
                    directory to chroot into
-P, --prefix PREFIX_DIR
                    prefix directory where are located the /etc/* files
-f, --force         delete group even if it is the primary group of a user
    --extrausers     Use the extra users database
```

- passwd

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

Options:
  -a, --all                report password status on all accounts
  -d, --delete             delete the password for the named account
  -e, --expire             force expire the password for the named account
  -h, --help              display this help message and exit
  -k, --keep-tokens        change password only if expired
  -i, --inactive INACTIVE set password inactive after expiration
                           to INACTIVE
  -l, --lock               lock the password of the named account
  -n, --mindays MIN_DAYS  set minimum number of days before password
                           change to MIN_DAYS
  -q, --quiet              quiet mode
  -r, --repository REPOSITORY change password in REPOSITORY repository
  -R, --root CHROOT_DIR   directory to chroot into
  -S, --status             report password status on the named account
  -u, --unlock             unlock the password of the named account
  -w, --warndays WARN_DAYS set expiration warning days to WARN_DAYS
  -x, --maxdays MAX_DAYS set maximum number of days before password
                           change to MAX_DAYS
```

5) FileSystems:

- ¿Cómo son definidos los permisos sobre los archivos en un sistema GNU/Linux?
- Investigue sobre la funcionalidad y parámetros de los siguientes comandos relacionados con los permisos en GNU/Linux:
 - chmod
 - chown
 - chgrp
- Al utilizar el comando chmod generalmente se utiliza una notación octal asociada para definir permisos ¿Qué significa esto? ¿A qué hace referencia cada valor?
- ¿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.
- Explique los conceptos de “full path name” y “relative path name”. De ejemplos claros de cada uno de ellos.
- ¿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.
- 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