



Practica 1 (GNU/Linux y Conceptos) - Resolucion

1. Características de GNU/Linux:

a. Mencione y explique las características más relevantes de GNU/Linux.

- Todo es un archivo en linux.
- Puede ser usado, copiado, estudiado y redistribuido ya que mantiene una característica de software libre
- Correccion rapida a los errores

b. Mencione otros sistemas operativos y compárelos con GNU/Linux en cuanto a los puntos mencionados en el inciso a.

- Comparando Windows con Linux:
 - Nula distribucion libre
 - Siempre viene con restricciones y no permite su modificacion directa
 - Correccion de fallas a cargo del propietario y normalmente hay que esperar a actualizaciones futuras
 - Siempre viene con un costo asociado

c. ¿Qué es GNU?

- GNU (GNU NotUnix) es un sistema operativo similar a Unix con la diferencia de que este esta compuesto en su totalidad por software libre, ademas de contener paquetes y demas.

d. Indique una breve historia sobre la evolución del proyecto GNU.

e. Explique qué es la multitarea, e indique si GNU/Linux hace uso de ella.

- Si hace uso de multitarea, ademas de tener multiprocesador (ejemplo pueden ser 4 cada uno con su PC) y multiusuarios. La multitarea tiene como objetvo permitirle al usuario estar realizando varias tareas al mismo tiempo sin que ninguna de ellas sea interrumpida por la otra

f. ¿Qué es POSIX?

-Es una familia de estándares que tiene como objetivo la interoperabilidad en los sistemas operativos. Si se sigue un estándar POSIX es muy probable que ese sistema operativo se pueda utilizar en SO de UNIX, de lo contrario puede tener alguna falla y no funcionar en todos los sistemas

2. Distribuciones de GNU/Linux:

a. ¿Qué es una distribución de GNU/Linux? Nombre al menos 4 distribuciones de GNU/Linux y cite diferencias básicas entre ellas.

-Son un conjunto de aplicaciones que permiten brindar mejoras o distintas herramientas para instalar fácilmente un sistema operativo Linux/GNU (Diferentes interfaces, uso de HW, etc). Se diferencian entre sí justamente por esto mismo, depende de las necesidades del usuario y de lo que lo que esta distribución ofrezca para que el usuario tome la decisión por gusto personal de instalar una u otra.

-Distribuciones:

- Ubuntu, se centra en ser lo más amigable o friendly para la hora de la instalación

- Debian, se centra en servidores Linux

- SteamOS, se centra en los videojuegos

- Linux Mint, uso de HW más potente para competir con Windows

b. ¿En qué se diferencia una distribución de otra?

-Lo único que tienen en común es el Kernel, y lo que cambia es todo el resto de herramientas/componentes/funcionalidades ya que la shell, la GUI, herramientas de cada distribución son diferentes y se personalizan desde 0.

c. ¿Qué es Debian? Acceda al sitio e indique cuáles son los objetivos del proyecto y una breve cronología del mismo.

-Debian es un grupo de personas con una causa común que se juntaron para crear un SO libre

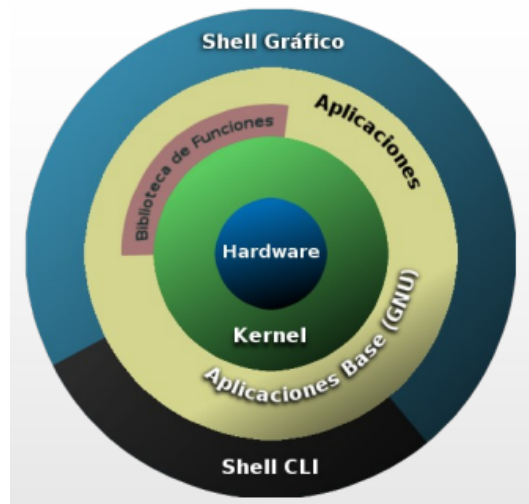
3. Estructura de GNU/Linux:

a. Nombre cuáles son los 3 componentes fundamentales de GNU/Linux.

-Kernel (núcleo), Shell (intérprete de comandos, intermediario entre usuario y Kernel) y FileSystem (sistema de archivos)

b. Mencione y explique la estructura básica del Sistema Operativo GNU/Linux.

-La estructura básica de un SO es el Kernel, el cual se encarga de ejecutar los programas y de toda la gestión de dispositivos HW ya que nosotros los usuarios no tenemos acceso directo a ello.



4. Kernel

a. ¿Cuáles son sus funciones principales?

-Su función principal es la administración de memoria, CPU y de dispositivos E/S

b. ¿Cuál es la versión actual? ¿Cómo se definía el esquema de versionado del Kernel en versiones anteriores a la 2.4? ¿Qué cambió en el versionado que se impuso a partir de la versión 2.6?

-5.6. Anteriormente los números pares indicaban que la versión lanzada era estable (1.2, 2.4, ..) y los números impares indicaban que eran versiones en desarrollo y podían contener fallos (2.5, 2.7)

c. ¿Es posible tener más de un Kernel de GNU/Linux instalado en la misma máquina?

-Si es posible tener más de uno instalado pero al momento de levantar el SO solo funciona uno

d. ¿Dónde se encuentra ubicado dentro del File System?

-Se encuentra en /boot en el primer sector del disco y nos indica informacion general de donde se almacena el Kernel y como arranca

5. Intérprete de comandos (Shell):

a. ¿Qué es?

-Es un programa que recibe lo que se escriba en la terminal y lo transforma en instrucciones para el Kernel. Es un intermediario entre el usuario y el Kernel

b. ¿Cuáles son sus funciones?

-Interfaz mas "amigable" para comunicar al usuario con el Kernel. Cual tipo de mensaje los espera mediante una ventana que espera comandos textuales y los interpreta, decodifica y entrega al SO para que se ejecuten

c. Mencione al menos 3 intérpretes de comandos que posee GNU/Linux y compárelos entre ellos.

d. ¿Dónde se ubican (path) los comandos propios y externos al Shell?

-Los comandos externos (ls, cat, mkdir) se buscan en los directions indicados en la variable PATH

-Los comandos internos (cd, bg, alias, eval) se ejecutan en el mismo proceso del shell

e. ¿Por qué considera que el Shell no es parte del Kernel de GNU/Linux?

-Ya que actua de intermediario o de mensajero con el Kernel, si estuviera en el mismo, el shell podria interpretar ordenes de usuario y ejectarlas al instante sobre el HW y no queremos eso, tendria acceso a instrucciones propias del SO de manera inmedianta.

f. ¿Es posible definir un intérprete de comandos distinto para cada usuario? ¿Desde dónde se define? ¿Cualquier usuario puede realizar dicha tarea?

-Si porque cada usuario tiene asignado un shell que se inicia con el proceso login, el cual hay que realizar y se logea a una terminal

6. El sistema de Archivos (File System) en Linux:

a. ¿Qué es?

-Es la forma en la que los archivos se organizan y se administran en un SO

b. ¿Cuál es la estructura básica de los File System en GNU/Linux? Mencione los directorios más importantes e indique qué tipo de información se encuentra en ellos. ¿A qué hace referencia la sigla FHS?

-Cuenta con paquetes de software para diferenciar distribuciones, editores de texto, herramientas de networking , paquete de oficina e interfaces graficas

-Directorios mas importantes segun FHS (Filesystem Hierarchy Standard):

- `/` → tope de la estructura (C:/ en windows)
- `/home` → almacena archivos de usuarios
- `/var` → informacion que varia de tamaño
- `/etc` → archivos de configuracion
- `/bin` → archivos binarios y ejecutables
- `/dev` → enlace a dispositivos
- `/usr` → aplicaciones de usuarios

al instalar un archivo las diferentes "partes" o características se encuentran en sus debidos directorios

c. Mencione sistemas de archivos soportados por GNU/Linux.

-ext2, ext3, FAT, FAT3, NTFS

d. ¿Es posible visualizar particiones del tipo FAT y NTFS (que son de Windows) en GNU/Linux?

-Linux no permite ver contenido de particiones de Windows

7. Particiones:

a. Definición. Tipos de particiones. Ventajas y Desventajas.

-Cada particion es una division presente en una sola unidad fisica de almacenamiento de datos (disco).

-Ventajas y Desventajas:

- Separa los datos del usuario de las aplicaciones y del SO
- Puede ubicar el Kernel en una particion read only
- Backup general del sistema

-Muchas particiones puede generar fragmentacion (archivos grandes no entren en el tamaño de una de las particiones)

-Tipos:

-Particion Primaria: Divisiones primarias (maximo 4) del disco que dependen de una tabla de particiones detectada por el SO al arrancar.

-Particion Secundaria o extendida: Se creo para obtener mas de 4 particiones en el disco duro y se utilizan unicamente para el almacenamiento de datos ya que no se puede instalar un SO en ellas

-Particion Logica: Se hacen dentro de la particion secundaria o extendida y se le asigna un tamaño y tipo de sistema de archivos

b. ¿Cómo se identifican las particiones en GNU/Linux? (Considere discos IDE, SCSI y SATA).

-Disqueteras:

-Primera : `/dev/fd0` ; Segunda: `/dev/fd1`

-Discos:

-Primer disco (todo): `/dev/hda`

-Particiones primarias:

-Primera: `/dev/hda1` ; Segunda: `/dev/hda2` ; Tercera: `/dev/hda3` ; Cuarta: `/dev/hda4`

-Particiones logicas:

-Primera: `/dev/hda5` ; Todas las demas: `/dev/hda#`

-Segundo disco: `/dev/hdb`

-Particiones primarias:

-Primera: `/dev/hdb1` ; Segunda: `/dev/hdb2` ; Tercera: `/dev/hdb3` ; Cuarta: `/dev/hdb4`

-Particiones logicas:

-Primera: `/dev/hdb5` ; Todas las demas: `/dev/hdb#`

-Discos SCSI:

-Primer disco SCSI: `/dev/sda`

-Segundo disco SCSI: `/dev/sdb`

-Tercero y sucesivamente: `/dev/sd#`

c. ¿Cuántas particiones son necesarias como mínimo para instalar GNU/Linux? Nómbralas indicando tipo de partición, identificación, tipo de File System y punto de montaje.

-Mínimo una para instalar el (/) aunque se recomiendan dos para también instalar el SWAP

d. Dar ejemplos de diversos casos de particionamiento dependiendo del tipo de tarea que se deba realizar en su sistema operativo.

e. ¿Qué tipo de software para particionar existe? Mencínelos y compare.

-Destructivos (permiten crear y eliminar particiones → fdisk) y No destructivos (permiten crear, eliminar y modificar particiones generalmente desde la interfaz → fips, gparted)

8. Arranque (bootstrap) de un Sistema Operativo:

a. ¿Qué es el BIOS? ¿Qué tarea realiza?

-La BIOS es el Basic Input Output System y se encarga del manejo y configuración de la placa base y sus componentes. Inicia los componentes del HW y lanza el SO cuando prendes la computadora ya que la BIOS es lo primero que se carga en él

b. ¿Qué es UEFI? ¿Cuál es su función?

-UEFI es Unified Extensible Firmware Interface) y es un nexo entre el sistema operativo y el firmware que fue desarrollado por Intel. Es una alternativa para reemplazar la BIOS

c. ¿Qué es el MBR? ¿Qué es el MBC?

-El MBR es el Master Boot Record y se encuentra en el primer sector del disco (cilindro 0, cabeza 0, sector 1). Es un registro de arranque principal que se carga en memoria y se ejecuta. Contiene un código de arranque llamado MBC (Master Boot Code) y una marca de 2 bytes que indica su presencia o también puede tener una tabla de particiones

d. ¿A qué hacen referencia las siglas GPT? ¿Qué sustituye? Indique cuál es su formato.

-GPT es GUID Partition Table, y es un solucionador para las limitaciones del MBR, las cuales pueden ser cantidad de particiones, o tamaño de las mismas. GPT establece una ubicación y formato para la tabla de particiones en el disco duro

e. ¿Cuál es la funcionalidad de un "Gestor de Arranque"? ¿Qué tipos existen? ¿Dónde se instalan? Cite gestores de arranque conocidos.

-Es un bootloader que se encarga exclusivamente de preparar todo lo necesario para que el SO funcione. Se instala normalmente en el MBR y nos permitira elegir el sistema operativo a arrancar. En Linux existen distintos tipos de gestor de arranque: LILO (Linux Loader), GAG (Gestor Arranque Grafico)

f. ¿Cuáles son los pasos que se suceden desde que se prende una computadora hasta que el Sistema Operativo es cargado (proceso de bootstrap)?

-Lo primero que sucede es que se arranca la BIOS y se encarga de iniciar la carga del SO a traves del MBC ubicado en el MBR. Carga el programa de booteo y el gestor de arranque toma presencia y carga el Kernel, el cual prueba y hace disponibles los dispositivos para darle el control al init

g. Analice el proceso de arranque en GNU/Linux y describa sus principales pasos.

-Cuando se arranca la computadora, el BIOS se ejecuta realizando el POST (Power-on self-test), que incluye rutinas que, entre otras actividades, fijan valores de las señales internas, y ejecutan test internos (RAM, el teclado, y otros dispositivos a través de los buses ISA y PCI). Luego se lee el primer sector del disco llamado MBR que se carga en memoria y se ejecuta el MBC. Este puede ser de varios tipos, en el caso de Linux, el más frecuentemente usado era LILO, pero ya hace tiempo que se usa en bastantes distribuciones un cargador alternativo, llamado GRUB. Otros Sistemas Operativos tienen su propio programa cargador. Usaremos LILO en la descripción, pues es más ilustrativo.

En el caso concreto de LILO, lo que se carga en el sector de arranque es una parte de éste, denominada "first stage boot loader" (primer paso del cargador de inicio). Su misión es cargar y ejecutar el segundo paso del cargador de inicio.

Esta segunda parte suele mostrar una selección de Sistemas Operativos a cargar, procediendo a cargar a continuación el sistema escogido por el usuario (o bien el que se haya predeterminado como sistema por defecto, tras un tiempo de espera, si no escogemos nada). Esta información está

incluida dentro del cargador de inicio y, para introducirla, se usa la orden 'lilo' que a su vez usa el contenido de '/etc/lilo.conf'. Todo ello sucede, por supuesto, con el ordenador ya en marcha.

Una vez LILO ha cargado el "kernel" (núcleo) de Linux, le pasa el control a éste. Al cargarlo, le ha pasado algunos parámetros. De éstos, el más importante es el que le dice al núcleo qué dispositivo usar como sistema de ficheros raíz, es decir, lo que en UNIX se denomina '/'. En un ordenador de sobremesa, la raíz sería típicamente una partición de un disco duro, pero en sistemas incrustados es frecuente usar como raíz una partición virtual basada en memoria (Flash, RAM,...). Si el núcleo ha conseguido montar el sistema de ficheros raíz, lo siguiente a ejecutar es el programa 'init'. Sólo si dicho programa es estático (es decir, no usa librerías de funciones externas), no será necesario tener acceso a dichas librerías en la raíz. La librería básica en todo sistema GNU/Linux es la librería estándar C, "glibc". En un sistema mínimo, es decir, con una funcionalidad muy concreta, inmutable y sencilla, con tener solamente el programa 'init' enlazado estáticamente sería suficiente (y el núcleo, claro). En ese caso, init sería en realidad nuestro programa de aplicación al completo.

En general, 'init' es sólo el programa que se encarga de arrancar el resto de procesos que la máquina debe ejecutar. Entre sus tareas está el comprobar y montar sistemas de archivos, así como iniciar programas servidores (daemons) para cada función necesaria. Otra tarea importante es la de arrancar procesos 'getty' cuya misión es proporcionar consolas donde poder registrarse y entrar en el sistema. Las órdenes a seguir por 'init' están en el fichero '/etc/inittab'. A partir de ese punto, y en función del sistema de inicialización utilizado (el más frecuente es el denominado "System V") el proceso seguido por 'init' es distinto, pero en el fondo obedece más a un factor de forma, es decir, a una estrategia de ordenamiento de los "scripts" de inicialización de los distintos procesos que a un factor de fondo. Una vez iniciados todos los servidores y procesos de entrada de usuario, o bien estamos delante de una consola de texto en la que el ordenador nos pide que nos identifiquemos, o bien estamos ante una consola gráfica que nos pide lo mismo, o bien estamos ante una pantalla llena de opciones sobre qué ejecutar (escuchar música, ver películas, por ejemplo) si el sistema arranca bajo un usuario predeterminado y no nos pide registrarnos. Esto es, si es que hablamos de un ordenador de sobremesa que, típicamente, nos

ofrece una interfaz basada en dispositivos de entrada (teclado, ratón, mando a distancia) y de salida (monitor, TV, audio) para interactuar con él. Pero si el ordenador que se ha iniciado es un dispositivo con una funcionalidad concreta y su misión es controlar una serie de procesos y accedemos a él a través de medios indirectos (como pueda ser un navegador Web), el ordenador se inicia cuando está en disposición de prestar sus servicios, aun cuando no haya una indicación visual de dicho estado.

h. ¿Cuáles son los pasos que se suceden en el proceso de parada (shutdown) de GNU/Linux?

-Una vez se ejecuta el comando shutdown pueden ocurrir 2 escenarios:

- 1) Usuario unico del sistema, debe terminar todos los programas en ejecucion y hacer un log out de tdoas las sesiones de todos lados o mantener la sesion si ya existe una pero cambiando de directorio de trabajo para evitar problemas de demontajed el fyle system. Se ejecuta el comando shutdown -h now y se para
- 2) Muchos usuario, se utiliza el comando shutdown -h +time mensaje, donde time es el numero de minutos en el que la detenida del programa se posterga y el mensaje explica por que se esta haciendo el shutdown. Advierte a todos los usuarios que el sistema se apagara en X tiempo y lo mejor seria que hagan una desconexion o logout para no perder informacion

i. ¿Es posible tener en una PC GNU/Linux y otro Sistema Operativo instalado? Justifique.

-Si , particionando el disco y en cada particion tener un file system diferente

9. Archivos y editores:

a. ¿Cómo se identifican los archivos en GNU/Linux?

-El sistema de archivos nos permite que Linux maneje los archivos, los cuales tienen un nombre entre 1 y 255 caracteres y es case sensitive. La base del sistema de archivos de linux es una estructura empleada por el SO para almacenar informacion en un dispositivo fisico

b. Investigue el funcionamiento de los editores vim, nano y mcedit, y los comandos cat, more y less.

-Algunas pruebas de vim escribiendo en el txt y dps mostrarlo de diferentes modos:

```
~
"archivo2.txt" [New] 2L, 24B written
agus@agus:~$ cat archivo2.txt
hola escribo en el txt
agus@agus:~$ _
```

```
agus@agus:~$ cat archivo2.txt
hola escribo en el txt

agus@agus:~$ more archivo2.txt
hola escribo en el txt

agus@agus:~$ less archivo2.txt
hola escribo en el txt

archivo2.txt (END)
```

c. Cree un archivo llamado "prueba.exe" en su directorio personal usando el vim. El mismo debe contener su número de alumno y su nombre.

- vim prueba.exe y escribo 23005/2 - Agustin Gonzalez , lo muestro:

```
~
"prueba.exe" [New] 2L, 28B written
agus@agus:~$ cat prueba.exe
23005/2 - AGUSTIN GONZALEZ
agus@agus:~$ _
```

d. Investigue el funcionamiento del comando file. Pruébelo con diferentes archivos. ¿Qué diferencia nota?

-Diferentes archivos muestra diferentes tipos, si mostras un ejecutable tambien cambia:

```
agus@agus:~$ file prueba.exe
prueba.exe: ASCII text
agus@agus:~$ file archivo2.txt
archivo2.txt: ASCII text
agus@agus:~$ file /bin/ls
/bin/ls: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked
a574bbc8ef951ddde9b21, for GNU/Linux 3.2.0, stripped
agus@agus:~$ _
```

e. Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el uso de archivos:

i. cd → cambias de carpeta en la que estas parado

ii. mkdir → creas carpeta

iii. rmdir → borras carpeta

```
mkdir carpeta1  
rmdir carpeta1
```

iv. ln → crea enlaces

```
ln prueba.exe enlace  
ln -s prueba.exe enlace_simbolico
```

v. tail → muestra ultimas lineas del archivo

```
agus@agus:~$ tail prueba.exe  
23005/2 - AGUSTIN GONZALEZ
```

vi. locate → busca archivos

vii. ls → lista archivos

```
agus@agus:~$ locate prueba.exe  
/var/lib/locate/locate.db: No such file or directory  
agus@agus:~$ ls -l  
total 12  
-rw-rw-r-- 1 agus agus 24 Aug 26 22:19 archivo2.txt  
-rw-rw-r-- 1 agus agus  0 Aug 26 22:18 archivo.txt  
-rw-rw-r-- 1 agus agus  0 Aug 26 22:18 Archivo.txt  
-rw-rw-r-- 2 agus agus 28 Aug 26 22:23 enlace_duro  
lrwxrwxrwx 1 agus agus 10 Aug 26 22:28 enlace_simbolico -> prueba.exe  
-rw-rw-r-- 2 agus agus 28 Aug 26 22:23 prueba.exe
```

viii. pwd → muestra direccion del directorio actual

ix. cp → copia el primer archivo en el 2do que lo crea

```
cp prueba.exe copiaprueba.exe
```

x. mv → mueve o renombra (si le pasas direccion lo mueve, si le pasas un archivo lo copia)

```
mv prueba.exe carpeta1
```

```
mv copiaprueba.exe nuevonombre.exe
```

xi. find → *busqueda avanzada (find ~ -name "prueba.exe")*

10. Indique qué comando es necesario utilizar para realizar cada una de las siguientes acciones. Investigue su funcionamiento y parámetros más importantes:

a. Cree la carpeta ISOCSO

b. Acceda a la carpeta

c. Cree dos archivos con los nombres isocso.txt e isocso.csv

d. Liste el contenido del directorio actual

e. Visualizar la ruta donde estoy situado

```
agusiso@agus:~$ mkdir ISOCSO
agusiso@agus:~$ cd ISOCSO
agusiso@agus:~/ISOCSO$ vim isocso.txt
agusiso@agus:~/ISOCSO$ vim isocso.csv
agusiso@agus:~/ISOCSO$ ls
isocso.csv  isocso.txt
agusiso@agus:~/ISOCSO$ pwd
/home/agusiso/ISOCSO
```

f. Busque todos los archivos en los que su nombre contiene la cadena "iso*"

g. Informar la cantidad de espacio libre en disco

h. Verifique los usuarios conectados al sistema

```

agusiso@agus:~$ find . -type f -name 'iso*'
./ISOCS0/isocso.txt
./ISOCS0/isocso.csv
agusiso@agus:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
none            3.9G   0    3.9G   0% /usr/lib/modules/6.6.87.2-microsoft-st
andard-WSL2
none            3.9G  4.0K   3.9G   1% /mnt/wsl
drivers         237G   82G   156G  35% /usr/lib/wsl/drivers
/dev/sdd        1007G  1.7G   955G   1% /
none            3.9G   76K   3.9G   1% /mnt/wslg
none            3.9G   0    3.9G   0% /usr/lib/wsl/lib
rootfs          3.8G  2.7M   3.8G   1% /init
none            3.9G  532K   3.9G   1% /run
none            3.9G   0    3.9G   0% /run/lock
none            3.9G   0    3.9G   0% /run/shm
none            3.9G   76K   3.9G   1% /mnt/wslg/versions.txt
none            3.9G   76K   3.9G   1% /mnt/wslg/doc
C:\             237G   82G   156G  35% /mnt/c
tmpfs           3.9G   16K   3.9G   1% /run/user/1000
agusiso@agus:~$ who
agusiso pts/1          2025-08-28 12:03

```

i. Editar a el archivo isocso.txt e ingresar Nombre y Apellido

j. Mostrar en pantalla las últimas líneas de un archivo.

11. Investigue el funcionamiento, parámetros y ubicación (directorio) de los siguientes comandos:

**→ man → dmesg → shutdown → lspci → reboot → at → halt → netstat →
uname → head → tail**

-man: muestra el manual de cualquier comando en Linux. man -k palabra, busca comando relacionados con palabra, man -f comando, descripción

-dmesg: Examina o controla el buffer ring del kernel. Muestra todos los mensajes del ring del buffer. con parametro -h se hace mas legible y con -w se hace un script para filtrar los eventos

-Shutdown: se utiliza para apagar o reiniciar Linux desde la terminal de forma segura (shutdown [OPTIONS][TIME][MESSAGE])

-lspci: imprime listas con informacion de todos los buses y dispositivos del sistema

-reboot: reinicia el SO, -reboot o -r

-at: permite programar tareas para que se ejecuten en determinada fecha y hora, at [hora] [fecha]

-halt: apaga el ordenador, pudiese ser con opciones -d,-h ...

- d	No escribir registro wtmp (en el archivo /var/log/wtmp) El flag -n implica -d
- h	Poner todos los discos duros del sistema en modo de espera antes de que el sistema se detenga o apague
- n	No sincronizar antes de reiniciar o detener
- i	Apagar todas las interfaces de red.
- p	Cuando detenga el sistema, lo apaga también. Esto es por defecto cuando el halt se llama como poweroff.
- w	No reiniciar o detener, sólo escribir el registro wtmp (en el archivo /var/log/wtmp)

-netstat: muestra las conexiones de red entrantes y salientes, tablas de enrutamiento y serie de estadísticas de interfaz de red

-uname: \$uname muestra la palabra Linux extraída del kernel, \$uname -r muestra información de la versión del kernel, \$uname -v muestra la fecha de cuando la versión del kernel fue liberada

-head: muestra las primeras diez líneas un archivo, head -nN al modificar N puedes mostrar más o menos o head -n 20 archivo, muestra las primeras 20 de archivo

-tail: muestra las últimas diez líneas un archivo, tail -n 20 archivo, muestra las primeras 20 de archivo

12. 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.

-Es un programa en ejecución. Programa ≠ Proceso. es dinámico, tiene PC, su ciclo de vida comprende desde que se solicita ejecutarse hasta que se termina. PID es Process ID y el PPID es la identificación del proceso padre. Todos tienen PPID excepto por los procesos de systemd o init, que son el primer proceso en ejecutarse y de donde derivan todos los demás

b. Investigue el funcionamiento, parámetros y ubicación (directorio) de los siguientes comandos relacionados a procesos. En caso de que algún comando no venga por defecto en la distribución que utiliza deberá proceder a instalarlo: i. top ii. htop iii. ps iv. pstree v. kill vi. pgrep vii. pkill viii. killall ix. renice x. xkill xi. atop xii. nice

Comando	Función	Parámetros comunes	Ubicación
top	Monitorea procesos en tiempo real (uso CPU/memoria).	q (salir), k (matar proceso).	/usr/bin/top
htop	Versión mejorada de top, con interfaz interactiva.	F6 cambiar sort, F9 matar, F10 salir.	/usr/bin/htop
ps	Muestra procesos en ejecución.	ps aux, ps -ef.	/bin/ps
pstree	Muestra procesos en forma de árbol jerárquico.	pstree -p (muestra PIDs).	/usr/bin/pstree
kill	Envía señales a procesos (ej. terminar).	kill -9 PID, kill -15 PID.	/bin/kill
pgrep	Busca procesos por nombre o atributos.	pgrep nombre, pgrep -u usuario.	/usr/bin/pgrep
pkill	Mata procesos por nombre (más flexible que kill).	pkill -9 nombre.	/usr/bin/pkill
killall	Mata todos los procesos que coinciden con un nombre.	killall nombre.	/usr/bin/killall
renice	Cambia la prioridad de un proceso en ejecución.	renice -n 10 -p PID.	/usr/bin/renice
xkill	Cierra ventanas gráficas con un clic (X11).	xkill (clic en ventana).	/usr/bin/xkill
atop	Monitorea procesos y recursos en detalle (avanzado).	atop -m (memoria), atop -c (procesos).	/usr/bin/atop
nice	Ejecuta un proceso con prioridad modificada.	nice -n 10 comando.	/usr/bin/nice

13. 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* (*initial ram disk*).
6. Se monta el *initrd* como sistema de archivos raíz y se inicializan componentes esenciales (por ejemplo, el *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 ser usado.

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. Tiene como función cargar todos los subprocesos necesarios para que funcione correctamente el SO. Se ejecuta desde `/sbin/init` y posee el PID 1. En SysV init se configura en el archivo */etc/inittab* y no tiene padre y es el padre de todos los procesos (*pstree*) / Es el encargado de montar los filesystems y hacer disponible a los demás dispositivos

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

-Son los modos en los que arranca el SO y se divden en niveles. Cada runlevel es rtesponsible de iniciar o parar una serie de archivos ya sea entrar (arranque) o salir (apagado) del releve

-Se define: `id:runlevels:accion:proceso`

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

-**id**: identifica entrada en inittab (1 a 4 caracteres)

-**runlevels**: el o en los runlevels en los que se realiza la accion

-**accion**: indica como se ejecutara el proceso (wait,initdefault,off ,etc)

-**proceso**: comando exacto que sera ejecutado

**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?**

- 0 → halt (parada o apagado).
- 1 → single-user mode (modo monousuario).
- 2 → multi-user without network support (multiusuario sin soporte de red).
- 3 → multi-user console mode (modo multiusuario en consola).
- 4 → N/A (no se utiliza).
- 5 → X11 (modo multiusuario con entorno gráfico basado en X.org).
- 6 → reboot (reinicio).

-En sistemas que usan SysVinit, el nivel de ejecucion se define en el archivo `/etc/inittab` . No todas siguen estos estandares de manera estricta, en los sitemas modernos ya se reemplazo el SysVinit con systemd, y se reemplazaron los runlevels por targets

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

-Es un archivo de configuracion el cual define el nivel de ejecucion (osea, runlevel) por defecto con el cual el sistema debe arrancar. Especifica como se deben gfestionar ciertos procesos durante el inicio o parado

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

- `$ init <Y>` es el comando para cambiar a otro runlevel. Cuando se reinicie el sistema volverá a su runlevel predeterminado por lo que este cambio no es permanente, sino momentáneo. Si lo quieres cambiar de manera permanente tienes que entrar al archivo de configuración (punto e) y modificarlo

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.

-Se ejecutan cuando init entra en un runlevel y se llama al script rc con un parametro numerico especificando el nivel de ejecucion al que ir. Inicia y detiene los servicios necesarios para llevar el sistema a ese nivel de ejecucion

-Los scripts que se ejecutan se guardan en `/etc/init.d` y en `/etc/rcX.d` (X siendo numero de runlevel) se hacen link simbolicos a los archivos que hay en `init.d`

-Los nombres de los links siguen el patron `[S][K]<orden><nombre>`

-**S**: indica que se debe iniciar el script (con argumento **start**)

-**K**: indica que se debe parar el script (con argumento **stop**)

-**<orden>**: valor numerico en dos digitos para asegurar el orden de ejecucion de los scripts

-**<nombre>**: nombre logico con el que se identifica el script

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

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

a. ¿Qué es systemd?

-Sistema que centraliza la administracion de deomonios y librerias del sistema. El demonio systemd reemplaza al proceso init y es el que tiene PID

1; los runlevels se reemplazan por **targets**; no usan el archivo de configuracion /etc/inittab

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

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

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, proporcionando 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?

- Son el reemplazo de los runlevels y es una forma flexible de agrupar unidades y definir el estado del sistema. Permite a los administradores configurar precisamente que servicios se deben iniciar, detener o reiniciar cuando se cambia el estado del sistema (como pasar de un runlevel a otro en sysv)

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

- Muestra los procesos en ejecución en forma de árbol. La raíz es init o systemd dependiendo del sistema de arranque que utilices

15. Usuarios:

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

-/etc/passwd es el archivo que almacena informacion sobre los usuarios locales del sistema. Se define
username:password:uid:gid:GECOS:/home/dir:shell.

-Username: nombre de usuario de login

-Password: donde se guardan las contraseñas cifradas. Actualmente se guardan en un archivo de texto aparte /etc/shadow

-Uid: Uniq Id, id unico del usuario con el cual se identfiica univocamente en el nivel mas basico

-Gid: Numero de Id de grupo del usuario

-GECOS: texto arbitrario que incluye el nombre real del usaurio e informacion adicional del mismo

-home/dir: es la ubicacion donde se encuentran los datos personales del usuario y archivos de configuracion

-shell: shell por defecto de los procesos y usuarios. /sbin/nologin se usa para bloquear los inicios de sesion en el sistema de forma interactiva

b. ¿A qué hacen referencia las siglas UID y GID? ¿Pueden coexistir UIDs iguales en un

sistema GNU/Linux? Justifique.

-Uid: Uniq Id, id unico del usuario con el cual se identfiica univocamente en el nivel mas basico. Por defecto el SO le pone UID 1000 al primer usuario del sistema y luego le asigna a los demas usuarios un PID en el rango disponible del 1000 en adelante, por lo que no pueden coexistir UIDS iguales

-Gid: Numero de Id de grupo del usuario

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

-Es un superusuario, es como un administrador del mas alto nivel que tiene todo el poder sobre el sistema. Puede anular los privilegios normales del sistema y se utiliza para administrar el mismo. Tiene UID 0 y solo existe un usuario root en el sistema, aunque se pueden crear usuarios con poderes similares agregandoles al grupo sudo

d. Agregue un nuevo usuario llamado isocso a su instalación de GNU/Linux, especifique que su home sea creada en /home/isocso, y hágalo miembro del grupo informatica (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.

```
agusiso@agus:~$ sudo groupadd informatica
[sudo] password for agusiso:
agusiso@agus:~$ sudo groupadd informatica
groupadd: group 'informatica' already exists
agusiso@agus:~$ sudo useradd -m -d /home/isocso -s /bin/bash isocso
agusiso@agus:~$ sudo usermod -aG informatica isocso
agusiso@agus:~$ sudo touch /home/isocso/archivo.txt
agusiso@agus:~$ sudo chown isocso:informatica /home/isocso/archivo.txt
agusiso@agus:~$ ls -l /home/isocso/archivo.txt
```

Verifico el archivo y sus permisos:

```
agusiso@agus:~$ sudo ls -l /home/isocso/archivo.txt
-rw-r--r-- 1 isocso informatica 0 Sep 10 16:47 /home/isocso/archivo.txt
agusiso@agus:~$ |
```

*muestra que el propietario es **isocso**, informatica es su grupo*

Y lo borro, verificando el archivo de grupo y de usuarios que no devuelve nada:

```
agusiso@agus:~$ sudo userdel isocso
userdel: user 'isocso' does not exist
agusiso@agus:~$ grep isocso /etc/passwd
agusiso@agus:~$ grep isocso /etc/group
```

e. Investigue la funcionalidad y parámetros de los siguientes comandos:

i.useradd y adduser v. groupadd ii.usermod vi. who iii.userdel vii. groupdel iv.su viii. passwd

-useradd:

```
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[uid]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
```

-groupadd:

```
agusiso@agus:~$ 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
```

-usermod:

```

agusiso@agus:~$ 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

```

-who:

```

agusiso@agus:~$ 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.

```

-userdel:

```

agusiso@agus:~$ 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

```

-groupdel:

```

agusiso@agus:~$ 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

```

-su:

```

agusiso@agus:~$ 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).

```

-passwd:


```

agusiso@agus:~$ 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

```

16. FileSystem y permisos:

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

-Los permisos sobre archivos tienen 3 categorías de usuario a la cual se le aplican permisos. El archivo pertenece al usuario, el que lo creo. El archivo también pertenece a un grupo, generalmente el grupo primario del usuario que creo el archivo. Se pueden establecer diferentes permisos para el usuario y el grupo propietario, y también a los otros usuarios del sistema que no sean ni el usuario que creo el archivo ni un miembro del mismo grupo que este usuario. Los permisos de usuario anulan los permisos de grupo, que anulan otros permisos

Tipos de permiso:

- Read(r): Archivos: puede leerse el contenido; Directorios: detalla los contenidos del directorio
- Write (w): Archivos: puede modificarse el contenido ; Directorios: puede crear o eliminar archivos dentro del directorio
- Execute (x): Archivos: puede ejecutar archivos como comandos ; Directorios: puede acceder al contenido del directorio

b. Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con los permisos en GNU/Linux: i. *chmod* ii. *chown* iii. *chgrp*

-chmod:

```

agusiso@agus:~$ chmod --help
Usage: chmod [OPTION]... MODE[,MODE]... FILE...
       or:  chmod [OPTION]... OCTAL-MODE FILE...
       or:  chmod [OPTION]... --reference=RFILE FILE...
Change the mode of each FILE to MODE.
With --reference, change the mode of each FILE to that of RFILE.

  -c, --changes          like verbose but report only when a change is made
  -f, --silent, --quiet  suppress most error messages
  -v, --verbose          output a diagnostic for every file processed
                        --no-preserve-root  do not treat '/' specially (the default)
                        --preserve-root    fail to operate recursively on '/'
                        --reference=RFILE  use RFILE's mode instead of specifying MODE values.
                                           RFILE is always dereferenced if a symbolic link.
  -R, --recursive       change files and directories recursively
  --help                display this help and exit
  --version              output version information and exit

Each MODE is of the form '[ugoa]*([-+]=([rwxXst]*|[ugo]))+|[-+]=[0-7]+'.
```

-chown:

```

agusiso@agus:~$ chown --help
Usage: chown [OPTION]... [OWNER][:[GROUP]] FILE...
       or:  chown [OPTION]... --reference=RFILE FILE...
Change the owner and/or group of each FILE to OWNER and/or GROUP.
With --reference, change the owner and group of each FILE to those of RFILE.

  -c, --changes          like verbose but report only when a change is made
  -f, --silent, --quiet  suppress most error messages
  -v, --verbose          output a diagnostic for every file processed
                        --dereference      affect the referent of each symbolic link (this is
                                           the default), rather than the symbolic link itself
  -h, --no-dereference   affect symbolic links instead of any referenced file
                                           (useful only on systems that can change the
                                           ownership of a symlink)
                        --from=CURRENT_OWNER:CURRENT_GROUP
                                           change the owner and/or group of each file only if
                                           its current owner and/or group match those specified
                                           here. Either may be omitted, in which case a match
                                           is not required for the omitted attribute
                        --no-preserve-root  do not treat '/' specially (the default)
                        --preserve-root    fail to operate recursively on '/'
                        --reference=RFILE  use RFILE's owner and group rather than specifying
                                           OWNER:GROUP values. RFILE is always dereferenced.
  -R, --recursive       operate on files and directories recursively

The following options modify how a hierarchy is traversed when the -R
option is also specified. If more than one is specified, only the final
one takes effect.

  -H          if a command line argument is a symbolic link
              to a directory, traverse it
  -L          traverse every symbolic link to a directory
              encountered
  -P          do not traverse any symbolic links (default)

  --help      display this help and exit
  --version   output version information and exit

Owner is unchanged if missing. Group is unchanged if missing, but changed
to login group if implied by a ':' following a symbolic OWNER.
OWNER and GROUP may be numeric as well as symbolic.
```

```
Owner is unchanged if missing. Group is unchanged if missing, but changed to login group if implied by a ':' following a symbolic OWNER. OWNER and GROUP may be numeric as well as symbolic.
```

Examples:

```
chown root /u      Change the owner of /u to "root".
chown root:staff /u Likewise, but also change its group to "staff".
chown -hR root /u   Change the owner of /u and subfiles to "root".
```

-chgrp:

```
agusiso@agus:~$ chgrp --help
Usage: chgrp [OPTION]... GROUP FILE...
or:  chgrp [OPTION]... --reference=RFILE FILE...
Change the group of each FILE to GROUP.
With --reference, change the group of each FILE to that of RFILE.

-c, --changes          like verbose but report only when a change is made
-f, --silent, --quiet  suppress most error messages
-v, --verbose          output a diagnostic for every file processed
--dereference          affect the referent of each symbolic link (this is
                        the default), rather than the symbolic link itself
-h, --no-dereference   affect symbolic links instead of any referenced file
                        (useful only on systems that can change the
                        ownership of a symlink)
--no-preserve-root     do not treat '/' specially (the default)
--preserve-root        fail to operate recursively on '/'
--reference=RFILE      use RFILE's group rather than specifying a GROUP.
                        RFILE is always dereferenced if a symbolic link.
-R, --recursive        operate on files and directories recursively

The following options modify how a hierarchy is traversed when the -R
option is also specified. If more than one is specified, only the final
one takes effect.

-H                    if a command line argument is a symbolic link
                        to a directory, traverse it
-L                    traverse every symbolic link to a directory
                        encountered
-P                    do not traverse any symbolic links (default)

--help               display this help and exit
--version            output version information and exit

Examples:
chgrp staff /u      Change the group of /u to "staff".
chgrp -hR staff /u  Change the group of /u and subfiles to "staff".
```

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?

Cuando usamos **chmod** con números, indicamos un dígito por cada categoría (usuario, grupo, otros)

Ejemplos:

`chmod 777 archivo` → `rw-rw-rw-` (todos pueden todo).

`chmod 755 archivo` → `rw-r-xr-x` (usuario todo, grupo y otros solo lectura/ejecución).

`chmod 644 archivo` → `rw-r--r--` (usuario lectura/escritura, grupo y otros solo lectura).

`chmod 600 archivo` → `rw-----` (solo el dueño puede leer/escribir).

Los permisos básicos son:

r = read (4) → permiso de lectura.

w = write (2) → permiso de escritura.

x = execute (1) → permiso de ejecución.

Cada uno tiene un valor **numerico** y la suma forma el valor de permiso:

`rwX = 4 + 2 + 1 = 7` (todo)

`rw- = 4 + 2 = 6` (read y write)

`r-x = 4 + 1 = 5` (read y execute)

`r-- = 4` (solo read)

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

-Si, el usuario root, ya que tiene los maximos privilegios y poderes y puede acceder a cualquier directorio/archivo en el sistema sin importar nada

e. Explique los conceptos de "full path name" (path absoluto) y "relative path name" (path relativo). De ejemplos claros de cada uno de ellos.

-Full path name (path absoluto): Nombre calificado que empieza con '/' raiz y especifica cada subdirectorio que se atraviesa para llegar, representa en forma exclusiva un solo archivo. Cada archivo del sistema de archivos tiene un unico nombre de ruta absoluta. Un nombre de archivo con una barra '/' como primer caracter es el nombre de la ruta absoluta `/var/log/messages`

-Relative path name (path relativo): Identifica un archivo unico y especifica solo la ruta necesaria para llegar al archivo desde el directorio de trabajo. Un nombre de archivo que no tenga una '/' como primer caracter es un nombre de ruta relativa. Por ejemplo un usuario en el directorio `/var` podria referirse de forma relativa a los mensajes como `log/messages`

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.

- `pwd` , imprime la ruta absoluta en la que se esta trabajando en el directorio actual. Para ingresar al directorio personal sin escribir todo el path se puede utilizar `cd` , o `cd ~` para acortar rutas. `'~'` permite tambien acceder al directorio personal de otro usuario.

`-cd ~Agus/Desktop` accederia al directorio de Escritorio de Agus

g. Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el uso del FileSystem: i. umount ii. du iii. df iv. mount v. mkfs vi. fdisk (con cuidado) vii. write viii. losetup ix. stat

Comando	Función	Parámetros comunes	Ubicación
umount	Desmonta un sistema de archivos previamente montado.	umount /dev/sdX, umount /mnt/punto.	/bin/umount
du	Muestra el uso de espacio en disco de archivos y directorios.	du -h (legible), du -sh carpeta.	/usr/bin/du
df	Muestra información del uso de discos y particiones montadas.	df -h (legible), df -T (tipo FS).	/bin/df
mount	Monta un sistema de archivos en un punto de montaje.	mount /dev/sdX /mnt, mount -t tipo.	/bin/mount
mkfs	Crea un sistema de archivos en una partición o dispositivo.	mkfs.ext4 /dev/sdX, mkfs.vfat /dev/sdX.	/sbin/mkfs
fdisk	Herramienta para gestionar particiones de disco (con cuidado).	fdisk -l (lista), fdisk /dev/sdX (editar particiones).	/sbin/fdisk
write	Permite enviar mensajes a otro usuario conectado al sistema.	write usuario (abrir sesión de mensaje).	/usr/bin/write
losetup	Configura y maneja dispositivos de bucle (loop devices).	losetup -f (primer loop libre), losetup -a (lista).	/sbin/losetup
stat	Muestra información detallada de un archivo (permisos, inodos, fechas).	stat archivo.txt.	/usr/bin/stat

17. Procesos:

a. ¿Qué significa que un proceso se está ejecutando en Background? ¿Y en Foreground?

-Background: proceso que se ejecuta en segundo plano sin bloquear la terminal, permitiendo al usuario seguir utilizando la terminal para otros comandos mientras el proceso sigue en ejecución. No recibe entradas de teclado directamente pero sigue ejecutandose

-Foreground: proceso que se ejecuta en primer plano activo directamente en la terminal desde la cual se inicio. Ocupa la terminal y entradas de teclado directamente del usuario, mostrando la salida de la misma. El usuario no puede interactuar con la terminal para realizar otras acciones hasta que este proceso termine

b. ¿Cómo puedo hacer para ejecutar un proceso en Background? ¿Como puedo hacer para pasar un proceso de background a foreground y viceversa?

-Para ejectuar un proceso en Background se agrega un `&` al final, `cat archivo.txt &`

-Para enviarlo de Foreground a Background usas Control Z para suspenderlo y despues ejecutas `'bg'` para enviarlo al segundo plano. Para

traerlo del Background se utiliza `'fg'`

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

-Cumple la función de encadenar comandos, permite usar la salida de un comando como entrada de otro y procesarlos en etapas aplicando diferentes comandos a los datos de manera sucesiva

`ps aux | grep firefox` → Ver todos los procesos y luego solo los que contienen firefox

`ls -l | wc -l #` → Listar el contenido del dir actual y luego contar las líneas

d. 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 (stdin)
- Salida estándar (canal 1) envía una salida normal al terminal (stdout)
- Error estándar (canal 2) envía mensajes de error al terminal (stderr)

-Redireccionamiento de la salida a un archivo: El redireccionamiento de E/S reemplaza los destinos de los canales predeterminados con nombres de archivos que representan dispositivos. Con el uso del mismo los mensajes de error y salida de un proceso que se envían a la terminal se terminan pudiendo capturar en un archivo, enviarse a un dispositivo o descartarse. El stdout evita que la salida de un proceso aparezca en la terminal, además si el archivo no existe se creará. Si el archivo existe y el redireccionamiento no es uno que se agregue al archivo, el archivo se sobrescribirá. El archivo especial /dev/null descartará discretamente la salida del canal redirigido a él y siempre es un archivo vacío

Operador	Función	Ejemplo
>	Redirige STDOUT a un archivo (sobrescribe).	ls > salida.txt
>>	Redirige STDOUT a un archivo (añade al final).	ls >> salida.txt
2>	Redirige STDERR a un archivo (sobrescribe).	ls inexistente 2> error.txt
2>>	Redirige STDERR a un archivo (añade al final).	ls inexistente 2>> error.txt
&>	Redirige STDOUT y STDERR a un archivo (sobrescribe).	comando &> todo.txt
&>>	Redirige STDOUT y STDERR a un archivo (añade al final).	comando &>> todo.txt
n>	Redirige un descriptor de archivo específico a un archivo (sobrescribe).	comando 3> archivo.txt
n>>	Redirige un descriptor de archivo específico a un archivo (añade al final).	comando 3>> archivo.txt

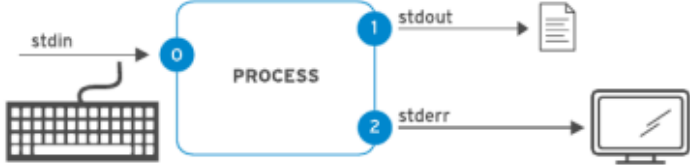
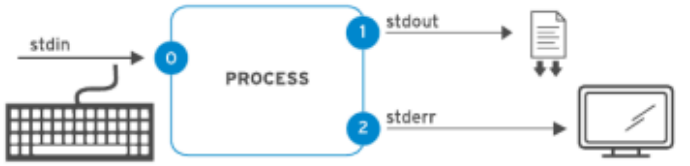


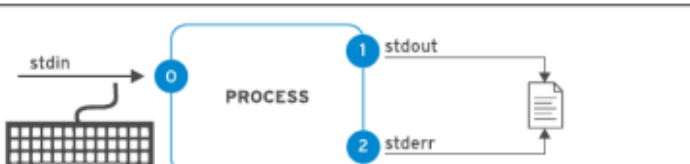
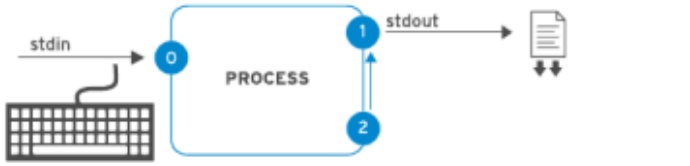
Uso	Explicación	Ayuda visual
>file	redirigir stdout para sobrescribir un archivo	
>>file	redirigir stdout para agregar a un archivo	
2>file	redirigir stderr para sobrescribir un archivo	
2>/dev/null	descartar mensajes de error stderr mediante el redireccionamiento a /dev/null	
>file 2>&1	redirigir stdout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

tabla del GH no encuentre ningun ejemplo mas grafico

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

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

-Hace referencia al proceso de agrupar varios archivos y directorios en un solo archivo comprimido/no comprimido. Facilita la distribución, almacenamiento y transferencia de este grupo de archivos. El 'paquete' resultante puede ser descomprimido o desempaquetado y con eso recuperarías los archivos originales que se agruparon

b. Seleccione 4 archivos dentro de algún directorio al que tenga permiso y sume el tamaño de cada uno de estos archivos.

a. Cree un archivo empaquetado conteniendo estos 4 archivos y compare los tamaños de los mismos.

b. ¿Qué característica nota?

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

```
drwxr-x--- 5 agusiso agusiso 4096 Sep 10 16:43 agusiso
agusiso@agus:~$ vim archivo1.txt
agusiso@agus:~$ vim archivo2.txt
agusiso@agus:~$ vim archivo3.txt
agusiso@agus:~$ vim archivo4.txt
agusiso@agus:~$ ls
ISOCS0 archivo1.txt archivo2.txt archivo3.txt archivo4.txt
agusiso@agus:~$ du --bytes
0      ./landscape
21     ./ISOCS0
0      ./cache
11828 .
```

```
agusiso@agus:~$ tar -cvf paquete.tar archivo1.txt archivo2.txt archivo3.txt archivo4.txt
archivo1.txt
archivo2.txt
archivo3.txt
archivo4.txt
agusiso@agus:~$ ls lh paquete.tar
ls: cannot access 'lh': No such file or directory
paquete.tar
agusiso@agus:~$ ls -lh paquete.tar
-rw-r--r-- 1 agusiso agusiso 20K Sep 10 17:50 paquete.tar
```

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

-Si quieres empaquetar y comprimir un conjunto de archivos (ejemplo de arriba):

- `tar -czvf archivo.tar archivo1 archivo2 archivo3 archivo4`

- `tar -cjvf archivo.tar archivo1 archivo2 archivo3 archivo4`

- `tar -cJvf archivo.tar archivo1 archivo2 archivo3 archivo4`

j usa *gzip* (.tar.gz), *z* usa *bzip2* (.tar.bz2) y *J* usa *xz* (.tar.xz)

-Si ya se dispone un archivo empaquetado y se desea comprimir:

- `gzip archivo1.tar archivo2.tar archivo3.tar archivo4.tar`

- `bzip2 archivo1.tar archivo2.tar archivo3.tar archivo4.tar`

e. Investigue la funcionalidad de los siguientes comandos: i. tar ii. grep iii. gzip iv. zgrep v. wc

Comando	Función	Parámetros comunes	Ubicación
tar	Empaqueta y (opcionalmente) comprime múltiples archivos en uno solo.	tar -cvf (crear), tar -xvf (extraer), tar -tvf (listar).	/bin/tar
grep	Busca patrones de texto dentro de archivos o entrada estándar.	grep 'texto' archivo, grep -i (ignorar mayúsculas), grep -r (recursivo).	/bin/grep
gzip	Comprime archivos individuales usando el algoritmo gzip.	gzip archivo.txt (crea archivo.txt.gz), gunzip archivo.gz (descomprime).	/bin/gzip
zgrep	Busca patrones de texto dentro de archivos comprimidos con gzip (.gz).	zgrep 'texto' archivo.gz, zgrep -i (ignorar mayúsculas).	/bin/zgrep
wc	Cuenta líneas, palabras y caracteres de un archivo o entrada estándar.	wc -l (líneas), wc -w (palabras), wc -c (bytes/caracteres).	/usr/bin/wc

19. 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 → Genera una lista de todos los directorios y archivos del directorio de inicio y lo guarda en un archivo llamado prueba

ps > PRUEBA → Guarda un listado con la información de los procesos ejecutados actualmente en un archivo llamado PRUEBA

chmod 710 prueba → Cambia los permisos de acceso al archivo prueba, da todos los permisos al propietario, ejecución al usuario y ninguno a los demás

chown root:root PRUEBA → Intenta cambiar el propietario y el grupo al que pertenece el archivo PRUEBA pero no puede porque no es root. **Changing ownership of 'PRUEBA': Operation not permitted**

chmod 777 PRUEBA → Cambia los permisos de acceso al archivo PRUEBA y le da tanto al propietario, usuario y otros todos los permisos

chmod 700 /etc/passwd → Intenta cambiar los permisos de lectura escritura y ejecución al directorio /etc/passwd pero no deja ya que solo un root pueden. **Changing permissions of '/etc/passwd': Operation not permitted**

`passwd root` → Intenta ver o cambiar la contraseña del usuario root pero no puede ya que solo el mismo root puede hacerlo. *You may not view or modify password information for root*

`rm PRUEBA` → Elimina el archivo PRUEBA

`man /etc/shadow` → No se permite ya que man no puede brindar acceso a las paginas del manual del sistema porque estas dan informacion detallada sobre comandos, funciones, utilidades del sistema y archivos del SO y no sobre directorios. Ademas el permiso es denegado porque intenta acceder al directorio shadow el que actualmente tiene las contraseñas de los usuarios. *man: can't open /etc/shadow: Permission denied*

`find / -name *.conf` → Busca y lista a partir del directorio raiz '/' todos los archivos que contengan .conf

`usermod root -d /home/newroot -L` → Intenta cambiar el directorio de inicio del Usuario root al directorio /home/newroot. No se permite ya que el usuario no posee permisos necesarios.

`cd /root` → Intenta acceder al directorio de inicio del usuario root, no posee permisos necesarios por lo cual no puede hacerlo. *cd: /root : Permission denied*

`rm *` → Se eliminan todos los archivos del directorio en el que esta posicionado el usuario

`cd /etc` → Se situa en el directorio /etc donde estan las configuraciones del sistema

`cp */home -R` → Intenta copiar todos los archivos del directorio actual (/etc porque se paro previamente) al directorio /home de manera recursiva. No es posible ya que solo root tiene permisos para copiar el contenido de /etc. *Cannot create regular file, directory, etc: Permission denied*

`shutdown` → Apaga el sistema aunque necesita permisos sudo para hacerlo. *Command not found*

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

a. Terminar el proceso con PID 23.

`kill 21`

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

`kill -9 1` pero no se podría ya que el proceso del arranque del sistema no puede ser terminado

c. Buscar todos los archivos de usuarios en los que su nombre contiene la cadena ".conf"

```
find /home - name * .conf *
```

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

```
ps > /home/agus/procesos
```

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

i. Usuario: Lectura, escritura, ejecución

ii. Grupo: Lectura, ejecución

iii. Otros: ejecución

```
chmod 751 /home/agus/xxxx
```

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

i. Usuario: Lectura, escritura.

ii. Grupo: Lectura, ejecución

iii. Otros: Ninguno

```
chmod 640 /home/agus/yyyy
```

g. Borrar todos los archivos del directorio /tmp

```
rm/tmp/*
```

h. Cambiar el propietario del archivo /opt/isodata al usuario isocso

```
sudo chown iso2010 /opt/isodata
```

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

```
pwd >> /home/usuario/donde
```

21. 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/AGonzalez
```

```
passwd pepe
```

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`, se creó un directorio en `/home/AGonzalez`

d. Crear un directorio en `/tmp` llamado `miCursada`

```
mkdir /tmp/miCursada
```

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

```
cp -r /var/log* /tmp/miCursada
```

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 AGonzalez:users miCursada/
```

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 miCursada/
```

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

```
su -AGonzalez
```

ingresas con la contraseña previamente creada

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

```
tty
```

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

```
ps -e | wc -l
```

con `wc -l` los listo

k. Verifique la cantidad de usuarios conectados al sistema.

```
who
```

l. Vuelva a la terminal del usuario `root` y envíele un mensaje al usuario anteriormente creado enviándole que el sistema va a ser apagado.

```
su.
```

```
wall "El sistema va a ser apagado"
```

m. Apague el sistema.

```
shutdown -P now
```

22. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes

acciones:

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

```
mkdir 23005_2
```

o. 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
```

p. 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
```

q. 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 cuál puede crear este archivo si ya existe un archivo llamado "LEAME" en este directorio?

```
cd/etc
```

ls > /home/leame Lista los archivos y los sobreescribe (en este caso crea en el archivo leame) y esto ocurre porque LINUX es case sensitive y LEAME y leame son 2 archivos diferentes

r. ¿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.

```
find / -name nombre_archivo_a_buscar
```

```
find / -name "*.caracteristicas_similares"
```

s. Utilizando los conceptos aprendidos en el punto anterior, busque todos los archivos cuya extensión sea .so y almacene el resultado de esta búsqueda en un archivo dentro del directorio creado en el primer inciso. El archivo deberá llamarse ejercicioF.

```
find / -type f -name "*.so" > /home/tu_usuario/23005_2/ejercicioF
```

23. 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:

- 01.** `mkdir iso` → Crea directorio iso en el directorio actual
- 02.** `cd ./iso; ps > f0` → Entra al directorio iso y guarda los procesos activos en el archivo f0
- 03.** `ls > f1` → Guarda en f1 el listado de archivos del directorio donde se encuentra parado
- 04.** `cd /` → Entra al directorio raíz del sistema
- 05.** `echo $HOME` → Muestra la ruta del directorio home del usuario actual
- 06.** `ls -l > $HOME/iso/ls` → Lista los archivos y los guarda dentro de iso
- 07.** `cd $HOME; mkdir f2` → Vuelve a home y crea un directorio llamado f2
- 08.** `ls -ld f2` → Muestra informacion detallada del directorio f2 unicamente
- 09.** `chmod 341 f2` → Cambia los permisos de f2, dueño tiene escritura y ejecucion, grupo tiene lectura y otros tienen ejecucion
- 10.** `touch dir` → Crea un archivo llamado dir
- 11.** `cd f2` → Entra al directorio f2
- 12.** `cd ~/iso` → Entra al directorio iso dentro del home (~)
- 13.** `pwd > f3` → Guarda la ruta absoluta del directorio actual dentro de un nuevo archivo f3
- 14.** `ps | grep 'ps' | wc -l >> ../f2/f3` → Se listan los procesos en ejecucion, grep recibe esa lista y en concreto se cuenta cuantas veces aparece el proceso 'ps' y lo agrega en el archivo f3 dentro del directorio f2
- 15.** `chmod 700 ../f2; cd ..` → Cambia los permisos de f2 y le da al usuario todos los permisos y a todo lo otro nada, Se posiciona un nivel arriba
- 16.** `find . -name etc/passwd` → Busca archivos llamados passwd en el directorio actual (deberia ir sin etc por lo que se muestra un warning)
- 17.** `find / -name etc/passwd` → Busca archivos llamados passwd en todo el sistema a partir de la raíz (deberia ir sin etc por lo que se muestra un warning)

18. `mkdir ejercicio5` → Crea el directorio ejercicio5 en el directorio actual

19.

20.

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 "explicacion_de_ejercicio" dentro del directorio creado en el ejercicio 22 y, para cada una de los comandos que ejecute en la otra sesión, realice una breve explicación de los resultados obtenidos.

b. Complete 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 24.a

20: Copiar el resto de los archivos y directorios que se crearon en este ejercicio al directorio creado en el ejercicio 24.a

19: `cp -r iso ~/ejercicio24.a/` por ejemplo si el directorio estuviera en home

20: `cp -r f0 f1 f2 dir f3 ejercicio5 ~/ejercicio24.a/`

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

19: `cp -r iso ~/ejercicio24.a/` → Copia todo contenido del directorio iso en el directorio 24.a

20: `cp -r f0 f1 f2 dir f3 ejercicio5 ~/ejercicio24.a/` → Copia el contenido de todos los directorios previamente creados en el directorio ejercicio24.a

24. 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. Utilizando la estructura de directorios anteriormente creada, indique qué comandos son necesarios para realizar las siguientes acciones:

a. Mueva el archivo "f3" al directorio de trabajo /home/usuario.

```
mv f3 $HOME
```

b. Copie el archivo "f4" en el directorio "dir11".

```
cp -r f4 $HOME/dir1/dir11
```

c. Haga lo mismo que en el inciso anterior pero el archivo de destino, se debe llamar "f7".

```
cp -r f4 $HOME/dir1/dir11/f7
```

d. Cree el directorio copia dentro del directorio usuario y copie en él, el contenido de "dir1".

```
mkdir copia ; cp -a dir1 copia
```

e. Renombre el archivo "f1" por el nombre archivo y vea los permisos del mismo.

```
mv f1 archivo ; ls -ld archivo
```

f. 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**

```
chmod 617 archivo
```

g. Renombre los archivos "f3" y "f4" de manera que se llamen "f3.exe" y "f4.exe" respectivamente.

```
mv f3 f3.exe ; mv f4 f4.exe
```

h. 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**

```
chmod 023 f3.exe f4.exe
```


25. 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.

```
mkdir /tmp/logs
```

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

```
cp -a /var/log* /tmp/logs
```

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

```
tar -cvf misLogs.tar -C /tmp logs
```

 Con -C te paras en /tmp

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

```
tar -cvfz misLogs.tar.gz -C /tmp logs
```

e. Copie los archivos creados en c) y d) al directorio de trabajo de su usuario.

```
cp misLogs.tar mislogs.tar.gz ~/
```

f. Elimine el directorio creado en a), logs.

```
rm -r /tmp/logs
```

g. Desempaquete los archivos creados en c y d en 2 directorios diferentes.

Creo 2 directorios para desempaquetar uno en cada uno:

```
mkdir ~/logs_c
```

```
mkdir ~/logs_d
```

Desempaquete:

```
tar -xvf mislogs.tar -C ~/logs_c
```

```
tar -xvfz mislogs.tar.gz -C ~/logs_d
```