

## 1. Editor de textos:

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

A. Vim, Vi 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.

B. Cat: Muestra un archivo sin paginar.

Les: Muestra un archivo paginado. Pulsando "/" y escribiendo una palabra, puedo buscar las coincidencias de la misma en el archivo. Con la tecla "n" me muevo entre coincidencias hacia adelante, y con shift + "n" me muevo entre coincidencias hacia atrás. Con espacio cambio de página. Salgo con la "q".

More: te permite mostrar el resultado de la ejecución de un comando en la terminal de a una página a la vez. Esto es especialmente útil cuando se ejecuta un comando que causa un gran desplazamiento, como el comando ls o el comando du.

Existen tres modos o estados de vi:

- Modo comando: este es el modo en el que se encuentra el editor cada vez que se inicia. Las teclas ejecutan acciones (comandos) que permiten mover el cursor, ejecutar comandos de edición de texto, salir de vi, guardar cambios, etc.
- Modo inserción o texto: este es el modo que se usa para insertar el texto. Existen varios comandos que se pueden utilizar para ingresar a este modo.
- Modo línea o ex: se escriben comandos en la última línea al final de la pantalla

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

C.**dd**: Borra una línea entera.

Puede insertar texto (pasar a modo edición) con varias teclas:

**i**: Inserta texto antes del carácter sobre el que está el cursor.

**a**: Inserta texto después del carácter sobre el que está el cursor.

**I**: Inserta texto al comienzo de la línea en la que está el cursor.

**A:** Inserta texto al final de la línea en la que está el cursor.

**o:** Abre espacio para una nueva línea después de la línea en la que está el cursor y permite insertar texto en la nueva línea.

**O:** Análogo al anterior, pero abre espacio en la línea anterior.

**ESC:** Abandonar el modo de inclusión de texto para volver al modo de comandos; también se usa para cancelar comandos. (Usarlo en caso de duda)

**Cntl-F:** Avanzar una página hacia adelante.

**Cntl-B:** Avanzar una página hacia atrás.

**Cntl-L:** Refrescar la pantalla.

**G:** Cursor al final del fichero.

**1G:** Cursor al principio del fichero.

**\$:** Cursor al final de la línea.

**0 (cero):** Cursor al principio de la línea.

Para pasar de modo edición a modo de comandos se emplea la tecla ESC, para desplazarse sobre el archivo se emplean las teclas **j**(abajo), **k** (arriba), **h** (izquierda) y **l** (derecha).

## 2. Proceso de Arranque SystemV :

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

1 Se empieza a ejecutar el código del BIOS

2 El BIOS ejecuta el POST

3 El BIOS lee el sector de arranque (MBR)

4 Se carga el gestor de arranque (MBC)

5 El bootloader carga el kernel y el initrd

6 Se monta el initrd como sistema de archivos raíz y se inicializan componentes esenciales (ej.: scheduler)

7 El Kernel ejecuta el proceso init y se desmonta el initrd

8 Se lee el /etc/inittab

9 Se ejecutan los scripts apuntados por el runlevel 1

10 El final del runlevel 1 le indica que vaya al runlevel por defecto

11 Se ejecutan los scripts apuntados por el runlevel por defecto

12 El sistema esta listo para usarse

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

B. Lo ejecuta el kernel, su función es cargar todos los subprocesos necesarios para el correcto funcionamiento del SO

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

C. El programa pstree facilita información sobre la finalización de una serie de procesos relacionados entre sí, esto es, todos los descendientes de un proceso particular. El programa deja claro desde un principio que proceso es el primario y cuales son los secundarios.

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

D. El término runlevel o nivel de ejecución se refiere al modo de operación en los sistemas operativos que implementan el estilo de sistema de arranque de iniciación tipo UNIX System V.

- Es el modo en que arranca Linux (3 en Redhat, 2 en Debian)
- El proceso de arranque lo dividimos en niveles
- Cada uno es responsable de levantar (iniciar) o bajar (parar) una serie de servicios

(e) ¿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?

E. 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

Se encuentran definidos en /etc/inittab id:nivelesEjecucion:acción:proceso.

Los scripts que se ejecutan están en /etc/init.d.

No todas las distribuciones las respetan

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

F. En `inittab` se define que runlevel a ejecutar. En el archivo `inittab` se guarda la configuración del proceso `init` con respecto a que runlevels ejecutar.

Su formato es:

**Id:runlevels:action:process**

Niveles\_ejecucion: el/los nivel de ejecución en los que se realiza la acción.

Acción: describe la acción a realizar:

Wait: Se inicia cuando se entra al runlevel e `init` espera a que termine `initdefault`

Ctrlaltdel: se ejecutará cuando `init` reciba la señal `SIGINT`  
off, repawn, once, boot, bootwait, powerwait, otras...

Proceso: el proceso exacto que será ejecutado

Niveles ejecucion: el/los nivel de ejecución en los que se realiza la acción.

Acción: describe la acción a realizar:

Wait: Se inicia cuando se entra al runlevel e `init` espera a que termine `initdefault`

Ctrlaltdel: se ejecutará cuando `init` reciba la señal `SIGINT`  
off, repawn, once, boot, bootwait, powerwait, otras...

Proceso: el proceso exacto que será ejecutado.

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

G. En la línea de comandos se permite cambiar de un nivel de ejecución a otro con la herramienta *init*. Para cambiar de nivel de ejecución sólo hay que ejecutar *init* seguido del número del runlevel. Por ejemplo:  
`init 0` : cambia el runlevel 0 o sea se apaga el sistema (como el comando `halt`).  
Ni idea si es permanente.

(h) 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

H. Los scripts RC se guardan en los directorios `etc/rcX.d/` donde X es un runlevel, son enlaces a las funciones que crean los servicios que están en `etc/init.d/`. Los nombres en estos directorios empiezan con

una letra (S o K) seguidos de un número (de 2 dígitos) y el nombre del servicio.

(i) ¿Qué es *insserv*? ¿Para qué se utiliza? ¿Qué ventajas provee respecto de un arranque tradicional?

- I. ***insserv*** analiza automáticamente los datos introducidos en la cabecera del script *init* y guarda los enlaces para los scripts de arranque y parada en los directorios de niveles de ejecución respectivos.

Se utiliza para administrar el orden de los enlaces simbólicos del */etc/rcX.d*, resolviendo las dependencias de forma automática

Sus ventajas son que es más rápido que un arranque tradicional. La efectividad de esta técnica depende del número de servicios a iniciar, así como de la red de dependencia que haya entre ellos. Cuantos más servicios independientes haya, más se acelerará el arranque (más servicios podrán ser arrancados en paralelo).

(j) ¿Cómo maneja Upstart el proceso de arranque del sistema?

- J. Upstart es un sistema de arranque compatible System five pero que ejecuta las tareas de forma asincrónica permitiendo así una mejora en las prestaciones, las tareas y servicios son ejecutados ante eventos (arranque del equipo o inserción de un dispositivo USB) definidos como tareas o Jobs.

En distribuciones como Fedora o Ubuntu se usa Upstart en vez de System V.

(k) Cite las principales diferencias entre SystemV y Upstart.

- K. *Upstart* ofrece funciones compatibles con *SysV*, no obstante también cuenta con sus propios scripts y difiere en algunos aspectos importantes, en concreto *Upstart* ignora el archivo */etc/inittab*, en su lugar proporciona un conjunto integrado de scripts de arranque que en principio pueden reemplazar totalmente a */etc/inittab* y a los scripts de arranque específico del modo de ejecución de

(l) ¿Qué reemplaza a los scripts *rc* de SystemV en Upstart? ¿En qué ubicación del filesystem se encuentran?

- I. *SysV*, que en vez de encontrarse bajo directorios como */etc/init.d/rc?.d*, */etc/rc.d/rc?.d* o */etc/rc?.d* se encuentran en el directorio ***/etc/init***, con nombre '**servicio**'.**conf**, donde *servicio* es el programa que *init* tratará como un job. Los scripts de *Upstart* ofrecen más acciones que los de *SysV*,

(n) ¿Qué es *systemd*?

N. *systemd* es un conjunto de demonios o 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

(ñ) ¿A qué hace referencia el concepto de activación de socket en systemd?

- Ñ. • No todos los servicios que se inician en el booteo se utilizan:
- impresora
  - servidor en el puerto 80
  - etc.
- Es un mecanismo de iniciación bajo demanda → podemos ofrecer una variedad de servicios sin que realmente estén iniciados
- Cuando el socket recibe una conexión spawna el servicio y le pasa el socket
- No hay necesidad de definir dependencias entre servicios → se inician todos los sockets en primer medida

(o) ¿A qué hace referencia el concepto de cgroup?

- O. • Permite organizar un grupo de procesos en forma jerárquica
- Agrupa conjunto de procesos relacionados (por ejemplo, un servidor web Apache con sus dependientes)
- Tareas que realiza:
- Tracking mediante subsistema cgroups → no se utiliza el PID → doble fork no funciona para escapar de systemd
  - Limitar el uso de recursos
  - etc.

### 3. Usuarios:

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

A. Se usan:

-etc/password, para guardar el UID, el GID su directorio de usuario su nombre, etc.

Tiene el siguiente formato:

nombre\_de\_cuenta : contraseña : numero\_de\_usuario : numero\_de\_grupo :  
comentario : directorio : programa\_de\_inicio

-etc/group para guardar los datos del grupo como su GID sus miembros. Su formato es nombre\_de\_grupo : campo\_especial : numero\_de\_grupo: miembro1, miembro2

Con frecuencia, el campo especial está vacío. El número de grupo corresponde al número del vínculo entre los archivos /etc/group y los archivos /etc/passwd

-etc/shadow para guardar la contraseña de forma encriptada, junto con la fecha de expiración.

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

B. El UID es el identificador único de usuario el GID es el identificador único del grupo , en versiones anteriores podía haber varios usuarios con el mismo UID , luego esto fue cambiado que no de no hacerlo se podía modificar el UID de un usuario a 0 y así este tendrá los mismos permisos que el root.

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

C. root: es el administrador del sistema (superusuario)  
No puede existir mas de un usuario root  
La uid es 0

(e) Investigue la funcionalidad y parámetros de los siguientes comandos: useradd ó usermod userdel su groupadd who groupdel passwd

E. adduser : agrega el usuario, modifica el archivo etc/passwd.(va a pedir todos los datos)

passwd <nombre usuario> asigna o cambia la contraseña, modifica los archivos etc/shadow

usermod<nombre usuario>

- g modifica el grupo inicial modifica etc/passwd
- G modifica grupos adicionales etc/group
- d modifica el directorio home del usuario, modif etc/password

userdel<nombre usuario> elimina el usuario

groupdel<nombre del grupo>elimina el grupo

groupadd<nombre del grupo>crea un nuevo grupo

su<nombre de usuario>cambia al usuario nombre de usuario antes pidiendo su contraseña, si no se pone nombre de usuario se establece que se quiere cambiar a ser super usuario.

who muestra quien esta logueado.

#### 4. FileSystem:

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

- Existen 3 tipos de permisos y se basan en una notación octal:

Permiso	Valor	Octal
Lectura	R	4
Escritura	W	2
Ejecución	X	1

A.

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

B. Chmod: Sirve para gestionar los permisos de los archivos o directorios del sistema. Cuando usamos este comando debemos tener presentes los tres niveles de gestión de permisos que existen: Lectura, Escritura y Ejecución.

Chown: El comando chown ("change owner", cambiar propietario en inglés) permite cambiar el propietario de un archivo o directorio en sistemas tipo UNIX. Puede especificarse tanto el nombre de un usuario, así como el identificador de usuario (UID) y el identificador de grupo (GID).

Chgrp: El comando chgrp permite cambiar el grupo de usuarios de un archivo o directorio en sistemas tipo UNIX. Cada archivo de Unix tiene un identificador de usuario (UID) y un identificador de grupo (GID), que se corresponden con el usuario y el grupo de quien lo creó.

(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?

• Existen 3 tipos de permisos y se basan en una notación octal:

Permiso	Valor	Octal
Lectura	R	4
Escritura	W	2
Ejecución	X	1

C.

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

D. No, pero en caso de que seas un usuario ROOT podrias hacerlo tranquilamente.

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

E. Full path name: Una ruta de acceso completa o una ruta absoluta es una ruta que apunta a la misma ubicación en un sistema de archivos independientemente del directorio de trabajo actual o rutas combinadas.

Relative path name: El nombre de la ruta relativa enumera solo aquellos directorios que son relativos al directorio actual.

(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.

F. PWD.

Si existe una forma y es escribiendo la ruta relativa.

Obvio.

Por ejemplo, si su directorio actual es `/usr/bin` y su directorio de destino es `/usr/bin/reports`, escriba el nombre de la ruta relativa `informes` (sin la barra inclinada inicial).

(g) Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con el uso del FileSystem: `-cd -umount -mkdir -du -rmdir -df -mount -ln -ls -pwd -cp -mv`

G. CD: El comando `cd` (change directory) permite moverse entre directorios del sistema. Puedes cambiar de directorio especificando la ruta absoluta desde el directorio raíz o relativa desde tu ubicación actual, en Linux el directorio actual se indica con el signo.

Umount: El comando `umount` le permite eliminar un sistema de archivos remoto que esté montando en la actualidad. El comando `umount` se puede utilizar con las opciones siguientes: `-V`. Permite realizar pruebas.

Mkdir: `mkdir` es una orden de los sistemas operativos UNIX, DOS, OS/2 y Microsoft Windows usada para crear un nuevo subdirectorio o carpeta del sistema de archivos.

Du: El comando `du`, de manera similar al anterior, nos expresa el uso en disco que realiza un conjunto de archivos y de manera recursiva en los directorios existentes. El comando se desplazará a través de todos los archivos y carpetas dentro del presente directorio de trabajo.

Rmdir: Elimina los directorios sólo si están vacíos.

df: se usa para chequear el espacio en el disco. Mostrará el almacenamiento disponible y utilizado de los sistemas de archivos en tu máquina.

mount: `mount` es un comando de sistemas basados en Unix que se utiliza para montar dispositivos y particiones para su uso por el sistema operativo.

ln: Con el comando `ln` de Linux (`ln` es la abreviatura de "link", enlace en inglés), se consigue crear enlaces a un archivo o carpeta.

ls: El comando `ls` se utiliza para listar archivos o directorios en Linux y otros sistemas operativos basados en Unix.

pwd: El comando `pwd` (de las siglas en inglés print working directory, cuya traducción sería imprimir directorio de trabajo) se utiliza para imprimir el nombre del directorio actual en una sesión de comandos bajo un sistema operativo Unix o derivado.

cp: El comando cp se emplea para hacer copias de archivos y directorios en nuestro sistema operativo.

mv: es usado para mover o renombrar archivos o directorios del sistema de archivos.

## 5. Procesos:

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

A. Un proceso es un programa en ejecución . PID es el identificador único del proceso, PPid es el identificador único del proceso padre. Todos los procesos en gnu Linux tienen estos atributos:

p o PID Process ID, número único o de identificación del proceso. P o PPID Parent Process ID, padre del proceso U o UID User ID, usuario propietario del proceso t o TT o TTY Terminal asociada al proceso, si no hay terminal aparece entonces un '?'

T o TIME Tiempo de uso de cpu acumulado por el proceso

c o CMD El nombre del programa o camndo que inició el proceso  
RSS Resident Sise, tamaño de la parte residente en memoria en kilobytes

SZ o SIZE Tamaño virtual de la imagen del proceso

NI Nice, valor nice (prioridad) del proceso, un número positivo significa menos tiempo de procesador y negativo más tiempo (-19 a 19)

C o PCPU Porcentaje de cpu utilizado por el proceso

STIME Starting Time, hora de inicio del proceso

S o STAT Status del proceso, estos pueden ser los siguientes

Process ID, número único o de identificación del proceso.

Parent Process ID, padre del proceso

User ID, usuario propietario del proceso

Terminal asociada al proceso, si no hay terminal aparece entonces un '?'

Tiempo de uso de cpu acumulado por el proceso

El nombre del programa o camndo que inició el proceso

Resident Sise, tamaño de la parte residente en memoria en kilobytes

Tamaño virtual de la imagen del proceso

Nice, valor nice (prioridad) del proceso, un número positivo significa menos tiempo de procesador y negativo más tiempo (-19 a 19)

Porcentaje de cpu utilizado por el proceso

Starting Time, hora de inicio del proceso

Status del proceso, estos pueden ser los siguientes

- R runnable, en ejecución, corriendo o ejecutándose
- S sleeping, proceso en ejecución pero sin actividad por el momento, o esperando por algún evento para continuar
- T sTopped, proceso detenido totalmente, pero puede ser reiniciado
- Z zombie, difunto, proceso que por alguna razón no terminó de manera correcta, no debe haber procesos zombies
- D uninterruptible sleep, son procesos generalmente asociados a acciones de IO del sistema
- X dead, muerto, proceso terminado pero que sigue apareciendo, igual que los Z no deberían verse nunca

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

B. Se puede usar ps que liste los procesos que se están ejecutando .

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

C. Cuando un proceso está en ejecución sin que sea mostrado en la terminal se dice que se está ejecutando en el background (segundo plano). Si se muestra la ejecución del comando dentro de la terminal se dice que está en el foreground (primer plano).

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

D. Para llevar un proceso de foreground a background se utiliza el comando `bg`, se utiliza usando `bg %326`. Para llevar un proceso de background a foreground se utiliza el comando `fg`, de la misma manera que el `bg`

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

E. El `|` nos permite comunicar dos procesos por medio de un pipe o tubería desde la shell, el pipe conecta `stdout` (salida estándar) del primer comando con la `stdin` (entrada estándar) del segundo. Ejemplo `head -5 archivo | tail -1` permite quedarse con la 4 línea de archivo.

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

F. La redirección tiene como objetivo poder concatenar las funciones de varios procesos consiguiendo un resultado en conjunto

Símbolo	Descripción
<code>&gt;</code>	Redirecciona <b>stdout</b> hacia un archivo. Lo crea si no existe, si existe lo sobrescribe. <code>ls -l &gt; lista.txt</code> (La salida del comando se envía a un archivo en vez de la terminal.)
<code>&gt;&gt;</code>	Redirecciona <b>stdout</b> hacia un archivo. Lo crea si no existe, si existe concatena la salida al final de este. <code>ps -ef &gt;&gt; procesos.txt</code> (Concatena al archivo <code>procesos.txt</code> la salida del comando.)
<code>&lt;</code>	Redirecciona <b>stdin</b> desde un archivo. El contenido de un archivo es la entrada o input del comando. <code>mail user &lt; texto.txt</code> (El cuerpo del correo a enviar proviene desde un archivo, en vez del teclado.)
<code>2&gt;</code> <code>2&gt;&gt;</code>	Redirecciona <b>stderr</b> hacia un archivo. Crea ( <code>&gt;</code> ) o concatena ( <code>&gt;&gt;</code> ) la salida de errores a un archivo. (ver ejemplos)
<code>1&gt;&amp;2</code>	Redirecciona <b>stdout</b> hacia donde <b>stderr</b> apunte. (ver ejemplos)
<code>2&gt;&amp;1</code>	Redirecciona <b>stderr</b> hacia donde <b>stdout</b> apunte. (ver ejemplos)
OTROS REDIRECCIONAMIENTOS QUE NO UTILIZAN FDs	
<code>&lt;&lt;</code>	Conocido como <b>HERE-DOCUMENT</b> o <b>HereDoc</b> (ver ejemplos)
<code>&lt;&lt;&lt;</code>	Conocido como <b>HERE-STRING</b> (ver ejemplos)
<code> </code>	El símbolo <code> </code> (pipe) es un tipo de redireccionamiento ya que la salida ( <b>stdout</b> ) de un comando es la entrada ( <b>stdin</b> ) de otro. <code>ls /etc   grep services</code> (La salida del comando a la izquierda de <code> </code> se convierte en la entrada del comando a la derecha.)
<code>tee</code>	El comando <code>tee</code> redirecciona la salida ( <b>stdout</b> ) a ambos, un archivo y a la terminal. <code>ps -ef   tee procesos.txt</code> (La salida de <code>ps</code> se muestra en la terminal y al mismo tiempo se redirecciona al archivo <code>procesos.txt</code> . Con la opción <code>-a</code> ( <code>tee -a</code> ) concatena al archivo.)

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

G. El comando kill (un Proceso) envía una señal a un proceso que por lo general implica que este pare, o se termine de ejecutar, según el contexto.

Es de la forma kill –opcion –pid.

Con la opción -9 se fuerza la terminación.

Con -19 o -20 se detiene momentáneamente

Con -18 se continua la ejecución,

Con -1 se obliga a releer los archivos de configuración.

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

H. ps: despliega la información de los procesos en ejecución.

pstree: lo mismo pero en forma de árbol.

top: Da informacion en tiempo real de las tareas que se estan ejecutando mostrando el % de cpu y memoria que se esa usando, sus IDs, usuarios que lo estan ejecutando,etc

kill y killall: Los comandos kill y killall permiten enviar señales a los procesos. El comando kill espera un número de proceso como argumento, mientras que killall espera un nombre de comando.

nice: El comando nice en Linux nos permite modificar la prioridad de un proceso frente al resto dentro del sistema.

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

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

A. Hace referencia a los archivos TAR, son archivos empaquetados.

**Empaquetar:** Es agrupar en un solo fichero varios ficheros y/o directorios.

En linux contamos con el comando `tar`, que nos permite realizar el proceso de empaquetación, su sintaxis es la siguiente:

```
tar [opciones] [nombre_fichero_tar]
[directorio_origen]
```

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

```
ignacio@ignacio-VirtualBox: ~/Desktop/Escritorio
ignacio@ignacio-VirtualBox:~/Desktop/Escritorio$ tar -cvf archivo.tar 'iso 2017'
iso 2017/
iso 2017/iso2017
iso 2017/archivo.tar
iso 2017/iso2017-2
iso 2017/sumando1
iso 2017/sumando2
ignacio@ignacio-VirtualBox:~/Desktop/Escritorio$ ls
archivo.tar  'iso 2017'  prueba2.exe  prueba.exe
ignacio@ignacio-VirtualBox:~/Desktop/Escritorio$ ls -l 'archivo.tar'
-rw-rw-r-- 1 ignacio ignacio 20480 sep  7 16:16 archivo.tar
ignacio@ignacio-VirtualBox:~/Desktop/Escritorio$ la -l 'iso 2017'
total 16
-rw-rw-r-- 1 ignacio ignacio 10240 sep  7 16:16 archivo.tar
-rw-rw-r-- 1 ignacio ignacio    10 ago 25 17:53 iso2017
-rw-rw-r-- 1 ignacio ignacio     0 ago 25 17:45 iso2017-2
-rw-rw-r-- 1 ignacio ignacio     0 sep  7 16:09 sumando1
-rw-rw-r-- 1 ignacio ignacio     0 sep  7 16:09 sumando2
ignacio@ignacio-VirtualBox:~/Desktop/Escritorio$ man ls
ignacio@ignacio-VirtualBox:~/Desktop/Escritorio$ ls -s
total 32
20 archivo.tar  4 'iso 2017'  4 prueba2.exe  4 prueba.exe
B. ignacio@ignacio-VirtualBox:~/Desktop/Escritorio$
```

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

C. Posicionarse un nivel antes de la carpeta a comprimir, luego utilizamos el comando `tar -cvf [nombre que le pones al archivo tar] [carpeta a comprimir]`

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

D. No.

(e) Investigue la funcionalidad de los siguientes comandos: `-tar -grep -gzip -zgrep -wc`

- E. Tar: El comando TAR es usado para comprimir o empaquetar un archivo o directorio.

Grep: busca las líneas que contengan un determinado patrón que se pasa como parámetro, por defecto imprime estas líneas.

Es de la forma `grep [opcion] patrón [fichero]`

Tiene varias opciones:

`grep -E` interpreta el patrón como una expresión regular extendida.

`grep -F` interpreta el patrón como una lista de strings, separadas por nuevas líneas, una coincidencia con cualquier línea, da como resultado un acierto.

`gzip`: El comando Gzip crea un fichero comprimido que finaliza con `.gz`

`zgrep`: invoca a `grep` para archivos comprimidos.

`wc`: imprime datos de los archivos como las líneas las palabras y el conteo de bytes para cada archivo, además de un total si especifica más de uno.

7. 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` // Se puede

`ps > PRUEBA` // Se puede

`chmod 710 prueba` // Se puede

`chown root : root PRUEBA` //

`chmod 777 PRUEBA` // Se puede

`chmod 700 / etc / passwd // No se puede ya que`

`passwd root rm PRUEBA`

`man / etc / shadow`

`find / -name * . conf`

`usermod root -d /home/ newroot -L`

`cd / root`

`rm *`

`cd /etc`

`cp * /home -R`

`shutdown`

a) `ls-l>` prueba: lista los permisos del archivo y otros datos en el archivo prueba

b) `ps>` prueba : crea un archivo llamado PRUEBA y guarda ahí los datos de los procesos en ejecución.

c) `chmod 710 prueba`: cambia los permisos de prueba, quedando todo para el propietario, ejecución para el grupo, nada para otros.

d) `chown root`: root Prueba intenta cambiar el propietario de PRUEBA al root, pero es no se puede ya que no se está invocando desde un root y solo éste puede utilizar `chown`.

e) `chmod 777 PRUEBA` cambia los permisos de PRUEBA haciendo que todos puedan leer ejecutar y escribir.

f) `chmod 700 etc/passwd` , no lo permite ya que no es el root y no puede dejar a otros usuarios sin permisos para lo que no es su directorio.

g) `passwd root` no me deja manipular la contraseña porque no soy el root.

h) `rm PRUEBA` elimina el archivo prueba.

i) `man /etc/shadow`: intenta abrir shadow (su manual) pero no lo permite porque es un archive del root.



- j) **find / -name \*.conf**. Busca todos los archivos en / con terminacion ".conf" y no encuentra ninguno.(entre los que tenga permiso)
- k)usermod root-d /home/newroot -L no encuentra la orden
- l) cd /root quiere ir al directorio /root permiso denegado.
- m) **rm \***: remueve todos los archivos pero no los directorios.
- n)cd /etc: va al directorio /etc
- o) cp \* /home -R home-R no es un directorio à intenta copiar todos los archivos de configuración, pero tiene El acceso denegado a todos ellos.
- p)shutdown no hace nada no encuentra la orden.

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

(a) Terminar el proceso con PID 23.

A. KillAll

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

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

C. Find o grep.

(e) Cambiar los permisos del archivo /home//xxxx a:

E. Chmod 111

Chmod 101

Chmod 001

(f) Guardar una lista de procesos en ejecución el archivo /home//procesos

D. Chmod 110

Chmod 101

Chmod 000

(g) Borrar todos los archivos del directorio /tmp

G. rm \*

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

H. Chown

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

i. pwd > /home/ignacio/donde

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

A Ingrese al sistema como usuario " root "

sudo -l

B. Cree un usuario. Elija como nombre, por convención, la primera letra de su nombre seguida de su apellido . Asigne una contraseña de acceso .

useradd s

passwd s

C. ¿ Qué archivos fueron modificados luego de crear el usuario y qué directorios se crearon ?

Fue modificado / etc / group and / etc / passwd / home / user / s fue creado . No sé qué comandos

usar para encontrar esa información

D. Crear un directorio en / tmp llamado cursada2022

```
mkdir / tmp / cursada2022
```

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

```
cp r / var / log / tmp / cursada2022
```

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 users / tmp / cursada2922
```

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

```
chmod 723 / tmp / cursada2822
```

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

```
su -s.
```

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

J . Verifique la cantidad de procesos actions que hay en el sistema

```
ps r
```

K. Verifiqué la cantidad de usuarios conectados al sistema

```
who -q
```

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

```
sudo
```

```
write s < " Se va a apagar el sistema "
```

M . Apague el sistema .

poweroff

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

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

mkdir / home / 185443

cd / home / 185443

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

touch leame.txt

vi leame.txt

-i Sofia Avila

-1 18544/3

-i avilasofia076@gmail.com

: W

: q

C . Cambie los permisos del archivo LEAME , de manera que se puedan ver reflejados los siguientes permisos :

▪ Dueño : ningún permiso

▪ Grupo : permiso de ejecución

■ Otros : todos los permiso

```
chmod 017 leame.txt
```

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

```
cd / etc
```

```
touch leame
```

```
ls > leame
```

Es posible hacerlo porque Linux es case sensitive , quiere decir que distingue las mayúsculas de las minúsculas

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

```
find / -iname < nombre del archivo >
```

```
find / -iname < * algo en común * >
```

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

```
find / -iname " * .so " > | tee / home / 185443 / ejercicio_f
```

11. 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 :

I . mkdir iso

II . cd . / iso ; ps > fo

III . ls > f1

IV . cd /

V. echo \$ HOME

VI. 1s 1 & > \$ HOME / iso / 1s

VII. cd \$ HOME ; mkdir f2

VIII. ls -ld f2

IX. chmod 341 f2

X. touch dir

XI.. cd f2

XII. cd / iso

XIII. pwd > f3

XIV. ps | grep ' ps ' | wc -1 >> .. / f2 / f3

XV. chmod 700 .. / f 2 ; cd ..

XVI. find -name etc / passwd

XVII. find - name etc / passwd

XVIII. mkdir ejercicios

I. Crea la carpeta iso

II. Accede a la carpeta / iso y lista los procesos ejecutándose guardándolos en fo

III.Lista el directorio / iso y lo guarda en f1

IV. Me posiciono en /

V. Imprime el directorio home del usuario

VI. Guarda el listado detallado del directorio actual en el archivo llamado ls contenido en el directorio iso

VII. Me posiciono en home y crea el directorio f2

VIII. Lista los detalles de f2

ix . Cambia los permisos de f2 . El propietario podrá escribir y ejecutar . El grupo puede leerlo y el resto puede ejecutarlo

X .Crea el archivo dir

xi .Me posiciono en f2

xii .Vuelve a / home / user / iso

xiii. Guarda en f3 la posición actual en la que se encuentra

xiv .Guarda en f3 sin sobreescritura ubicado en el directorio f3 la cantidad de líneas que contienen la palabra " ps " entre la lista de procesos

A . Inicie 2 sesiones utilizando su nombre de usuario y contraseña . En una sesión vaya siguiendo paso a paso las órdenes que se encuentran escritas en el cuadro superior . En la otra sesión , cree utilizando algún editor de textos un archivo que se llame .ejercicio10\_explicacion " dentro del directorio creado en el ejercicio 9.a ) y , para cada

una de las órdenes que ejecute en la otra sesión , realice una breve explicación de los resultados obtenidos .

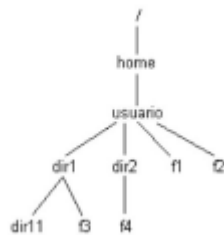
B . Complete en el cuadro superior los comandos 19 y 20 , de manera tal que realicen la siguiente acción :

19 : Copiar el directorio iso y todo su contenido al directorio creado en el inciso 9.a ) .

20 : Copiar el resto de los archivos y directorios que se crearon en este ejercicio al directorio creado en el ejercicio 9.a )

C . Ejecute las ordenes 19 y 20 y coméntelas en el archivo creado en el inciso a )

12. 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 :



cd / home / sofia

mkdir dir1

mkdir dir2

mkdir dir1 / dir11

touch dir1 / f3

touch dir2 / f4

touch f1



touch f2

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

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

```
cd dir1
```

```
mv -v f3 / home / sofia
```

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

```
cd / home / dir2
```

```
cp f4 / home / sofia / dir1 / dir11
```

. Haga los mismo que en el inciso anterior pero el archivo de destino , se debe llamar "F7"

```
cd / home / dir2
```

```
cp f4 / home / sofia / dir1 / dir11 / f7
```

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

```
" dir1 " .
```

```
mkdir copia
```

```
cp dir1 / home / sofia / copia
```

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

```
cd / home / sofia
```

`mv f1 archivo`

`ls -l archivo`

. 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`

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

`mv f3 f3.exe`

`cd / home / sofia / dir1 / dir11`

`mv f4 f4.exe`

. 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

`cd / home / sofia / copia`

`chmod 026 *`

13. 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 .

```
cd / tmp
```

```
mkdir logs
```

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

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

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

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

D . Empaquete y comprima el directorio creado en 1 , el archivo resultante se debe llamar

```
" misLogs.tar.gz " .
```

```
tar -cvzf misLogs.tar.gz / tmp / logs
```

E . Copie los archivos creados en 3 y 4 al directorio de trabajo de su usuario

```
cp misLogs.tar / home / sofia / primero
```

```
cp misLogs.tar.gz / home / sofia / segundo
```

F . Elimina el directorio creado en 1 , logs

```
cd / tmp
```

```
rmdir logs
```

G . Desempaquete los archivos creados en 3 y 4 en 2 directorios diferentes

```
tar -xvf misLogs.tar > / home / sofia / des1
```

```
tar -xvf misLogs.tar.gz > / home / sofia / des2
```

