

run 1. Editor de textos:

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

A-Neovim, Gedit,FeatherPad.

(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- Que los comandos son justamente comandos, no editores de texto, estos sirven para crear scripts y corren en cualquiera de los editores previamente mencionados en el A.

Cat : el comando cat imprimirá el contenido de uno o varios archivos sin ningún tipo de paginación o forma de modificarlo.

*mostrar el contenido de un fichero: “cat fichero”

*mostrar el contenido de varios archivos en uno:

“cat archivo1 archivo2 > archivo3”

More: al igual que cat permite visualizar el contenido del archivo, con la diferencia que este pagina los resultados.

Primero mostrará por pantalla todo lo que se pueda visualizar sin hacer scroll y después, pulsando la tecla espacio avanzará de igual modo por el archivo.

Less:El comando ‘less’ es el más completo de los tres, pues puede hacer todo lo que hace ‘more’ añadiendo mayor capacidad de navegación por el fichero (avanzar y retroceder) además de que sus comandos están basados en el editor ‘vi’, del cual se diferencia en que no tiene que leer todo el contenido del fichero antes de ser abierto.

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

- vi: abre el programa sin ningún archivo.
- vi fichero: edita el fichero si existe, y si no, lo crea.
- vi fichero1 fichero2: edita varios archivos.
 - :next o :n para pasar al siguiente archivo.
 - :prev o :N para ir al archivo previo.
- vi + [número fichero] : edita el fichero iniciando en la línea indicada.
- vi+/patrón fichero: edita el fichero iniciando en la primera vez que encuentre el patrón

Demás comandos:

- i para insertar texto a la izquierda del cursor.
- a para insertar texto a la derecha del cursor.
- I se encarga de insertar texto al inicio de la línea.
- A es el encargado de insertar texto al final de la línea.
- ESC vuelve al modo comando.
- X es el comando que borra el carácter bajo el cursor.
- dd para borrar la línea actual.
- dw borra la palabra actual.
- h o flecha izquierda moverá el cursor hacia la izquierda.
- j o flecha abajo moverá el cursor a la línea de abajo.
- k o flecha arriba mueve el cursor a la línea de arriba.
- l o flecha derecha mueve el cursor a la derecha.
- :w guarda los cambios.
- :q para salir del editor.

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.

A-

1.El BIOS realiza las tareas de inicio específicas de la [plataforma de hardware](#).

2. Una vez que el [hardware](#) es reconocido y se inicia correctamente, el BIOS carga y ejecuta el código de la partición de arranque del dispositivo de arranque designado, que contiene la fase 1 de un gestor de arranque Linux. La fase 1 carga la fase 2 (la mayor parte del código del gestor de arranque). Algunos cargadores pueden utilizar una fase intermedia (conocida como la fase 1.5) para lograr esto, ya que los modernos discos de gran tamaño no pueden ser totalmente leídos sin código adicional.

3.El gestor de arranque a menudo presenta al usuario un menú de opciones posibles de arranque. A continuación, carga el sistema operativo, que descomprime en la memoria, y establece las funciones del sistema como del hardware esencial y la [paginación de memoria](#), antes de llamar a la función start_kernel().

4. La función start_kernel() a continuación realiza la mayor parte de la configuración del sistema ([interrupciones](#), el resto de la [gestión de memoria](#), la inicialización del dispositivo, controladores, etc), antes de continuar por separado el proceso inactivo y planificador, y el proceso de [Init](#) (que se ejecuta en el espacio de usuario).

5.El planificador toma control efectivo de la gestión del sistema, y el núcleo queda dormido (inactivo).

6. El proceso Init ejecuta secuencias de comandos ([Scripts](#)) necesarios para configurar todos los servicios y estructuras que no sean del sistema operativo, a fin de permitir que el entorno de usuario sea creado y pueda presentarse al usuario con una pantalla de [inicio de sesión](#)

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

B- El proceso init es el primer proceso en ejecución tras la carga del núcleo y el que a su vez genera todos los demás procesos, y el kernel es el encargado de ejecutarlo.

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

C- El comando Pstree facilita información sobre la finalización de una serie de procesos relacionados entre sí, esto sería como todos los descendientes de un proceso particular. El programa cuando se ejecuta ese comando, deja en claro quien es el primario y quienes son los secundarios

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

D- El término runlevel se refiere al modo de operación en los sistemas operativos que implementan el estilo de sistema de arranque de iniciación tipo UNIX System. En términos prácticos, cuando el computador entra al runlevel 0, está apagado, y cuando entra al runlevel 6, se reinicia.

(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-

Los 7 niveles de ejecución (runlevels) estándares

Nivel de ejecución	Nombre o denominación	Descripción
0	Alto	Alto o cierre del sistema (Apagado).
1	Modo de usuario único (Monousuario)	No configura la interfaz de red o los demonios de inicio, ni permite que ingresen otros usuarios que no sean el usuario root , sin contraseña. Este nivel de ejecución permite reparar problemas, o hacer pruebas en el sistema.
2	Multiusuario	Multiusuario sin soporte de red.
3	Multiusuario con soporte de red.	Inicia el sistema normalmente sin GUI .
4	Multiusuario con soporte de red.	No es igual que el 3.
5	Multiusuario gráfico (X11)	Similar al <i>nivel de ejecución 3</i> + display manager .
6	Reinicio	Se reinicia el sistema.

El [sistema operativo GNU/Linux](#) puede aprovechar los **niveles de ejecución** a través de los programas del proyecto [sysvinit](#). Después de

que el [núcleo Linux](#) ha [arrancado](#), el programa [init](#) lee el archivo /etc/inittab para determinar el comportamiento para cada **nivel de ejecución**. A no ser que el usuario especifique otro valor como un parámetro de [autoarranque](#) del [núcleo](#), el sistema intentará entrar (iniciar) al **nivel de ejecución** por defecto.

No todas respetan los niveles de ejecución, la mayoría de distribuciones de linux si.

(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- Este programa es el encargado de lanzar los scripts de inicialización del sistema y de modificar el sistema operativo de su estado inicial de arranque al estado estándar multiusuario.

La información contenida es ID: identifica la entrada en inittab

Niveles Ejecución: el/los niveles de ejecución en los que se realiza la acción

Acción: describe la acción a realizar

Wait: inicia cuando entra al runlevel e init espera a que termine Initdefault Ctrlaltdel. Se ejecutará cuando init reciba la señal SIGINT Off,respawn, once, sysinit, boot, boowait, power wait, etc

Proceso: el proceso exacto que será ejecutado

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

G- Depende, hay 2 formas de cambiar el runlevel, una que tenemos que darle al comando id:5:initdefault: que lo q hace es poner ese runlevel por defecto, en cambio podemos ejecutar por ejemplo "init 2" que cambia el runlevel en la ejecución.

(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 una orden para llamarlos? Justifique.

H- La finalidad del Scripts RC es: cuando el init entra en nivel de ejecución llama al script rc con un argumento numérico que especifica el nivel de ejecución al que ir, rc luego inicia y detiene los servicios en el sistema según sea necesario para llevar el sistema a ese nivel de ejecución. Aunque normalmente se llama al inicio, init puede llamar al script rc para cambiar a los niveles de ejecución

- Todos los scripts rc se encuentran bajo /etc/init. d.
- . Cuando init llama al programa /etc/rc para cambiar los niveles de ejecución, lo hace yendo al directorio para el nivel de ejecución de destino y trabaja a través de todos los archivos de script K## para detener los servicios en ejecución que no deberían estar en el nivel de ejecución de destino. y archivos S##script para servicios que deberían ejecutarse, pero que no se ejecutan actualmente, en el nivel de ejecución de destino Es importante que los scripts comiencen y terminen en el orden correcto. Por ejemplo, para que NFS y NIS funcionen correctamente, primero se debe iniciar el Daemon rpcbind del mapeador de puertos RPC

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

- Isserv es usado para administrar el orden de los enlaces simbólicos del /etc/rcx.d, resolviendo las dependencias de forma automática. Permite la ejecución de trabajos en forma asincrónica a través de eventos como principal diferencia con sysVinit que es estrictamente sincrónico, estos trabajos se denominan Jobs y su principal objetivo es definir servicios a ser ejecutadas por init. Son scripts de texto plano que define las acciones a ejecutar ante determinados eventos

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

- Upstart es un reemplazo basado en eventos para el daemon /sbin/init que maneja el inicio de tareas y servicios durante el arranque, los detiene durante el apagado y los supervisa mientras el sistema se está ejecutando.

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

- Upstart: es un reemplazo basado en eventos para el daemon /sbin/init
- SystemD: Es un nuevo sistema init y administrador de sistema que se adaptó/implementó en todas las principales distribuciones de linux sobre los sistemas tradicionales SysV.

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

(m) Dado el siguiente job de upstart perteneciente al servicio de base de datos del mysql

indique a qué hace referencia cada línea del mismo:

```
# MySQL Service
description "MySQL Server "
author "info autor"
start on (net-device -up
and local-file-systems
and runlevel[2 3 4 5])
stop on runlevel[0 1 6]
[...]
exec /usr/sbin/mysqld
[...]
```

(n) ¿Qué es systemd?

N- systemd es un conjunto de daemons de administración de sistema, bibliotecas y herramientas diseñados como plataforma de administración y configuración central para interactuar con el núcleo del SO gnu linux.

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

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

3. Usuarios:

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

A- ext2, ext3 y ext4.

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

UID: es el identificador de un usuario, mientras que GID es el identificador de un grupo de Usuarios.

Si, pueden existir dos usuarios con el mismo UID, ya que este no es más que un archivo de texto que sirve como identificador. Si bien se puede, no es recomendable porque puede traer consecuencias no deseadas.

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

El usuario root es un usuario con funciones de superusuario o “cuenta privilegiada”, esta siempre se caracteriza por tener el UID “0”. Linux te da la libertad de hacer lo que tu quieras con el sistema, es decir que si quieres tener dos usuarios con el UID 0 es posible, que puedas hacer algo no significa que esté bien o no traiga consecuencias.

(d) Agregue un nuevo usuario llamado iso 2017 a su instalación de GNU/Linux, especifique que su home sea creada en /home/iso 2017, y hágalo miembro del grupo cátedra (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.

D- sudo useradd -m -g cátedra -s /bin/bash usuario:

-m: Crear automáticamente la carpeta del usuario en la carpeta /Home/<NombreUsuario>

-g: grupo principal al que será agregado

sudo mkdir /Catedras/Clase1/"iso 2017"-> crea la carpeta

sudo chown usuario:usuario -R/Catedras/Clase1/"iso 2017" -> pone como propietario de la carpeta al usuario

sudo usermod -d /Catedras/Clase1/usuario iso 2017 -> con el -d aclaramos que le vamos a asignar Clase1 como su carpeta personal

sudo userdel -r filemon

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

useradd o adduser -> agregar un usuario

usermod -> sirve para modificar múltiples configuraciones del usuario, ya sea su path de directorios, su contraseña, a los grupos que pertenece etc.

userdel -> borra un usuario, aun es visible ciertas cosas del mismo como su directorio que sigue vigente

groupadd -> agrega un usuario a un grupo.

who -> indica el nivel de ejecución del sistema y que usuarios están conectados.

groupdel -> borra un grupo.

passwd -> se utiliza para modificar la contraseña del usuario que se pase como argumento, tmb puede borrar la contraseña entre otras cosas según el argumento que se le pase.

4. FileSystem:

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

A- Los permisos están divididos en tres tipos: lectura, escritura, y ejecución. Estos permisos pueden ser fijados para tres clases de usuarios: el propietario del archivo o directorio, los integrantes del grupo al que pertenece y todos los demás users.

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

chmod: El **CHMOD** (abreviatura de cambio de modo) permite cambiar los permisos de acceso de un archivo o de un directorio.

chown: Para cambiar el propietario de un archivo o directorio, en la mayoría de distribuciones **Linux** se deberá utilizar el comando CHOWN, pero esta acción, solo la podrá realizar el usuario root del sistema.

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?

C-

R: read: solo lectura = 4

W: write: solo escritura = 2

X: Execute: solo ejecución = 1

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

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

Full path name: un nombre de vía de acceso completo empieza en el directorio raíz y efectúa un rastreo de todos los directorios que quedan por debajo hasta llegar al archivo y directorio de destino. Por ejemplo /etc/uupc/Devices hace referencia al archivo Devices del directorio raíz, siempre es preciso especificar delante el carácter de barra.

Separe siempre los elementos de la vía de acceso mediante el carácter /
Relative path name: el nombre de vía de acceso relativo sólo contiene los directorios que dependen del directorio actual. Por ejemplo, si el directorio actual es /usr/bin y el directorio de destino es /sr/bin/reports, escribo el nombre de vía de acceso relativo reports

(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

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

cd **permite moverse entre directorios del sistema.**

umount el comando **umount** le permite eliminar un sistema de archivos remoto que esté montando en la actualidad

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 nos expresa el uso en disco que realiza un conjunto de archivos y de manera recursiva en los directorios existentes.

rmdir se usa para eliminar un directorio y sus subdirectorios.

df f, que significa Disk Filesystem **se usa para chequear el espacio en el disco.**

mount se utiliza para montar dispositivos y particiones para su uso por el sistema operativo.

ln es la abreviatura de link, se consigue crear enlaces a un archivo o carpeta. Los enlaces simbólicos son una referencia cruzada que se dirige a un archivo original mediante una ruta de referencia. Si se traslada o elimina el archivo original, deja de haber un destino y la indicación se queda vacía, pero si en cambio se elimina el enlace simbólico, el archivo original se mantiene. Los enlaces duros solo pueden generarse con la línea de comandos, tienen mayor alcance que los enlaces simbólicos en la administración de memoria de un sistema operativo y en el sistema de archivos integrado usado.

ls usado para listar archivos o directorios. Permite enumerar todos los archivos en el directorio actual de forma predeterminada e interactuar con ellos a través de la línea de comandos

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 un comando de Unix usado para mover o renombrar archivos o directorios del sistema de archivos

5. Procesos:

(a) ¿Qué es un proceso? ¿A qué hacen referencia las siglas PID y PPID?

¿Todos los procesos tienen estos atributos en GNU/Linux?

¿Qué procesos tienen estos atributos en GNU/Linux? Justifique. Indique qué otros atributos tiene un proceso.

A- Un proceso es un conjunto de instrucciones que vienen de un programa que se está ejecutando. PID es el identificador del proceso que se está ejecutando, mientras que ppid se relaciona con el pid del proceso padre de este. Ya que cada proceso invoca a varios procesos, estos son los padres de los procesos que invocan y su ppid está relacionado con el pid del padre.

Si, inclusive el init tiene estos atributos, pero es un caso muy especial del cual no vamos a abordar.

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

ps aux -> muestra todos los procesos del sistema

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

background -> lugar donde se ejecutan procesos ocultos.

foreground -> lugar donde se ejecutan procesos visibles.

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

fg y bg % PID

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

Pipe es un comando muy potente el cual sirve como anexo entre varias instrucciones, permitiendo así una mayor y mas rápida manipulación de archivos y sus contenidos. ej:

```
$ cat contents.txt |grep
```

```
file 0 Aug 9 13:55
```

```
file1 0 Aug 9 13:55
```

```
file2 0 Aug 9 13:55
```

```
file3 0 Aug 9 13:55
```

```
file4 0 Aug 9 13:55
```

```
file5 0 Aug 9 13:55
```

```
$ cat contents.txt |grep "file"|awk '{print $8}'
```

```
file1 file2 file3 file4 file5
```

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

comando > “archivo” reemplaza el contenido del archivo con el nuevo input del comando.

mientras que comando >> “archivo” escribe en el archivo el nuevo input sin sobrescribir lo anterior.

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

el comando kill sirve para enviar mensajes sencillos a los distintos procesos, este por defecto envía la señal de terminación (SIGTERM). otras señales interesantes son: HUP, TRAP, INT y ALRM.

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

ps -> "process status" muestra el estado del proceso

kill -> ya fue mencionado arriba, envía una señal al proceso, por defecto le pide que termine, pero no necesariamente sirve solo para eso.

pstree -> muestra el árbol de procesos, indicando padres hijos y sus estados.

killall -> en lugar de una señal a un proceso de identificación manda la señal a un grupo de procesos.

top -> El comando top te permite ver las tareas del sistema que se ejecutan en tiempo real

nice -> inicia un proceso con un nivel de prioridad definido por el usuario, esto puede darle más tiempo en CPU asignado por el kernel.

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

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

a- Es agrupar en un solo fichero varios ficheros y/o directorios

(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?

b-

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

Para comprimir varios archivos a la vez usando gzip, usamos el comando gzip, seguido de los nombres de los archivos a comprimir separados por un espacio. gzip main.py file.py process.py

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

D- Podemos comprimir un conjunto de archivos usando el comando .tar, que se encarga de comprimir un conjunto de archivos

(e) Investigue la funcionalidad de los siguientes comandos:
tar se encarga de comprimir un conjunto de archivos.

grep Así pues, **Grep** (por sus siglas en inglés Globally Search For Regular Expression and Print out) **es** una herramienta de líneas de comando usada para buscar cadenas de texto y encontrar coincidencias dentro de este. También puede ser utilizada para encontrar una palabra o combinación de palabras dentro de un fichero.

gzip **Gzip** (GNU Zip) es una herramienta de compresión, que se utiliza para trunca el tamaño de un archivo

zgrep comando para buscar dentro de un archivo comprimido. Sería lo mismo que la operación en un archivo sin comprimir grep -i nombre de archivo. Todas las opciones del comando zgrep se pararán a grep y el archivo se enviará al comando grep

wc word count es un comando que permite realizar diferentes conteos, ya sea de palabras, caracteres o saltos de líneas. wc -l número de líneas wc -c número de bytes wc -m imprime el número de caracteres wc -L imprime la longitud de la línea más larga wc -w imprime el número de palabras

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 loguea). En caso de no poder ejecutarse el comando, indique la razón:

```
ls -l > prueba
ps > PRUEBA
chmod 710 prueba
chown root:root PRUEBA
chmod 777 PRUEBA
chmod 700 /etc/passwd
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`

i. Listará el contenido del directorio en forma de columna y lo guardará en el archivo “prueba”

ii. Los estados de los procesos se guardarán en el archivo “PRUEBA”

iii. Cambiará los permisos para el archivo “prueba”. Propietario: leer, escribir y ejecutar. Grupo: ejecutar. Otros: nada

iv. Cambia el archivo PRUEBA al grupo y usuario root

v. Cambia los permisos para el archivo “PRUEBA”. Propietario, grupo y otros podrán leerlo, escribirlo y ejecutarlo

vi. Cambia los permisos de /etc/passwd, Propietario podrá leerlo, escribirlo y ejecutarlo mientras que el grupo y otros no podrán hacer nada

vii. Cambia la contraseña de root

viii. Eliminará el archivo PRUEBA

ix. Mostrará información con respecto al directorio /etc/shadow x. Busca todos los archivos de tipo .conf

xi. Cambia el directorio de inicio de root al de /home/newroot

xii. Me posiciono en /root xiii. Borra todo lo contenido en ese directorio

xiv. Me posiciono en /etc

xv. Copio todo lo del directorio /home incluidos los subdirectorios

xvi. Apaga el sistema

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

(a) Terminar el proceso con PID 23.

A- kill 23

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

B- kill init (no está permitido).

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

C- find /home -name “.conf”

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

D- ps > home/iso/procesos

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

Usuario: Lectura, escritura, ejecución

Grupo: Lectura, ejecución

Otros: ejecución

E- chmod 751 /home/iso/xxxx

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

Usuario: Lectura, escritura.

Grupo: Lectura, ejecución

Otros: Ninguno

F- chmod 650 /home/iso/yyyy

(g) Borrar todos los archivos del directorio /tmp
usamos rmdir /tmp o rm /tmp

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

H- usamos el comando chown para cambiar de propietario:

chown iso2022 : iso2022 /opt/isodata

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

I- pwd >> /home/bruno/donde

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

(a) Ingrese al sistema como usuario "root"

A- sudo -i

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

B - useradd s
passwd s

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

C- Fue modificado /etc/group and /etc/passwd y /home/user/s fue creado

(d) Crear un directorio en /tmp llamado cursada 2017

D- mkdir /tmp/cursada2022

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

E- 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.

F- chown s : users /tmp/cursada2022

(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.

G- chmod 723 /tmp/cursada2022

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

H- su -s

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

I- hostname

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

J- ps r

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

K- who -q

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

L- sudo write s < "Se va a apagar el sistema"

(m) Apague el sistema.

M- 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.

A- mkdir /home/187735 cd /home/187735

(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".

B- touch leame.txt vi leame.txt

-i Bruno Pestana

-i 18773/5

-i brunpesta@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 permisos

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

(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.

(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".

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:

```
mkdir i s o
cd . / i s o ; ps > f 0
l s > f 1
cd /
echo $HOME
l s - l $> $HOME/ i s o / l s
cd $HOME; mkdir f 2
l s -l d f 2
chmod 341 f 2
touch d i r
cd f 2
cd ~/ i s o
pwd >f 3
ps | g r e p 'ps' | wc - l >> . . / f 2 / f 3
chmod 700 . . / f 2 ; cd . .
f i n d . -name e t c / passwd
f i n d / -name e t c / passwd
mkdir e j e r c i c i o 5
```

i. Crea la carpeta iso

ii. Accede a la carpeta /iso y lista los procesos ejecutándose guardándolos en f0

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

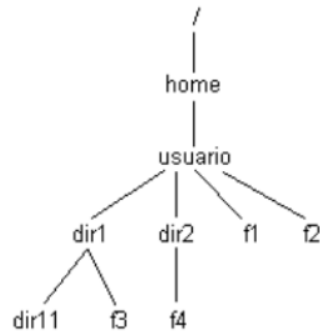
una de las órdenes que ejecute en la otra sesión, realice una breve explicación de los resultados obtenidos.

(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 órdenes 19 y 20 y comentelas 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:

(a) Utilizando la estructura de directorios anteriormente creada, indique que comandos son necesarios para realizar las siguientes acciones:
Mueva el archivo "f3.al directorio de trabajo /home/usuario.

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

```
cd /home/dir2 cp f4 /home/bruno/dir1/dir11
```

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

```
"f7" cd /home/dir2
```

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

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

```
mkdir copia cp dir1 /home/bruno/copia
```

Renombre el archivo "f1"por el nombre archivo y vea los permisos del mismo. cd /home/bruno 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" "f4" de manera que se llamen "f3.exe" "f4.exe" respectivamente.

```
mv f3 f3.exe cd /home/sofia/dir 1/dir 11 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) Empaque el directorio creado en 1, el archivo resultante se debe llamar "misLogs.tar".

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

(d) Empaque y comprime 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/bruno/primero cp misLogs.tar.gz /home/bruno/segundo
```

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

```
cd /tmp rmdir logs
```

(g) Desempaque 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
```