

## ISO PRAC 2

### 1. Editor de textos:

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

Vim nano y

(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.

El editor de texto permite crear, modificar, y guardar archivos de texto. Ejemplos son **vi**, **nano**, y **gedit**.

El comando cat muestra el contenido de un archivo en la terminal, concatenando si son varios archivos (no permite edición).

El comando more muestra el contenido de un archivo de texto de forma paginada, permitiendo avanzar en bloques, pero no editar.

El comando less permite navegar hacia adelante y atrás en el archivo de manera más flexible. Tampoco permite edición.

En el editor de texto se puede modificar el contenido de un archivo.

Los comandos cat, more y less solo muestran contenido; no permiten modificar archivos.

Modo de comando es el predeterminado en el vi, se puede navegar borrar texto copiar y pegar pero no editar directamente el contenido.

El modo de inserción es **i** (insertar), **a** (añadir después del cursor), o **o** (abrir una nueva línea).

Y el modo de línea de comandos, se accede presionando **:** desde el **modo de comando**.

Permite realizar tareas como guardar (**:w**), salir (**:q**), buscar (**:/patrón**) o reemplazar (**:s/antiguo/nuevo/**).

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

#### #### Navegación:

- **h**: Mueve el cursor a la izquierda.
- **j**: Mueve el cursor hacia abajo.
- **k**: Mueve el cursor hacia arriba.
- **l**(**L**): Mueve el cursor a la derecha.
- **w**: Avanza al inicio de la siguiente palabra.
- **b**: Retrocede al inicio de la palabra anterior.
- **0**: Va al inicio de la línea actual.
- **\$**: Va al final de la línea actual.
- **gg**: Va al inicio del archivo.
- **G**: Va al final del archivo.

#### #### Inserción de texto:

- **i**: Inserta texto antes del cursor.
- **I**: Inserta texto al inicio de la línea.
- **a**: Inserta texto después del cursor.
- **A**: Inserta texto al final de la línea.

- ``o``: Abre una nueva línea debajo del cursor e inserta.
- ``O``: Abre una nueva línea arriba del cursor e inserta.

#### #### Borrar texto:

- ``x``: Borra el carácter bajo el cursor.
- ``dw``: Borra hasta el inicio de la siguiente palabra.
- ``dd``: Borra la línea actual.
- ``D``: Borra desde el cursor hasta el final de la línea.

#### #### Deshacer y rehacer:

- ``u``: Deshace la última acción.
- ``Ctrl + r``: Rehace la última acción deshecha.

#### #### Guardar y salir:

- ``w``: Guarda el archivo.
- ``wq`` o ``ZZ``: Guarda y cierra.
- ``q``: Sale si no hay cambios.
- ``q!``: Sale sin guardar los cambios.

#### #### Copiar, cortar y pegar:

- ``yy``: Copia la línea actual.
- ``p``: Pega después del cursor.
- ``P``: Pega antes del cursor.
- ``dd``: Corta la línea actual.

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

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

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

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

El que lo ejecuta es el núcleo (kernel), después de que el núcleo ha completado su inicialización y ha montado el sistema de archivos raíz, busca y ejecuta el proceso **init**. En sistemas GNU/Linux modernos, el proceso **init** se inicia automáticamente con el kernel y es el primer proceso

que se ejecuta en modo de usuario. El núcleo ejecuta init como el proceso con el ID de proceso (PID) 1.

Su objetivo es iniciar y gestionar otros procesos y servicios en el sistema, configurando el entorno del sistema, y manejando las transiciones entre diferentes estados del sistema, como el apagado y reinicio.

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

Los runlevels son niveles predefinidos en los que el sistema puede operar, y cada runlevel define qué servicios y procesos se deben iniciar o detener. Es el modo en el que arranca linux. Divide el proceso de arranque en niveles. Cada uno es responsable de levantar o bajar una serie de servicios.

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

Runlevel 0: Apagar el sistema, halt (parada)

Runlevel 1: Modo de usuario único

Runlevel 2: Modo multiusuario sin red

Runlevel 3: Modo multiusuario con red por consola

Runlevel 4: no se usa

Runlevel 5: X11 Modo multiusuario con login gráfico basado en X

Runlevel 6: Reiniciar el sistema (reboot)

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

El propósito de este archivo era controlar el proceso de inicio y el comportamiento de los niveles de ejecución (runlevels) del sistema. Al información que se almacena en él es niveles de ejecución (runlevels), acciones específicas que quiero que se ejecuten y configuración de terminales. su estructura es **id:nivelesEjecucion:acción:proceso**

**Id:** : identifica la entrada en inittab (1 a 4 caracteres)

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

Las **acciones** pueden ser

wait inicia cuando entra al runlevel e init espera a que termine

initdefault que define el nivel de ejecución predeterminado

ctrlaltdel: se ejecutará cuando init reciba la señal SIGINT

respawn, once, off, sysinit, boot, bootwait, powerwait ,etc

Proceso: el proceso que será ejecutado

por ejemplo :

**1:2345:respawn:/sbin/getty 38400 tty1**

el 1 siendo el

/sbin/getty 38400 tty1 el proceso que se ejecuta el getty en la terminal tty1

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

who -r para saber en que runlevel estoy

init Y

No es permanente. Es un cambio temporal que afecta el estado actual del sistema hasta que se reinicie o se cambie el runlevel nuevamente.

Para que un cambio de runlevel sea permanente (es decir, para que el sistema arranque siempre en un runlevel específico después de reiniciar), debes configurar el runlevel predeterminado en el archivo `/etc/inittab`.

Id:5:initdefault:

**(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.**

????????????????????

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

**(a) ¿Qué es systemd?**

SystemD es un sistema y un manager de servicio para los sistemas operativos de Linux. Este provee un sistema de dependencia entre varias entidades llamadas units. Estas encapsulan varios objetos que son relevantes para el reinicio de y mantenimiento del sistema.

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

SystemD provee un sistema de dependencia entre varias entidades llamadas units. Estas encapsulan varios objetos que son relevantes para el reinicio de y mantenimiento del sistema. La mayoría de units están configuradas en archivos de configuración de units. Las units pueden estar activas o inactivas.

Hay muchos units

- 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 booteo (.target) → dependencia de unidades
- Snapshot: almacena el estado de un conjunto de unidades que puede ser restablecido más tarde (.snapshot)
- etc.

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

Es una herramienta de línea de comandos que se utiliza para interactuar y gestionar los servicios y procesos controlados por **systemd**

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

Estos son la equivalencia de runlevels en systemd

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

Muestra un árbol de procesos que están corriendo.

### **4. Usuarios:**

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

Los archivos usados para guardar la información son `/etc/passwd`, `/etc/group` y `/etc/shadow`.

`/etc/passwd` es el archivo principal donde se almacena la información básica de los usuarios del sistema.

`/etc/group` este archivo contiene la información de los grupos en el sistema.

`/etc/shadow` este archivo almacena las **contraseñas cifradas** de los usuarios, así como la información sobre la expiración de las contraseñas

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

UID (user id?) es un número que identifica de manera única a un usuario dentro del sistema

GID es un número que identifica de manera única a un grupo en el sistema

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

Root es el administrador del sistema (superusuario).

El **UID** (User ID) del usuario **root** es siempre **0**. Este es un valor reservado que le otorga los privilegios completos del sistema.

Técnicamente, **sí**, se pueden crear otros usuarios con los mismos privilegios que el usuario **root**, pero el sistema los reconoce a través de su **UID 0**. Esto significa que puedes tener más de un usuario con el **UID 0**, lo que efectivamente le daría al nuevo usuario los mismos privilegios que root. Es más común usar **sudo** para dar acceso limitado a privilegios elevados a otros usuarios.

**(d) Agregue un nuevo usuario llamado iso2017 a su instalación de GNU/Linux, especifique que su home sea creada en /home/iso\_2017, y hágalo miembro del grupo catedra (si no existe, deberá crearlo). Luego, sin iniciar sesión como este usuario cree un archivo en su home personal que le pertenezca. Luego de todo esto, borre el usuario y verifique que no queden registros de él en los archivos de información de los usuarios y grupos.**

```
Sudo groupadd catedra
```

```
Sudo useradd -m -d /home/iso_2017 -G catedra iso2017
```

```
Sudo passwd iso2017
```

```
Sudo touch /home/iso_2017/texto.txt
```

```
Sudo chown iso2017:catedra /home/iso_2017/texto.txt
```

```
Sudo userdel -r iso2017
```

```
Cat /etc/passwd | grep iso2017
```

```
Cat /etc/group | grep catedra
```

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

**useradd** ó **adduser** sirve para crear un nuevo usuario

**Usermod** cambia los permisos de un usuario

**Userdel** borra un usuario

**Su** sirve para cambiar de usuario la terminal

**Groupadd** crea un nuevo grupo de usuarios

**Who** muestra una lista de los usuarios actualmente logueados en el sistema, junto con la terminal que están utilizando y la hora de inicio de sesión.

**Groupdel** borra un grupo de usuarios

**Passwd** sirve para cambiar la contraseña de un usuario

## **5. 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

**(b) Investigue la funcionalidad y parámetros de los siguientes comandos relacionados con**

## los permisos en GNU/Linux:

**Chmod** modifica los permisos de acceso de cada archivo

El formato es [ugoa][-=] [perms]

ugoa dice quien va a disponer el acceso. u: el usuario que lo posee;g: otros usuarios del mismo grupo; o: otros usuarios que no están en el grupo del archivo; a: todos los usuarios.(si no se especifica ninguno se sobreentiende a)

+: para añadir un determinado modo al archivo; -: se usa para quitar un modo al archivo y = sirve para añadir los modos que se especifiquen borrando todos los otros.

R: lectura; w:escritura; x: ejecución o permiso de búsqueda; X: ejecución/búsqueda solo si se trata de un directorio o ya tiene permisos;s: SETGID o SETUID en la ejecución

Hay un modo numerico. (0-7)

**Chown** modifica la propiedad del usuario o grupo de cada archivo dado.

chown [opcion] [propietario] [:grupo] [fichero]

Las opciones pueden ser **-R**: Recursivo, aplica el cambio a todos los archivos y subdirectorios.

**--from**: Cambia el propietario solo si coincide con un propietario específico.

**-h**: Cambia el propietario de un enlace simbólico.

**-v**: Salida detallada de los cambios.

**-c**: Salida de cambios solo si hay modificaciones.

**Chgrp** cambia el grupo de pertenencia de archivos y directorios

**chgrp [options] <group> <file/directory>**

Tiene los mismos comandos que chown

## (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?

Significa que se usa un número en base 8 para representar la asignación de permisos a archivos y directorios

**0**: Sin permisos.

**1**: Sólo ejecución.

**2**: Sólo escritura.

**3**: Escritura y ejecución.

**4**: Sólo lectura.

**5**: Lectura y ejecución.

**6**: Lectura y escritura.

**7**: Lectura, escritura y ejecución.

## (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.

```
sudo chmod 600 archivito.txt
```

```
cat archivito.txt
```

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

El **nombre de ruta completo** especifica la ubicación exacta de un archivo o directorio desde la raíz del sistema de archivos. Siempre comienza con el carácter /, que representa la raíz. Este tipo de ruta es independiente del directorio de trabajo actual.

```
/home/usuario/documentos/documento.txt
```

El **nombre de ruta relativo** especifica la ubicación de un archivo o directorio en relación con el directorio de trabajo actual. No comienza con el carácter `/` y se basa en la posición actual en el sistema de archivos.

documentos/documento.txt

**(f) ¿Con qué comando puede determinar en qué directorio se encuentra actualmente?**

**¿Existe alguna forma de ingresar a su directorio personal sin necesidad de escribir todo el path completo? ¿Podría utilizar la misma idea para acceder a otros directorios?**

**¿Cómo? Explique con un ejemplo.**

El comando para saber en qué directorio se encuentra es `cd`. Para ingresar al directorio personal sin escribir todo el path completo es con `cd ~` o `cd solo` o `cd $HOME`. Si con el nombre de la carpeta si esta está en el directorio personal. Por ejemplo `cd carpetita` o `cd carpetita/yo`.

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

**cd** sirve para entrar a un directorio en específico.

**umount** desmonta un sistema de archivos. Se utiliza para desconectar un dispositivo o una partición del sistema de archivos.

**mkdir** Crea un nuevo directorio.

**du** Muestra el uso del espacio en disco de los archivos y directorios. Por defecto, muestra el tamaño de cada directorio y subdirectorio en el directorio actual.

**rmdir** elimina un directorio vacío. Si el directorio contiene archivos o subdirectorios, no podrá eliminarlo.

**df** muestra información sobre el uso del espacio en disco de los sistemas de archivos montados. Incluye el tamaño total, el uso y el espacio disponible.

**mount** monta un sistema de archivos, conectando un dispositivo (como un disco duro o una USB) al sistema de archivos, que es el proceso de hacer que un dispositivo o partición sea accesible en el sistema de archivos.

**ln** Crea enlaces a archivos. Puede crear enlaces simbólicos (similares a accesos directos) o enlaces duros.

**ls** lista los archivos y directorios en el directorio actual. Hay varias opciones para modificar su comportamiento. (El `-l` muestra información detallada.)

**pwd** Muestra el directorio de trabajo actual (el directorio en el que estás trabajando).

**cp** copia archivos o directorios. Puedes usarlo para copiar un solo archivo o múltiples archivos/directorios.

**mv** Mueve o renombra archivos y directorios.

## **6. 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.**

Un proceso es una instancia en ejecución de un programa. En sistemas operativos, un proceso incluye el programa en sí, así como su estado en ejecución, los datos, y la información de contexto necesaria para que el sistema operativo gestione su ejecución.

**PID: Process Identifier (Identificador de Proceso)** es un número único asignado por el sistema operativo a cada proceso en ejecución. Este identificador se utiliza para gestionar y realizar un seguimiento de los procesos.

**PPID: Parent Process Identifier (Identificador de Proceso Padre)** es el PID del proceso que creó (o "padre" de) otro proceso. Cada proceso puede tener un solo proceso padre.

Sí, todos los procesos en GNU/Linux tienen un PID y un PPID. Esto es fundamental para la gestión de procesos, ya que permite al sistema operativo rastrear la jerarquía de procesos y gestionar su ejecución.

Otros atributos son estado, prioridad, y uso de recursos

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

ps muestra información sobre los procesos en ejecución o top muestra un resumen dinámico de los procesos en ejecución, actualizándose en tiempo real.

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

Que se esté ejecutando en background es que se está ejecutando en segundo plano, se puede seguir usando la terminal.

Un proceso que se está ejecutando en **foreground** se ejecuta directamente en la terminal y ocupa la entrada y salida estándar. Esto significa que puedes interactuar con él, proporcionando entradas y recibiendo salidas en la misma ventana de terminal.

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

Se puede escribir **&** al final del comando.

**Pasar a Foreground:** Usa fg %job\_number.

**Pasar a Background:** Presiona Ctrl + Z para suspender, luego usa bg.

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

Un pipe es una herramienta poderosa que permite encadenar comandos, redirigiendo la salida de uno como entrada a otro, lo que facilita el procesamiento de datos en la terminal de manera eficiente y modular.

ls | grep "informe" | wc -l

**ls:** Lista los archivos en el directorio.

**grep "informe":** Filtra la salida para mostrar solo los archivos que contienen la palabra "informe".

**wc -l:** Cuenta las líneas de la salida anterior, que corresponde al número de archivos que cumplen con el criterio.

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

La redirección es una herramienta poderosa en la línea de comandos que permite gestionar la entrada y salida de datos de manera flexible, facilitando el trabajo con archivos y la automatización de tareas.

**>:** Redirige la salida estándar de un comando a un archivo. Si el archivo ya existe, se sobrescribe.

**>>:** Redirige la salida estándar de un comando a un archivo, pero agrega el contenido al final del archivo en lugar de sobrescribirlo.

**<:** Redirige la entrada estándar de un comando a partir de un archivo, en lugar de tomar la entrada desde la terminal.

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

Enviar señales a procesos en ejecución. Aunque su nombre sugiere que solo se utiliza para terminar procesos, en realidad, kill puede enviar diferentes tipos de señales, lo que permite realizar diversas acciones sobre los procesos.

kill [opciones] <PID>



kill -STOP 1234

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

**ps** Muestra procesos activos

-e: Muestra todos los procesos.

-f: Muestra información completa (formato extendido).

**kill** envía una señal a un proceso.

-9: Fuerza el cierre inmediato del proceso (SIGKILL).

-l: Lista todas las señales disponibles.

**ps tree** Muestra la jerarquía de procesos en forma de árbol

-p: Muestra los PIDs junto a los nombres de los procesos.

-u: Muestra el usuario que ejecuta cada proceso.

**killall** Envía señales a todos los procesos con un nombre específico

-9: Fuerza el cierre inmediato de los procesos.

-r: Usa una expresión regular para coincidir con el nombre del proceso.

**top** Muestra procesos en tiempo real

-u: Muestra solo los procesos de un usuario específico.

-p: Muestra solo los procesos con los PIDs especificados.

**nice** Establece prioridad de un proceso

-n: Especifica el valor de "nice" (de -20 a 19).

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

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

El concepto de **empaquetar archivos** en GNU/Linux hace referencia al proceso de **agrupar múltiples archivos y directorios en un solo archivo**, sin necesariamente comprimirlos. Esto se suele hacer para facilitar el manejo, transferencia o almacenamiento de un conjunto de archivos que están relacionados entre sí, como el contenido de un proyecto o una carpeta con muchos ficheros.

**(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?**

Tar -cvf archivo.tar ar1.sh ar2.sh ar3.sh ar4.sh

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

Tar -cvf archivo.tar ar1.sh ar2.sh ar3.sh ar4.sh

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

Si

**(e) Investigue la funcionalidad de los siguientes comandos:**

**Tar** empaqueta o empaqueta y comprime archivos

```
tar -czvf archivo_comprimido.tar.gz archivo1.txt archivo2.txt
```

```
tar -cvf archivo.tar directorio1 archivo2
```

**grep** muestra las líneas que concuerdan con el patrón. Busca patrones en cada archivo , grep muestra cada línea donde se encuentra una concordancia con dicho patrón.

```
grep [opcion]"texto" archivo.txt
```

-i ignora la distinción entre mayus y minus

-n muestra el numero de linea de la coincidencia antes de imprimir la línea

-v busca solo las lineas que no contienen el patrón

-r busca el patrón en todos los archivos dentro de un directorio y sus subdirectorios.

^ : busca al principio de la línea y \$ al final de la línea

```
ls -l | grep "txt" filtra todas las lineas que contienen txt
```

**gzip** comprime archivos utilizando el algoritmo de compresión DEFLATE. Su principal función es reducir el tamaño de archivos para facilitar el almacenamiento o la transferencia.

```
gzip [opcion] archivo.txt
```

-k crea un archivo comprimido pero no borra el anterior **gzip -k archivo.txt**

-d descomprime un archivo gzip -d archivo.txt.gz

-número especifica el nivel de compresión siendo el 9 el más alto **gzip -9 archivo.txt**

**zgrep** se utiliza para buscar patrones dentro de archivos comprimidos con gzip, sin descomprimirlos

```
zgrep [opciones] patrón [archivo.gz...]
```

```
zgrep "texto" archivo.txt.gz
```

-i ignora la distinción entre mayus y minus

-n muestra el numero de linea de la coincidencia antes de imprimir la línea

-v busca solo las lineas que no contienen el patrón

-r busca el patrón en todos los archivos dentro de un directorio y sus subdirectorios.

^ : busca al principio de la línea y \$ al final de la línea

```
ls -l | grep "txt" filtra todas las lineas que contienen txt
```

MISMOS COMANDOS QUE GREP

**wc** se utiliza para contar líneas, palabras y caracteres en archivos o en la salida de otros comandos, su nombre viene de word count

```
wc [opcion] archivo.txt
```

-l muestra solo el numero de lineas

-w muestra solo el numero de palabras

-m muestra solo el numero de caracteres

-c muestra el numero de bytes en el archivo

```
ls | wc -l cuenta cuántos archivos hay en el directorio actual
```

**8. Indique qué acción realiza cada uno de los comandos indicados a continuación considerando su orden. Suponga que se ejecutan desde un usuario que no es root ni pertenece al grupo de root. (Asuma que se encuentra posicionado en el directorio de trabajo**

**del usuario con el que se logueó). En caso de no poder ejecutarse el comando, indique la razón:**

**ls -l > prueba** lista el contenido del directorio actual en formato largo (detallado) y redirige la salida al archivo llamado prueba

**ps > PRUEBA** muestra los procesos en ejecución y redirige la salida a un archivo llamado PRUEBA  
**chmod 710 prueba** le da todos los permisos al usuario que está usando la terminal solo permisos de solo ejecución a los usuarios del mismo grupo y ningún permiso a otros usuarios que no son del grupo

**chown root : root PRUEBA** Cambia el propietario y el grupo del archivo PRUEBA al usuario y grupo root **no se ejecuta** porque el usuario no es root

**chmod 777 PRUEBA** le da todos los permisos al archivo prueba a el grupo al usuario y a los usuarios fuera del grupo

**chmod 700 / etc / passwd** Intenta cambiar los permisos del archivo /etc/passwd a 700, dando solo al propietario (root) todos los permisos, y bloqueando el acceso a otros usuarios. **No se ejecuta** xq el archivo es un archivo del sistema y solo el usuario root puede cambiarlos

**passwd root** Intenta cambiar la contraseña del usuario root. **No se ejecuta** xq no es el usuario root  
**rm PRUEBA** borra el archivo prueba

**man / etc / shadow** quiere mostrar la página del manual del archivo pero **no puede** ya que solo el usuario root tiene los permisos y este no tiene una página del manual

**find / -name \* . c o n f** Busca archivos con extensión .conf en todo el sistema a partir del directorio raíz /

**usermod root -d /home/ newroot -L** Intenta modificar el usuario root, cambiando su directorio personal a /home/newroot y bloqueando su cuenta (-L). **No se puede ejecutar** porque el comando root solo se puede ejecutar por el usuario root o el superusuario

**cd / root** intenta cambiar el directorio a /root. **No se puede ejecutar** xq solo el root tiene accesos a ese directorio

**rm \*** si tiene permisos el usuario en el directorio borra todos los archivos en el directorio

**cd / etc** cambia el directorio al /etc

**cp \* /home -R** intenta copiar todos los archivos al directorio /home de manera recursiva pero **no se puede ejecutar** xq el usuario no tiene permisos en el directorio /home

**Shutdown** intenta apagar el sistema pero **no se ejecuta** porque solo el usuario root puede hacerlo

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

**(a) Terminar el proceso con PID 23.**

Sudo kill 23 o kill 23

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

Kill 1 pero el kernel no lo permite porque es un proceso restringido, si se detiene se apaga o reinicia el sistema

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

find ~ -name "\*.conf"

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

ps aux > /home/\$(whoami)/procesos ps (lista de proc en ejecución) la redirige al archivo procesos en /home/mi nombre de usuario/procesos

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

**Usuario:** Lectura, escritura, ejecución

**Grupo:** Lectura, ejecución

**Otros:** ejecución

`chmod 751 /home/$(whoami)/xxxx`

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

**Usuario:** Lectura, escritura.

**Grupo:** Lectura, ejecución

**Otros:** Ninguno

`chmod 650 /home/$(whoami)/yyyy`

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

`Sudo rm -rf /tmp/*` recursivamente en /tmp

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

`sudo (xq no tiene permisos) chown iso2010 /opt/isodata`

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

`pwd >> /home/$(whoami)/donde`

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

**(a) Ingrese al sistema como usuario “root”**

`su -`

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

`sudo useradd MDattoli`

`sudo passwd MDattoli`

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

Se crea el directorio de inicio y se modifica el /etc/passwd, el /etc/shadow, el /etc/group

**(d) Crear un directorio en /tmp llamado cursada2017**

`mkdir /tmp/cursada2017`

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

`cp /var/log /* /tmp/cursada2017/` cualquier cosa del var al nuevo

**(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.**

`sudo chown -R MDattoli:users /tmp/cursada2017`

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

`chmod 725 -R /tmp/cursada2017`

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

`Ctrl + Alt + F2 a F6`

**(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 aux | wc -l`

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

`who |wc -l`

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

Write MDattoli

**(m) Apague el sistema.**

Shutdown now o shutdown +1(en un minuto)

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

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

mkdir 24568/9

cd 24568/9

oo

mkdir 24568/9 && cd \$\_

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

vi LEAME

datos

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

**Dueño: ningún permiso**

**Grupo: permiso de ejecución**

**Otros: todos los permisos**

chmod 017 LEAME

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

cd /etc

ls -l

ls -l >~/leame

**(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.**

Depende si tengo que buscar el contenido o el nombre

Si es por el nombre **ls /etc | grep "config"**

Si es por el contenido **ls /etc | grep "config"**

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

find / -type f -name "\*.so" > ~/24568/9/.ejercicio\_f (busco por nombre)

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

**mkdir i s o** crea la carpeta iso en el directorio actual

**cd /iso ; ps > f0** se cambia el directorio actual a /iso la otra parte muestra los procesos en ejecución y los guarda en f0

**ls > f1** guarda todos los archivos en el directorio actual a f1 se ejecuta si el usuario tiene permisos de escribir en el directorio actual

**cd /** cambia el directorio actual al raíz

**echo \$HOME** muestra el valor de la variable HOME

**ls -l \$> \$HOME/iso/ls** ls lista los archivos en el directorio actual de forma detalladamente y no sigue ya que esta mal escrito sería > solamente

**cd \$HOME; mkdir f2** cambia el directorio actual a \$HOME crea el directorio f2

**ls -ld f2** muestra información del archivo f2

**chmod 341 f2** le da permisos a f2 si existe

**touch dir** crea un archivo llamado dir en el directorio personal

**cd f2** se posiciona en el directorio f2

**cd ~/iso** cambia al directorio iso en el directorio personal del usuario

**pwd >f3** guarda la ruta del directorio actual en f3

**ps | grep 'ps' | wc -l >> . / f2 / f3** Muestra la lista de procesos en ejecución, filtra la salida para encontrar líneas que contengan "ps" y cuenta cuántas líneas hay, luego redirige la salida al archivo f3 dentro del directorio f2.

**chmod 700 . / f2 ; cd . .** cambia los permisos del directorio f2 y regresa al directorio anterior

**find . -name etc/passwd** Busca un archivo llamado etc/passwd en el directorio actual y sus subdirectorios. Pero no encuentra nada a menos q haya uno llamado así en el directorio actual

**find / -name etc/passwd** busca un archivo llamado etc/passwd en todo el sistema de archivos.

**mkdir ejercicio5** crea un directorio llamado ejercicio5

.....

.....

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

ninganas

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

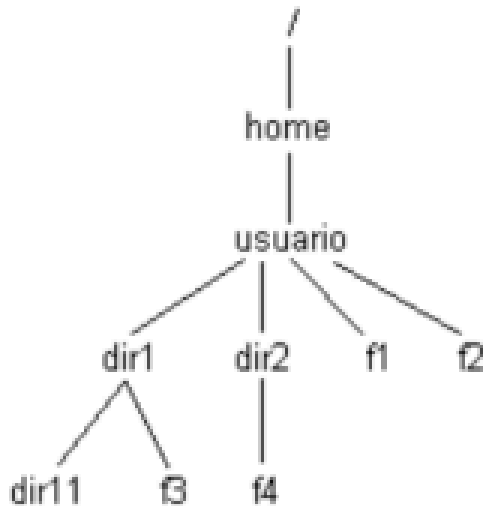
**19: Copiar el directorio iso y todo su contenido al directorio creado en el inciso 9.a).**

**cp /iso/\* iso2017**

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

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

13. Cree una estructura desde el directorio /home que incluya varios directorios, subdirectorios y archivos, según el esquema siguiente.



Asuma que “usuario” indica cuál es su nombre de usuario. Además deberá tener en cuenta que dirX hace referencia a directorios y fX hace referencia a archivos:

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

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

```
mv /home/usuario/dir/f3 /home/usuario
```

Copie el archivo “f4” en el directorio “dir11”.

```
cp /home/usuario/dir2/f4 /home/usuario/dir1/dir11
```

Haga lo mismo que en el inciso anterior pero el archivo de destino, se debe llamar “F7”.

```
cp /home/usuario/dir2/f4 /home/usuario/dir1/f7
```

Cree el directorio copia dentro del directorio usuario y copie en él, el contenido de “Dir1”.

```
mkdir /home/usuario/copia
```

```
cp -r /home/usuario/dir1/* /home/usuario/copia
```

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

```
mv /home/usuario/f1 /home/usuario/archivo
```

```
ls -l /home/usuario/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 /home/usuario/archivo
```

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

```
mv /home/usuario/dir2/f4 /home/usuario/dir2/f4.exe
```

```
mv /home/usuario/dir1/f3 /home/usuario/dir1/f3.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

```
chmod 023 /home/usuario/dir1/f3.exe /home/usuario/dir2/f4.exe
```

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

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

```
mkdir /tmp/logs
```

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

```
mkdir /tmp/logs
```

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

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

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

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

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

```
cp /tmp/logmisLogs.tar.gz /tmp/logmisLogs.tar /home/$(whoami)
```

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

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

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

```
mkdir logs_from_tar
```

```
tar -xvf misLogs.tar -C logs_from_tar
```

```
mkdir logs_from_tar_gz
```

```
tar -xzvf misLogs.tar.gz -C logs_from_tar_gz
```