



# Trabajo Práctico 2

☰ Materia	ISO Práctica
⚙ Estado	In progress
📅 Fecha	
☰ Text	

## 1 - Editor de textos

- A. Nombre al menos 3 editores de textos que puede utilizar desde la línea de comandos
- 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.
- C. Nombre los comandos más comunes que se le pueden enviar al editor de textos vi

## 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.
- B. Proceso INIT. ¿Quién lo ejecuta? ¿Cuál es su objetivo?
- C. Ejecute el comando `ps tree`. ¿Qué es lo que se puede observar a partir de la ejecución de este comando?
- D. RunLevels. ¿Qué son? ¿Cuál es su objetivo?
- 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?
- 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?
- G. Suponga que se encuentra en el runlevel <X>. Indique qué comando(s) ejecutaría para cambiar al runlevel <Y>. ¿Este cambio es permanente? ¿Por qué?
- 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.
- I. ¿Qué es insserv? ¿Para qué se utiliza? ¿Qué ventajas provee respecto de un arranque tradicional?
- J. ¿Cómo maneja Upstart el proceso de arranque del sistema?
- K. Cite las principales diferencias entre SystemV y Upstart.
- L. Qué reemplaza a los scripts rc de SystemV 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:
- N. ¿Qué es systemd?

Ñ. ¿A qué hace referencia el concepto de activación de socket en systemd?

O. ¿A qué hace referencia el concepto de cgroup?

### 3. Usuarios

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

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

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

(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.

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

### 4. FileSystem

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

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

(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?

(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.

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

(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.

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

### 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? Justifique. Indique qué otros atributos tiene un proceso.

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

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

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

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

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

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

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

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

- (a) ¿A qué hace referencia el concepto de empaquetar archivos en GNU/Linux?
- (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?
- (c) ¿Qué acciones debe llevar a cabo para comprimir 4 archivos en uno solo? Indique la secuencia de comandos ejecutados.
- (d) ¿Pueden comprimirse un conjunto de archivos utilizando un único comando?
- (e) Investigue la funcionalidad de los siguientes comandos:

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:

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

- (a) Terminar el proceso con PID 23.
- (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”
- (d) Guardar una lista de procesos en ejecución el archivo /home/<su nombre de usuario>/procesos
- (e) Cambiar los permisos del archivo /home/<su nombre de usuario>/xxxx a:
- (f) Cambiar los permisos del archivo /home/<su nombre de usuario>/yyyy a:
- (g) Borrar todos los archivos del directorio /tmp
- (h) Cambiar el propietario del archivo /opt/isodata al usuarioiso2010
- (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.

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

- (a) Ingrese al sistema como usuario “root”
- (b) Cree un usuario. Elija como nombre, por convención, la primer letra de su nombre seguida de su apellido. Asígnele una contraseña de acceso.
- (c) ¿Qué archivos fueron modificados luego de crear el usuario y qué directorios se crearon?
- (d) Crear un directorio en /tmp llamado cursada2017
- (e) Copiar todos los archivos de /var/log al directorio antes creado.
- (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.
- (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.
- (h) Acceda a otra terminal virtual para loguearse con el usuario antes creado.
- (i) Una vez logueado con el usuario antes creado, averigüe cuál es el nombre de su terminal.
- (j) Verifique la cantidad de procesos activos que hay en el sistema.

- (k) Verifiqué la cantidad de usuarios conectados al sistema.
- (l) Vuelva a la terminal del usuario root, y envíele un mensaje al usuario anteriormente creado, avisándole que el sistema va a ser apagado.
- (m) Apague el sistema.
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.
- (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".
- (c) Cambie los permisos del archivo LEAME, de manera que se puedan ver reflejados los siguientes 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:
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:
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):

# 1 - Editor de textos

## A. Nombre al menos 3 editores de textos que puede utilizar desde la línea de comandos

VI, EMACS 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.**

Los comandos cat, more y less sirven para **mostrar el contenido de ficheros de texto**. La diferencia está en cómo se muestra el contenido. La **diferencia con un editor de textos, es que justamente el editor permite editarlos**.

Los **modos de vi**:

- Modo comandos: podemos desplazarnos dentro de un archivo y efectuar operaciones de edición (buscar texto, eliminarlo, modificarlo, etc)
- Modo insertar: Aquí podemos escribir texto nuevo en el punto de inserción de un archivo.

## **C. Nombre los comandos más comunes que se le pueden enviar al editor de textos vi**

<https://docs.oracle.com/cd/E19620-01/805-7644/6j76klopr/index.html>

## **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 inician componentes esenciales
7. El kernel ejecuta el proceso init y se desmonta el initrd
8. Se lee el /etc/inittab
9. Se ejecutan los scripts apuntados por el runlevel 1
10. El final del runlevel 1 le indica que vaya al runlevel por defecto
11. Se ejecutan los scripts apuntados por el runlevel por defecto

12. El sistema está listo para usarse.

### **B. Proceso INIT. ¿Quién lo ejecuta? ¿Cuál es su objetivo?**

**INIT es ejecutado por el Kernel** al iniciar la maquina (es el primer proceso en ejecución tras la carga del núcleo) **y su objetivo es generar todos los demás procesos.**

### **C. Ejecute el comando `ps tree`. ¿Qué es lo que se puede observar a partir de la ejecución de este comando?**

Se muestra el árbol de procesos.

### **D. RunLevels. ¿Qué son? ¿Cuál es su objetivo?**

**El proceso de arranque lo dividimos en niveles, cada uno es responsable de levantar (iniciar) o bajar (parar) una serie de servicios. Los runlevels son los niveles de ejecución del sistema,** mientras mas bajo el nivel, menos procesos se van a encontrar en ejecución. **Los niveles más bajos se utilizan para el mantenimiento** o la recuperación de emergencia, ya que por lo general no ofrecen ningún servicio de red.

### **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?**

- 0 -> HALT (parada)
- 1 -> single user mode (monousuario)
- 2 -> multiusuario sin NFS (soporte de red)
- 3 -> full multiuser mode console (multiusuario completo por consola)
- 4 -> no se usa
- 5 -> X11 (todo completo pero con login gráfico basado en X)
- 6 -> reboot

Se encuentra definido en `/etc/inittab`.

No todas las distribuciones respetan este estándar.

### **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?**

Los modos específicos para una distribución particular se encuentran en el archivo `/etc/inittab`, su finalidad es definir el runlevel a ejecuta. Este archivo almacena líneas de comentarios que explican acerca de los roles del sistema y 1 línea que define el nivel a ejecutar.

```
id : nivelesEjecución : acción : proceso
```

Dónde:

- ID → identifica la entrada en inittab (1 a 4 caracteres)
- Niveles de 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 cuándo init reciba la señal SIGINT
  - off, respawn, once , sysinit , boot, bootwait, powerwait
- proceso → el proceso exacto que será ejecutado

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

Usaría el comando `init`. El cambio **no es permanente** ya que puedo usarlo para ir a otro runlevel. Para ir al runlevel Y, debería hacer: `init Y`

**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.**

- Script RC es el administrador de los cambios de runlevel.
- Se almacena en `etc/rc`

Al iniciar o al detenerse, se determina qué script ejecutar porque se buscan en la carpeta `/etc/rc.d/rc.local` el script a usar.

## I. ¿Qué es insserv? ¿Para qué se utiliza? ¿Qué ventajas provee respecto de un arranque tradicional?

- **Insserv es una herramienta de bajo nivel** la cual sirve para iniciar o apagar los servicios del sistema y resolver problemas de dependencias entre los scripts.
- La ventaja que provee es que no es necesario modificar los scripts en base a sus dependencias, gracias al *dependency based boot sequencing*.

## J. ¿Cómo maneja Upstart el proceso de arranque del sistema?

Upstart maneja el arranque del sistema **mediante jobs**, que son los encargados de definir que servicios o tareas van a ser ejecutados por `init`.

Cada **JOB** tiene un objetivo (goal start/stop) y un estado (state), en base a ellos se ejecuta un proceso específico y, al inicio, init emite un evento startup.

## K. Cite las principales diferencias entre SystemV y Upstart.

SystemV	Upstart
<i>Dependecy-based</i> , los eventos son <b>sincrónicos</b> . Bloquea futuras tareas hasta que la actual se haya completado, sus tareas deben ser definidas por adelantado y sólo pueden ser ejecutadas cuándo el demonio <code>inited</code> cambia de estado.	<i>Event-based</i> (ejecución de trabajos en forma <b>asincrónica</b> mediante eventos). Esto quiere decir que no arranca y detiene un servicio después de otro, sino que puede hacerlo <b>en paralelo</b> . Gestiona las tareas y servicios de inicio cuándo el sistema arranca y los detiene cuándo el sistema se apaga.
No tiene jobs	Tiene <b>jobs</b> (scripst de texto plano iniciados por el init)
Existe el <code>etc/inittab</code>	Existe pero ignora el <code>etc/inittab</code>

## L. Qué reemplaza a los scripts rc de SystemV en Upstart? ¿En que ubicación del filesystem se encuentran?

Los scripts de SystemV son reemplazados en **Upstart por un conjunto integrado de scripts de arranque**, que reemplazan por completo a `etc/inittab`. Estos se van a encontrar en el directorio `etc/init` con el nombre de `servicio.conf` dónde **servicio** es el programa que init tratará como un job.

**Los scripts de Upstart ofrecen mas acciones que los de SystemV.**

Se encuentran en la ruta `/etc/init/*.conf`



**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 S e r v i c e
d e s c r i p t i o n "MySQL S e r v e r "
a u t h o r " i n f o a u t o r "
s t a r t o n ( n e t - d e v i c e - u p # el trabajo inicia en:
a n d l o c a l - f i l e s y s t e m s
a n d r u n l e v e l [ 2 3 4 5 ] )
s t o p o n r u n l e v e l [ 0 1 6 ] # el trabajo para en:
[ . . . ]
e x e c / u s r / s b i n / m y s q l d #esto es lo que se ejecutará para el trabajo
[ . . . ]
```

## **N. ¿Qué es sytemd?**

**Es un sistema** que centraliza la administración de demonios, herramientas y librerías del sistema. **Mejora el paralelismo del booteo. Interactúa con el núcleo del sistema operativo GNU/Linux.**

## **Ñ. ¿A qué hace referencia el concepto de activación de socket en systemd?**

**Socket es un mecanismo de iniciación bajo demanda**, así podemos ofrecer una variedad de servicios sin que realmente estén iniciados. Cuando el sockect recibe una conexión spawna el servicio y le pasa el socket. No hay necesidad de definir dependencias entre servicios ya que se inician todos los sockets en primer medida.

## **O. ¿A qué hace referencia el concepto de cgroup?**

**Permite organizar un grupo de procesos en forma jerárquica.** Agrupa conjunto de procesos relacionados (por ejemplo: un servidor web apache con sus dependientes)

# **3. Usuarios**

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

- `./etc/passwd` → tiene la información sobre los usuarios (login, nombre y cualquier otra info que el administrador quiera agregar)

- `.(/etc/group)` → tiene los grupos y los usuarios que pertenecen a estos grupos
- `.(/etc/shadow)` → tiene las contraseñas encriptadas.

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

- **UID** son las ID de usuarios.
- **GID** son las ID de los grupos.

No, no pueden coexistir dos UIDs iguales, ya que si tenemos dos UIDs iguales que corresponde a distintos usuarios, el sistema operativo los va a tomar como una sola cuenta.

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

El **UID** de un usuario **root** es 0.

Ser **usuario root** significa tener acceso al directorio raíz, en dónde tenemos instalado todo el sistema operativo. Se le suele llamar **superusuario**, porque tiene acceso a todo el SO. Es similar a lo que ocurre con los derechos de administrador en Windows.

**Sólo puede haber 1 usuario root, pero puede haber más de un usuario administrador.**

**(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.**

Que paja.

```
# agrego al usuario "iso2017" con "sudo useradd"
# -d para especificar el directorio, en este caso /home/iso_2017
# -m para crear el directorio personal
# -g para agregarlo al grupo especificado, en este caso CATEDRA
sudo useradd -d /home/iso_2017 -m -g catedra iso2017
cd /home/iso_2017 # para cambiar de directorio
sudo touch carpeta #creo la carpeta "carpeta"
```

```
ls # me fijo que haya sido creada
sudo userdel -r iso_2017 # borro el directorio
sudo userdel iso2017 #elimino el usuario
```

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

- `useradd` ó `adduser`

`useradd` crea un nuevo usuario haciendo esto: `useradd nombre_usuario`

`adduser` es un enlace simbólico a `useradd`.

- `usermod`

Modifica los parámetros del usuario, como grupos, carpeta, etc

- `userdel`

`userdel nombre_usuario` borra a un usuario y todos sus datos

- `su`

Entra en el usuario root

- `groupadd`

`groupadd nombre_grupo` crea un grupo

- `who`

Dice cuál es el usuario activo en este momento

- `groupdel`

`groupdel nombre_grupo` borra un grupo

- `passwd`

`passwd nombre_usuario` modifica la contraseña de un usuario.

## 4. FileSystem

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

Los permisos son definidos por **tres caracteres**.

- El primer conjunto representa la clase del usuario
- El segundo representa la clase del grupo
- El tercero representa la clase de <<otros>> usuarios

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

- `chmod`

Es un comando que **permite modificar y especificar** quien

**puede manejar el archivo y cómo puede hacerlo.**

- **u**, **g**, **o** parámetros de usuario.
  - **u** dueño
  - **g** grupo
  - **o** otros usuarios
- **+** añade permiso
- **-** quita permiso
- **r**, **w**, **x** permisos
  - **r** read
  - **w** write
  - **x** ?????

```
chmod u+x # permiso de ejecución al dueño
chmod o+r+w # permiso de lectura y escritura a los otros usuarios
chmod g+r-w-x #deja solo permiso de lectura al grupo al que pertenece el archivo
```

- **chown** Es un comando que **permite cambiar el propietario de un archivo o directorio** en GNU. Se puede especificar el nombre de usuario o un GID.

```
chown nombreUsuario archivo1 [archivo2 archivo3...]
chown -R nombreUsuario nombreDirectorio
```

- **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.

```
chgrp [opciones] archivo(s) o directorio(s)
```

**(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?

La notación octal que se usa **representa los permisos del usuario dueño, el grupo y los otros usuarios**. Los números se representan por **la suma de permisos, siendo:**

- 4 = lectura
  - 2 = escritura
  - 1 = ejecución
- Entonces, el 5 = lectura + ejecución. El 7 = lectura + escritura + 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.

El **usuario root** puede, ya que tiene acceso a la carpeta del sistema (permisos de superusuario). Los usuarios normales no tienen permitido el acceso por razones de seguridad. **Pero, Kubuntu no tiene el usuario root (justo el SO que acabo de instalar, piola)**. Los usuarios con acceso `sudo` también pueden acceder con o sin permiso, ya que este comando posee permisos de kernel/superuser.

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

- “*full path name*” es la ruta desde el directorio raíz (root) “/”
- “*relative path name*” es la ruta iniciando desde la carpeta desde donde “se esta parado”

## (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.

- Se puede determinar el directorio en el que nos encontramos parados con `pwd`
- Para ingresar al directorio personal sin tener que escribir el path completo se puede usar `/home` o con `cd`
- Para acceder a otros directorios, es con `/etc`

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

• <code>cd</code>	Cambia de directorio
• <code>umount</code>	Desmonta un dispositivo
• <code>mkdir</code>	Crea un directorio
• <code>du</code>	Muestra lo que ocupa y el tamaño total de los directorios dentro del directorio donde me encuentro
• <code>rmdir</code>	Elimina un directorio
• <code>df</code>	Muestra los sistemas de ficheros montados
• <code>mount</code>	Monta un dispositivo
• <code>ln</code>	Crea enlaces a archivos y un fichero que apunta a otro
• <code>ls</code>	Lista los archivos y directorios dentro del entorno de trabajo
• <code>pwd</code>	Imprimir el nombre del directorio actual
• <code>cp</code>	Copia los archivos en el directorio indicando
• <code>mv</code>	Renombra un conjunto

## 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.**

Un proceso es una entidad "viva" /se modifica/es dinámico, a diferencia de los programas que son estáticos.



Un proceso es una unidad de actividad que se caracteriza por la ejecución de una secuencia de instrucciones, un estado actual y un conjunto de recursos del sistema asociados.

- **PID** hace referencia a la ID del proceso.
- **PPID** hace referencia a la ID del proceso PADRE del proceso.

Todos los procesos tienen estos atributos, además de otros como:

- usuarios (UID)
- grupos (GID)
- prioridad

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

- `ps` ("process status") permite visualizar el estado de un proceso.
- `pstree` nos muestra en forma de árbol la relación entre los procesos padres e hijos.
- `top` permite obtener información de nuestro sistema en tiempo real.
- `htop` una interfaz mas bonita (con colores), es una mejora de TOP.

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

El estado **background** se refiere a que al ser activado desde una terminal, el proceso se ejecuta de manera independiente a la terminal sin "amarrarla" durante el tiempo de proceso.

El estado **foreground** en cambio, "amarra" la terminal al proceso dejándola sin capacidad de correr otra tarea en esa terminal mientras el proceso se ejecuta.

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

La manera de ejecutar un proceso en *background* es adicionar el carácter `&` al dar el comando. Para ejecutar en *foreground* simplemente se escribe el comando en la línea de comandos

```
fg [id_trabajo ...]  
bg [id_trabajo ...]
```

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

El pipe nos permite encadenar la ejecución de programas, pasando la salida de uno como la entrada de otro.

```
history | grep "rm"
```

- `history` muestra el historial de los últimos 500 comandos utilizados.
- `grep` busca una palabra o patrón en un lugar, y devuelve todas las coincidencias.

Con el ejemplo, history genera los últimos 500 comandos usados, dicha salida va a ser la entrada para grep, que usa el argumento "rm", por lo tanto su salida van a ser todos los comandos que usaron la instrucción rm.

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

Una redirección consiste en trasladar la información de un tipo a otro, por ejemplo de la salida estándar a la entrada estándar o del error estándar a la salida estándar. Esto lo logramos usando el símbolo `>` para las **destructivas** y `>>` para las **no destructivas**.

En caso de las **destructivas**:

- Si el archivo destino no existe, se lo crea
- Si el archivo existe, se lo trunca y se escribe el nuevo contenido

En caso de las **no destructivas**:

- Si el archivo de destino no existe, se lo crea
- Si el archivo existe, se agrega la información al final

Y después tenemos a **los descriptores de ficheros**:

- 0 (Entrada estándar y normalmente el teclado)
- 1 (Salida estándar y normalmente la consola)
- 2 (Salida de error)



## Ejemplos

```
$ ls -l >fichero # guarda la salida de ls -l en fichero. (si no existe lo crea y sino lo sobrescribe)
$ ls -l 2>fichero # añade la salida del comando a fichero (si no existe lo crea y sino lo sobrescribe)
$ ls -l 2>fichero # si hay algun error, lo guarda en fichero
```

**Si no se especifica el descriptor del fichero se asume que se redirige la salida estándar.** En el caso del operador `<` se redirige la entrada estándar, es decir, el contenido del fichero que especificáramos, se pasaría como parámetro al comando.

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

El comando `kill` se usa para terminar un proceso manualmente. El comando envía una señal a un proceso para que termine (la señal es SIGTERM). Las señales disponibles tienen distintos nombres asignados a determinados números, el número de las señales puede cambiar entre distintas implementaciones de UNIX.

- SIGKILL suele tener el número 9
- SIGTERM el 15.

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

- |                        |   |
|------------------------|---|
| • <code>ps</code>      | Muestra la lista de procesos del usuario.         |
| • <code>kill</code>    | Mata un proceso.                                  |
| • <code>pstree</code>  | Muestra el árbol de procesos                      |
| • <code>killall</code> | Mata todos los procesos calculo                   |
| • <code>top</code>     | ps en la terminal, tipo texto                     |
| • <code>nice</code>    | Iniciar un proceso con una determinada prioridad. |

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

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

**Empaquetar** hacer referencia a agrupar en un solo fichero varios ficheros y/o directorios. QUE ES DISTINTO A COMPRIMIR, **comprimir** es reducir el tamaño de un fichero mediante el uso de un algoritmo de compresión.

En linux tenemos el comando `tar` que nos permite realizar el proceso de empaquetación.

### (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?

```
cd /home/user
tar -cfv paquete.tar ruta
-c # --create: Crea un nuevo archivo.
-x # --extract: Extrae ficheros de un archivo.
-v # --verbose: Lista detalladamente los ficheros procesados.
-f # \\$&fichero\\$&: Empaqueta o desempaqueta en o hacia un fichero.
-t # --list: Lista los contenidos de un archivo.
```

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

```
tar -cvzf edteam.tar.gz ruta
```

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

No. Los archivos deben comprimirse de uno en uno.

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

- `tar` Empaqueta o comprime archivos usando la extensión `.tar`
- `grep` Escribe en la salida estándar aquellas líneas que coincidas con un patrón.
- `gzip` Comprime sólo archivos utilizando la extensión `.gz`
- `zgrep` Busca archivos comprimidos para una expresión.
- `wc` Cuenta los caracteres, palabras y líneas del archivo de texto.

**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:**

---

- `.chmod 700 /etc / passwd`

intenta cambiar los permisos del archivo password a 700 pero no puede hacerlo ya que no posee root

- `.passwd root`

no puede modificar la contraseña para root

- `.rm prueba`

elimina el archivo "prueba"

- `.man /etc/shadow`

syntax error ya que man no debe recibir una ruta

- `.find /-name*.conf`

- `ls -l > prueba`

guarda lo que da el comando ls -l en el archivo "prueba"

- `.ps > prueba`

guarda lo que da el comando ps en el archivo "prueba"

- `.chmod 710 prueba`

cambia los permisos del archivo prueba haciendo que el creador posee todos los permisos, el grupo solo ejecución y los otros no tienen permisos

- `.chown root:root prueba`

se intenta cambiar el propietario del archivo prueba pero no posee permisos para hacerlo

- `.chmod 777 prueba`

intenta cambiar los permisos del archivo "prueba" para UGO dando todas las acciones

lista todos los archivos cuyos nombres terminan en .conf empezando la búsqueda en "/"

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

cambia la ruta del root a "/home/newroot" y bloquea la contraseña

- `.cd /root`

se intenta acceder a la carpeta root

- `.rm *`

borra todos los archivos del directorio

- `.cd /etc`

se accede a la carpeta /etc

- `.cp * /home -R`

copia todos los archivos de /etc a home pero el usuario no tiene el permiso de crear archivos en /home

- `.shutdown`

apaga el equipo

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

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

```
.kill -9 23
```

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

```
.kill -9 1 (el proceso init no puede terminarse)
```

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

```
.find /-name*.config
```

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

```
.ps > /home/user/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/user/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/user/yyyy
```

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

```
.rm /tmp/*
```

**(h) Cambiar el propietario del archivo /opt/isodata al usuarioiso2010**

```
.chown iso2017 /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/user/donde
```

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

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

```
.su root
```

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

```
useradd lballetero  
passwd 123456
```

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

Se modifico ( /home/password ), ( /home/shadow ) y se creo ( /home/user )

**(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
```

**(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 adibello:users /tmp/sursada2017
```

**(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 775 /tmp/cursada2017/ -R
```

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

```
screen su user
```

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

```
whoami  
#guami
```

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

```
ps
```

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

```
who
```

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

```
write user  
echo "El sistema va a ser apagado"
```

**(m) Apague el sistema.**

```
shutdown
```

## 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 180582  
cd 180582
```

**(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  
:wq
```

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

- Dueño: ningún permiso
- Grupo: permiso de ejecución

```
chmod 017 LEAME
```



- 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?.**

```
ls -l >> /home/ubuntu/leame
```

UNIX es caseSensitive, así que es la diferencia eso

**(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.**

```
#type para buscar solo archivos
sudo find /-name"archivo" -type f
sudo find /-name "*.jpg" -type f
```

**(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".**

```
sufo find /-name "*.so" >> /home/ubuntu/leame
```

**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 # crea el directorio ISO
cd . / i s o ; ps > f 0 # se posiciona en la carpeta /iso y guarda ps en el archivo f0
l s > f 1 # guarda en f1 todos los archivos del directorio actual
cd / # te posiciona en el directorio "/"
echo $HOME # imprime la ruta del usuario que esta ejecutando
l s -l $> $HOME/ i s o / l s # guarda en la carpeta ls lo que devuelve el ls -l
cd $HOME; mkdir f 2 #te posiciona en tu directorio y crea el directorio "f2"
l s -l d f 2 # lista los datos del directorio f2
chmod 341 f 2 # cambia los permisos del archivo f2 a dueño (escritura y ejecución), gr
upo (lectura) y otros(ejecucion)
touch d i r # crea un archivo llamado "dir"
cd f 2 # posiciona en el directorio f2
cd ~/ i s o # cambia el directorio del usuario "/iso"
pwd >f 3 #guarda la ruta donde se esta pocisionado en el archivo "f3"
ps | g rep ' ps ' | wc -l >> . . / f 2 / f 3
# guarda en el archivo "../f2/f3" las veces que esta la palabra "ps"
# en la lista de procesos
# {grep ps //devuelve las lineas que incluyen "ps"
# wc -l devuelve la cantidad de lineas}
chmod 700 . . / f 2 ; cd . .
# da permisos totales al dueño y deja sin permisos al resto
# y posiciona e la carpeta anterior a la posicionado
f i n d . -name e t c / passwd
f i n d / -name e t c / passwd
# imprime todos archivos que incluyan "etc" en su nombre en la carpeta /passwd
# sin ser admin no te permite ya que un usuario comun no puede buscar en raiz,
# solo el root
mkdir e j e r c i c i o 5
# crea el directorio "ejercicio5"
```

```
cp /home/letizia/iso/home/LETIZIAB -R # copiar el directorio de iso y su contenido
cp -n* /home/LETIZIB-R
```

**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:**



```
$ mkdir dir1
$ mkdir dir2
$ touch f1
$ touch f2
$ cd dir1
$ mkdir dir11
$ touch f3
$ cd .. # vuelvo para atras 2 directorios
$ cd dir2
$ touch f4
$ cd .. # vuelvo a usuario
```

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 /home/letizia/dir1/f3 /home/letizia
ls
ls /home/letizia
```

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

```
cd ../dir2
cp f4 ../dir1
cd ../dir1
ls
cd ../dir2
ls ../dir2
```

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

```
cp f4 /home/letizia/f7
cd ..
ls
```

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

```
mkdir copia
cp /home/letizia/dir1/* /home/letizia/copia -r
cd copia
ls
```

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

```
mv f1 archivo
ls
```

- 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
ls -l
```

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

```
mv f3 f3.exe
ls
cd dir2
touch f4
mv f4 f4.exe
ls
```

- Utilizando un único comando cambie los permisos de los dos archivos renombrados en el inciso anterior, de manera de reflejar lo siguiente:
  - Usuario: Ningún permiso
  - Grupo: Permisos de escritura
  - Otros: Permisos de escritura y ejecución

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

### 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.

```
mkdir /tmp/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 /tmp/logs/misLogs.tar /home/letizia
```

```
cp /tmp/logs/misLogs.tar.gz /home/letizia
```

(f) Elimine el directorio creado en 1, logs

```
rmdir /tmp/logs
```

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

```
tar -xvf misLogs.tar > /home/letizia/dir1
```

```
tar -xvzf misLogs.tar.gz >  
/home/letizia/dir2
```