
Trabajo Práctico 2 “Introducción a los sistemas operativos”

Facultad de Informática. UNLP.

Alumno: Gonzalez, Joaquín Manuel.

Fecha de realización: 11/09/2023.

Contenido

1. Editor de Textos:	3
2. Proceso de Arranque "System V"	5
3. Usuarios	13
4. FileSystem	17
5. Procesos	22
6. Otros comandos de Linux (Indique funcionalidad y parámetros)	26
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.....	28
8. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones.....	31
9. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones.....	32
10. Indique qué comando sería necesario ejecutar para realizar cada una de las siguientes acciones:	35
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:	37
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:	40
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)	41

1. Editor de Textos:

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

- Vi.
- Vim.
- Emacs.
- Joe.
- 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.

- La diferencia que presenta un Editor de Texto de los comandos “cat”, “more” o “less” es que son herramientas diferentes utilizadas para diferentes propósitos en el manejo de archivos de texto en sistemas Unix like. **Principales diferencias:**

- ❖ **Editor de Texto:** aplicación que te permite crear, modificar y guardar archivos de texto de manera interactiva.
- ❖ **“cat”:** Comando que se utiliza para mostrar el contenido de uno o varios archivos de texto en la salida estándar de consola. Se puede usar para ver el contenido completo de un archivo.
- ❖ **“more”:** Es un visualizador de texto paginado. Muestra el contenido de un archivo de manera incremental, una página a la vez.
- ❖ **“less”:** Es similar al “more” ya que se utiliza para ver el contenido de un archivo de texto, ya que permite avanzar y retroceder a través del archivo. También permite buscar texto y realizar otras acciones de navegación dentro del archivo.

Modos de operación que posee el editor Vi:

- ❖ Modo de Comandos.
- ❖ Modo de Inserción.
- ❖ Modo de Reemplazo.
- ❖ Modo de Visualización.

- ❖ Modo de Línea de Comandos.
- ❖ Modo de Bloque.
- ❖ Modo de Inserción de Línea.
- ❖ Modo de Comandos de Ex.

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

- **Guardar y Salir:**
 - ❖ “:w”: Guarda los cambios en el editor.
 - ❖ “:q”: Sale del editor si no hay cambios.
 - ❖ “:wq”: Guarda los cambios y sale del editor.
 - ❖ “:q!”: Sale del editor sin guardar cambios.
- **Navegación:**
 - ❖ “h”: Mueve el cursor a la izquierda.
 - ❖ “j”: Mueve el cursor hacia abajo.
 - ❖ “k”: Mueve el cursor hacia arriba.
 - ❖ “l”: Mueve el cursor a la derecha.
 - ❖ “G”: Mueve el cursor a la última línea del archivo.
 - ❖ “:n”: Mueve el cursor a la línea “n” siendo “n” un número positivo.
- **Edición:**
 - ❖ “i”: **Cambia al modo inserción.**
 - ❖ “dd”: Borra la línea completa.
 - ❖ “yy”: Copia la línea completa.
 - ❖ “p”: Pega el texto copiado o cortado.
- **Búsqueda:**
 - ❖ “:/patrón”: Busca el patrón en el archivo.
- **Deshacer y Rehacer:**
 - ❖ “u”: Deshace la última operación.
 - ❖ “Ctrl + r”: Rehace la última operación deshecha.
- **Modo Visual:**
 - ❖ “v”: **Cambia a modo visual.**
- **Ayuda:**
 - ❖ “help”: Abre la ayuda.
- **Pasaje a los demás modos:**
 - ❖ **Empezamos en Modo Comandos.**

- ❖ **"R"**: Pasamos a Modo de Reemplazo.
- ❖ **":"**: Pasamos a Modo de Línea de Comandos.
- ❖ **"Ctrl + v"**: Pasamos a Modo de Bloque.
- ❖ **"O"**: Inserta una línea encima de la línea actual.
- ❖ **"o"**: Inserta una línea debajo de la línea actual.
- ❖ **"Esc"**: Pasamos a Modo de Comandos.

2. Proceso de Arranque "System V"

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.

- **Pasos del proceso de Arranque:**
 - 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 file system raíz y se inicializan 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 del runlevel 1
 - 10) El final del runlevel 1 le indica que vaya al runlevel por defecto.
 - 11) Se ejecutan los scripts del runlevel por defecto.
 - 12) El sistema está listo para usarse.

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

- El proceso INIT tiene como objetivo llevar al sistema desde un estado de apagado o reinicio hasta un estado funcional completamente cargado y listo para que los usuarios y otros

procesos puedan interactuar con él, **cumpliendo con las siguientes funciones:**

- ❖ Cargar todos los subprocessos necesarios para el correcto funcionamiento del Sistema Operativo.
- ❖ Es el encargado de montar los filesystems y de hacer disponible los demás dispositivos.

Este proceso es ejecutado por el Kernel en procesos de arranque que usan System V.

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

- Al ejecutar ese comando se puede ver lo siguiente (fragmento):

```
joacoogonz@joacoogonz-VirtualBox:~$ ps tree
systemd--ModemManager--2*[{ModemManager}]
      |NetworkManager--2*[{NetworkManager}]
      |accounts-daemon--2*[{accounts-daemon}]
      |acpid
      |agetty
      |avahi-daemon--avahi-daemon
      |colord--2*[{colord}]
      |cron
      |csd-printer--2*[{csd-printer}]
      |cups-browsed--2*[{cups-browsed}]
      |cupsd
      |dbus-daemon
      |gnome-keyring-d--3*[{gnome-keyring-d}]
      |irqbalance--{irqbalance}
      |2*[kerneloops]
      |lightdm--Xorg--{Xorg}
```

Lo que se ve es la estructura jerárquica de los procesos en el sistema de forma gráfica. Muestra una representación visual del árbol de procesos.

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

- Los RunLevels son el modo en que arranca Linux, cada uno es responsable de levantar (iniciar) o bajar (parar) una serie de

servicios. Su objetivo principal es definir el estado en el que un sistema operativo se encuentra durante su arranque y el conjunto de servicios y entornos que están disponibles en ese estado. Cada runlevel tiene un propósito específico y configura el sistema para un uso particular.

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?**

- **Runlevels según el estándar:**
 - ❖ **Runlevel 0 (halt):** Este nivel se utiliza para apagar el sistema de manera segura. Todos los servicios se detienen y se apaga el hardware de manera adecuada.
 - ❖ **Runlevel 1 (Single User Mode):** Este nivel se utiliza para tareas de mantenimiento y recuperación del sistema. El sistema arranca en modo de usuario único con acceso mínimo.
 - ❖ **Runlevel 2 (Multiuser without NFS):** Se usa para iniciar el sistema en modo multiusuario sin habilitar la red. Es útil en situaciones de servidores locales.
 - ❖ **Runlevel 3 (Full Multiuser Mode Console):** Similar al Runlevel 2 pero con servicios de red habilitados.
 - ❖ **Runlevel 4 (No estándar):** Este nivel no se utiliza en muchas distribuciones y por lo tanto no tiene una configuración estándar.
 - ❖ **Runlevel 5 (X11 Modo Gráfico o Multiusuario con Interfaz Gráfica):** Este nivel incluye todos los servicios de Runlevel 3, también inicia un sistema de escritorio o entorno gráfico.
 - ❖ **Runlevel 6 (Reboot):** En este nivel, el sistema se reinicia.

La configuración del runlevel al iniciar un sistema operativo Unix like se encuentra definida en **“/etc/inittab”**.

No, no 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?**

- **Archivo “/etc/inittab”:**
 - ❖ **Finalidad:** Controla cómo se inicia el sistema operativo, qué servicios y procesos se inician automáticamente y en qué runlevels.
 - ❖ **Tipo de información que se almacena en el:** Contiene líneas de configuración que definen las acciones a realizar en diferentes situaciones en diferentes runlevels, y cómo manejar procesos específicos.
 - ❖ **Estructura de la información**
 - **Id:** Identifica la entrada en inittab (1 a 4 caracteres).
 - **Runlevels:** El/Los runlevels 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:** Establece el runlevel predeterminado.
 - ✓ **Ctrlaltdel:** Se ejecutará cuando init reciba la señal SIGINIT.
 - ✓ **Off, respawn (inicia el proceso especificado), once, sysinit, boot, bootwait, powerwait, etc.**
 - **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?

- El comando que utilizaría si estoy utilizando el estándar System V es:

```
joacogoonz@Debian:~$ sudo telinit X
```

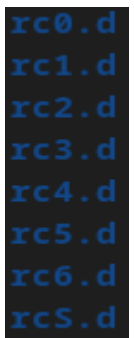
- ❖ **“sudo”:** Comando que permite a los usuarios ejecutar comandos con los privilegios de otro usuario, generalmente el superusuario (root).
- ❖ **“telinit”:** Comando que se utiliza para cambiar el runlevel del sistema. Normalmente requiere permisos de

superusuario por eso se utiliza junto con “sudo”. En este caso “X” debería cambiarse por el número del runlevel. Este cambio no es permanente. Al reiniciar el sistema este volverá al runlevel predeterminado que está configurado en “/etc/inittab”.

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

- **Scripts RC (Scripts de inicio):** Tienen como finalidad configurar y preparar el sistema operativo durante el proceso de arranque, se encargan de iniciar y detener servicios, configurar el entorno y realizar otras tareas necesarias para poner a funcionar al sistema. En sistemas que utilizan System V estos scripts en los directorios “/etc/rcX.d/”.

En System V el sistema determina qué scripts ejecutar en función del runlevel en el que se encuentra, el orden de ejecución de los mismos se determina por el nombre de los enlaces simbólicos a los archivos del “/etc/init.d” que están alocados en los directorios “rcX.d” asociados a cada runlevel. Los enlaces que comienzan con “S” indican que se debe iniciar el servicio con el argument start, y los que comienzan con “K” indican que se debe detener con el argument stop, **el número que sigue a cada letra indica el orden el que se ejecutan los scripts.** [S|K]<ORDEN><NOMBRE_SCRIPT>



```
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
rc6.d
rcS.d
```

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

- **Insserv:** Herramienta utilizada en sistemas operativos GNU/Linux que utilizan System V para administrar y organizar los Scripts RC, se utiliza para administrar el orden de los enlaces simbólicos del “/etc/rcX.d”, resolviendo las dependencias de forma automática. **Ventajas respecto un arranque tradicional:**

- ❖ **Gestión de Dependencias Automática:** Analiza los comentarios en los Scripts RC para determinar las dependencias entre servicios. Haciendo que los servicios se inicien en el orden correcto. Previendo de esta forma los conflictos y errores de inicio.
- ❖ **Facilita la Organización y Mantenimiento:** Al usar comentarios estructurados genera que los administradores puedan entender rápidamente las relaciones de dependencia y el orden de inicio de los servicios.
- ❖ **Adaptabilidad a Diferentes Runlevels:** Facilita la configuración de qué servicios deben ejecutarse en cada runlevel.
- ❖ **Simplificación de la Administración del Sistema:** Reduce la carga de trabajo de los administradores de sistemas debido a los puntos anteriores.

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

- Upstart fue el primer reemplazo de System V, este maneja el proceso de arranque a través de **eventos** (sucesos que desencadenan acciones, como por ejemplo el inicio de un servicio) y **trabajos/jobs** (scripts de texto plano que definen las acciones/tareas “unidades de trabajo” a ejecutar ante determinados eventos) definidos en “/etc/init” (.conf). **Los Jobs suelen ser de 2 tipos:**

- ❖ **Task:** Ejecución finita -> not respawning -> exit 0 o uso de stop.

- ❖ **Service:** Ejecución indeterminada -> respawning

Cada **job** posee un objetivo y un estado y en base a estos, ejecuta un proceso específico (al inicio, init emite el evento startup). También, los **Jobs** pueden tener una o varias tareas ejecutables durante su ciclo de vida y siempre ha de existir la tarea principal, estas tareas se definen mediante **exec o script ... end script**.

En Upstart el archivo /etc/inittab no existe más.

k) Cite las principales diferencias entre SystemV y Upstart.

- **Diferencias:**
 - ❖ System V se creó para operar en un entorno estático, **Upstart** se creó para operar en un entorno flexible.
 - ❖ **Upstart** permite la ejecución de trabajos en forma asincrónica mediante el uso de eventos, System V es estrictamente sincrónico.
 - ❖ **Upstart** hace uso de Jobs con sus propios objetivos y estados mientras que System V utiliza runlevels.

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

- Los reemplazos de los **Scripts RC** son los **Jobs** que se encuentran definidos en el **“/etc/init (.confg)”**

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-filesystems
          and runlevel [2345])
stop on runlevel [016]

[...]
exec /usr/sbin/mysqld
[...]
```

- **Referencias de cada línea:**
 - 1) **“# MySQL Service”**: Comentario que nos dice qué servicio se va a iniciar y el propósito del mismo.
 - 2) **“description ‘MySQL Server’”**: Descripción del servicio que va a aparecer en los registros del sistema.

- 3) **“autor ‘info autor’”**: Información sobre el autor del archivo de configuración.
- 4) **“start on (net-device-up and local-filesystems-and runlevel [2345])”**: El servicio debe iniciarse cuando se detecten dispositivos de red, los sistemas de archivos locales estén disponibles y el sistema esté en los runlevels 2, 3, 4 o 5.
- 5) **“stop on runlevel [016]”**: Cuando el sistema pase a alguno de los niveles especificados (Halt, Reboot o Single User Mode) debe detenerse.
- 6) **“[...]”**: Más líneas de configuración.
- 7) **“exec /usr/sbin/mysqld”**: Indica qué comando se debe ejecutar cuando se inicia el servicio. En este caso el demonio de MySQL “mysqld”.

n) ¿Qué es systemd?

- Es un sistema de inicialización y administración de servicios que centraliza la administración de demonios y librerías del sistema, actualmente es el estándar en la mayoría de distribuciones de GNU/Linux, es el reemplazo de System V y Upstart. Systemd está diseñado para optimizar el proceso de arranque y proporcionar un mejor control y administración de los servicios del sistema. Los runlevels son reemplazados por targets y al igual que en Upstart el archivo /etc/inittab no existe más y el demonio systemd reemplaza el proceso init -> este pasa a tener PID 1.

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

- Para hablar de sockets primero hay que hablar de unidades de trabajo denominadas **units** que son de los siguientes tipos:
 - ❖ **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) -> dependencias de unidades.

- ❖ **Snapshot:** Almacena el estado de un conjunto de unidades que puede ser restablecido más tarde (.snapshot).

Las units pueden tener dos estados **active o inactive**.

La **activación de socket** hace referencia a la característica que permite a los servicios esperar hasta que una conexión de red llegue a un puerto específico antes de activarse y procesar esa conexión produciendo un ahorro de recursos, respuesta rápida, escalabilidad y mayor seguridad.

p) ¿A qué hace referencia el concepto de cgroup?

- Permite organizar un grupo de procesos en forma jerárquica. Agrupa conjunto de procesos relacionados y las tareas que realiza son:
 - ❖ Tracking mediante subsistema cgroups -> no se utiliza el PID -> doble fork no funciona para escapar de systemd.
 - ❖ Limitar el uso de recursos.

3. Usuarios

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

- **Archivos utilizados:**
 - ❖ **“/etc/passwd”:** Contiene información básica sobre cada usuario (nombre de usuario, identificación de usuario, identificación de grupo, nombre completo, directorio de inicio y Shell por defecto).

```
joacogoonz@Debian:~$ cat /etc/passwd | grep "joacogoonz"
joacogoonz:x:1000:1000:joacogoonz,,,:/home/joacogoonz:/bin/bash
```
 - ❖ **“/etc/shadow”:** Contiene las contraseñas encriptadas de los usuarios, solo es accesible por el superusuario y proporciona una capa más de seguridad.

- ❖ **“/etc/group”**: Contiene información sobre los grupos de usuarios (nombre del grupo, identificación del grupo y una lista de miembros).

```
joacogoonz@Debian:~$ cat /etc/group | grep "joacogoonz"
users:x:100:joacogoonz
```

- ❖ **“/etc/gshadow”**: Similar a “/etc/shadow” pero para grupos.

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

- **UID**: User Identifier. Es un número único asociado a cada usuario en un sistema Unix like.
GID: Group Identifier. Es similar al UID pero se refiere a grupos en lugar de usuarios.
No pueden existir UIDs iguales en GNU/Linux esto se debe a que cada UID debe ser único, factor esencial para la correcta gestión de permisos y la identificación de los usuarios 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?.

- **Usuario Root**: Es el administrador del sistema o superusuario. Tiene privilegios administrativos completos y puede realizar cualquier operación en el sistema. Este usuario es esencial para la administración y configuración del sistema, puede detener el sistema, instalar software en el mismo, modificar o reconfigurar el kernel, drivers, etc.
Solo puede existir un usuario con ese perfil en GNU/Linux.
Su UID y GID es 0 (cero).

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.

- No lo voy a hacer en terminal pero dejo los pasos:
sudo useradd iso2023
sudo gropadd catedra
sudo usermod -a -G catedra iso2023
sudo -u iso2023 touch /home/iso2023/archive_propio.sh
sudo userdel iso2023

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

- **Comandos**
 - ❖ **“useradd ó adduser”**: Comando que se utiliza agregar un nuevo usuario al sistema, crea una nueva entrada de usuario en los archivos: **/etc/passwd, /etc/shadow, /etc/group y /etc/gshadow**. Requiere permisos de superusuario/sudo.
Parámetros:
 - **“-m”**: Crea un directorio de inicio para el usuario en **/home**.
 - **“-s Shell_a_usar”**: Especifica que Shell usará el Usuario.
 - **“-g grupo_a_asignar”**: Especifica el grupo inicial del usuario.
 - **“-G grupos_a_asignar”**: Asigna al usuario a grupos adicionales separados por comas.
 - **“-d home_dir”**: Especifica el directorio de inicio del usuario.
 - ❖ **“usermod”**: Permite hacer cambios en la configuración de un usuario sin necesidad de eliminar y recrear al mismo. Se

puede usar para cambiar el username, contraseña, añadir o eliminar grupos, modificar el directorio de inicio, cambiar el Shell, etc. Solo lo puede ejecutar un usuario **root**.

Parámetros:

- **“-c”**: Añade un comentario o información adicional al usuario.
 - **“-d”**: Cambia el directorio de inicio.
 - **“-g”**: Cambia el grupo primario del usuario.
 - **“-G”**: Añade o quita el usuario de grupos adicionales.
 - **“-l”**: Cambia el username.
 - **“-p”**: Cambia la contraseña del usuario (debe estar encriptada).
 - **“-L”**: Bloquea la contraseña del usuario poniendo el símbolo **“!”** al inicio de la contraseña encriptada del mismo haciéndola inaccesible.
- ❖ **“userdel”**: Se encarga de eliminar un usuario del sistema, así como su directorio de inicio y otros archivos asociados. No eliminará los archivos pertenecientes a ese usuario en otros lugares del sistema.

Parámetros:

- **“-r”**: Elimina también el directorio de inicio y el correo de usuario.
- ❖ **“su”**: Permite a un usuario cambiar a otra cuenta de usuario sin necesidad de cerrar y volver a iniciar sesión, es útil cuando se necesita realizar tareas como **root** o como usuario con otros permisos.

Parámetros:

- **“-”**: Cambias completamente de usuario, lo que incluye el entorno del nuevo usuario, el directorio de trabajo, variables de entorno, etc.
 - **“nombre_del_usuario”**: Especifica el nombre del usuario al que se quiere cambiar.
- ❖ **“groupadd”**: Agrega un nuevo grupo al sistema, que luego puede ser asignado a uno o más usuarios.

Parámetros:

- **“-g”**: Define el **GID** para el nuevo grupo, sino se especifica el sistema agrega uno único.
- **“-r”**: Crea un grupo del sistema.

- ❖ **“who”**: Proporciona detalles sobre los usuarios que tienen sesiones activas en el sistema en el momento que se ejecuta el comando.

Parámetros:

- **“-H”**: Muestra el header de las columnas de información.
- **“-q”**: Muestra solo el número total de usuarios conectados.
- **“-a”**: Muestra toda la información disponible sobre los usuarios.
- **“-u”**: Solo muestra nombres de usuarios.
- ❖ **“groupdel”**: Se encarga de eliminar un grupo específico del sistema, incluyendo su entrada en el archivo **/etc/group**.

Parámetros:

- **“-r”**: Elimina el grupo y el directorio asociado al mismo en **/home** si ha de existir.
- ❖ **“passwd”**: Proporciona una forma segura de cambiar las contraseñas de los usuarios en el sistema. Para cambiar la de otro usuario vas a necesitar permisos root.

Parámetros:

- **“nombre_de_usuario contraseña_nueva”**: Dice a qué usuario se le va a cambiar la contraseña.

4. FileSystem

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

- **Permisos que se definen en archivos:**

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

- ❖ **Lectura**: Permite al usuario leer el contenido de un archivo o directorio y listar su contenido con el comando **ls**.

- ❖ **Escritura:** Permite la escritura, modificación y eliminación, en caso de directorios, la escritura permite crear nuevos archivos o borrar archivos existentes.
- ❖ **Ejecución:** Permite a un usuario ejecutar el archivo si este es un programa, en caso de directorio, permite entrar al mismo usando el comando **cd**.

A la hora de interpretar los permisos de un archivo es importante analizarlos bien, supongamos tener la siguiente cadena de permisos **-rwx-w-rw-** podemos ver lo siguiente:

- ❖ **-**: Nos indica el tipo del archivo, si fuese una **"d"** sería un directorio.
- ❖ **rwx**: Son los permisos para el propietario del archivo.
- ❖ **-w-**: Son los permisos para miembros del grupo al que pertenece el archivo.
- ❖ **rw-**: Son los permisos para los demás usuarios del sistema.

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

- **Comandos:**

- ❖ **"chmod"**: Se utiliza para cambiar los permisos de acceso a archivos y directorios (**Lectura, Escritura, Ejecución**).

Parámetros:

- **"-R"**: Cambia los permisos recursivamente dentro de la ruta especificada, es decir, lo hace para todos los archivos y subdirectorios dentro del directorio especificado.
- **Modos Numéricos:** Especifican los permisos utilizando los números, por ejemplo, el **755** establece permisos de **lectura, escritura y ejecución** para el **propietario**, y solo permisos de **lectura y ejecución** para el **grupo y otros**.

Valores numéricos:

- ✓ **4(r--)**: Otorga solo permiso de lectura.
- ✓ **2(-w-)**: Otorga solo permiso de escritura.
- ✓ **1(--x)**: Otorga solo permiso de ejecución.

- ✓ **0(---)**: No se concede permiso alguno.
- ✓ El primer número hace referencia al dueño **7 (4 + 2 + 1) "U"**, el segundo número referencia al grupo **5 (4 + 1) "G"**, y el tercero a los demás usuarios **5 (4 + 1) "O"**.

➤ **Modos Simbólicos**: Permite especificar el cambio de permisos de una manera más legible y comprensible, por ejemplo, **chmod u+r archivo** añade permisos de **lectura** para el propietario.

Tipos de Permisos:

- ✓ **"u"**: Para el propietario.
- ✓ **"g"**: Para el grupo al que pertenece el archivo.
- ✓ **"o"**: Para el resto de usuarios.
- ✓ **"a"**: Para todos los usuarios.

Operadores:

- ✓ **"+"**: Agrega permisos.
- ✓ **"-"**: Quita permisos.
- ✓ **"="**: Establece permisos específicos.

❖ **"chown"**: Se utiliza para cambiar el propietario y el grupo de un archivo o directorio. Solo puede ejecutar este comando el **usuario root**.

Parámetros:

- **"usuario[:grupo] ruta_archivo/dir"**: Especifica el nuevo propietario y, opcionalmente, el nuevo grupo. Si no se proporciona un grupo, queda igual.
- **"-R"**: Cambia el propietario y el grupo recursivamente dentro de la ruta especificada, es decir, lo hace para todos los archivos y subdirectorios dentro del directorio especificado.

❖ **"chgrp"**: Permite a los usuarios y roots cambiar el grupo al que pertenece un archivo o directorio.

Parámetros:

- **"grupo"**: Especifica el nombre del grupo destino.

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?

- RESPONDIDO EN LA PREGUNTA ANTERIOR. VER COMANDO “chmod” MODOS NUMÉRICOS.

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.

- Sí, existe la posibilidad de que algún usuario del sistema pueda acceder a determinado archivo para el cual no posee permisos, este sería el caso del usuario **root** ya que este último puede acceder y modificar cualquier archivo o directorio. Para verlo mejor cree un archivo en mi escritorio, lo llamé “prueba.sh” y mientras tenía permisos le agregue un texto, luego le saqué todos los permisos y paso lo siguiente:

```
joacogoonz@Debian:~/Desktop$ chmod 0 prueba.sh
joacogoonz@Debian:~/Desktop$ ls -l prueba.sh
----- 1 joacogoonz joacogoonz 21 Sep  8 23:55 prueba.sh
joacogoonz@Debian:~/Desktop$ sudo cat prueba.sh
Accedí sin permisos
```

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

- Estos dos conceptos son formas de especificar la ubicación de un archivo o directorio en un file system.

❖ “full path name”: Especifica la ubicación del archivo o directorio desde la raíz del file system “/”. Incluye todo el camino necesario para llegar al destino en orden.

```
joacogoonz@Debian:~/Desktop$ cat /home/joacogoonz/Desktop/prueba.sh
```

❖ “relative path name”: Especifica la ubicación de un archivo o directorio en relación con el directorio actual en donde estás parado.

```
joacogoonz@Debian:~/Desktop$ cat ./prueba.sh
```

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.**

- Puedo determinar en qué directorio me encuentro actualmente utilizando el comando **pwd**. Podes ingresar a tu directorio personal sin necesidad de escribir todo el path usando **cd** ó **cd ~**. Se puede usar ese mismo comando para acceder a otros directorios haciendo uso de **relative paths**.

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

- **Comandos:**

- ❖ **“du”**: Herramienta utilizada para estimar el espacio utilizado por directorios y archivos, por defecto muestra esta información en Kilobytes.

- Parámetros:**

- **“-h”**: Muestra los tamaños en formato más legible, usando Megabytes y Gigabytes.
 - **“-a”**: Muestra el tamaño de todos los archivos y directorios incluyendo los ocultos.

- ❖ **“rmdir”**: Comando utilizado para eliminar directorios **vacíos**, es decir, que no contienen ningún archivo o subdirectorio.

- Parámetros:**

- **“-p”**: Indica que se tiene que eliminar el directorio especificado y, de ser necesario, los directorios padres que se hayan convertido en vacíos después de realizar la eliminación del directorio.
 - **“-v”**: Muestra mensajes detallados sobre las operaciones que está realizando.

- ❖ **“ln”**: Se usa para crear enlaces entre archivos, estos enlaces son referencias a un archivo desde uno o más lugares en el file system.

- Tipos de Enlaces:**

- **Hard link:** Es una referencia directa al contenido del archivo en el file system. El archivo tiene múltiples nombres pero ocupa el mismo espacio.
- **Soft Link:** Referencia indirecta que puede apuntar a archivos o directorios en cualquier parte del sistema.

Parámetros:

- **“-s”:** Se usa para crear soft links en vez de hard links.
- ❖ **“cp”:** Duplica archivos y/o directorios, puede copiar unos o varios archivos a un directorio específico o también puede copiar un directorio entero. **cp [opciones] origen destino**

Parámetros:

- **“-i”:** Pide confirmación antes de sobrescribir un archivo existente.
- **“-u”:** Copia solo cuando el archivo de origen es más reciente que el destino o si este no existe.
- **“-a”:** Copia todos los archivos y subdirectorios de manera recursiva manteniendo todas las propiedades y atributos originales.
- ❖ **“mv”:** Cambia la ubicación de un archivo o directorio y también se utiliza para cambiar el nombre de los mismos.

Parámetros:

- **“-i”:** Pide confirmación antes de sobrescribir un archivo existente.
- **“-u”:** Mueve solo cuando el archivo de origen es más reciente que el destino o si este no existe.
- **Para únicamente cambiar el nombre del archivo** solo se debe especificar **el nombre viejo y el nombre nuevo, no hay que usar una ruta de destino.**
- ❖ **LOS DEMÁS FUERON DEFINIDOS EN LA PRÁCTICA 1.**

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.

- **Proceso:** Es la instancia de ejecución de un programa de software, cada uno de estos procesos tiene su propio espacio de memoria y recursos asignados, y pueden estar en ejecución en paralelo con otros procesos. Tienen un program counter, un ciclo de vida y pasan por varios estados, además poseen un stack para ellos y uno para el kernel.

PID (Process ID): Identificador único de un proceso en el sistema operativo, estos se asignan secuencialmente.

PPID (Parent Process ID): Se refiere al PID del proceso padre que creó/inició el proceso actual.

Todos los procesos tienen un PID y un PPID.

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

- **Comandos:**
 - ❖ **“ps”:** Muestra información sobre los procesos en ejecución.
Parámetros:
 - **“-e”:** Muestra todos los procesos del sistema.
 - **“-aux”:** Proporciona una vista detallada de todos los procesos del sistema.
 - ❖ **“top”:** Herramienta interactiva que proporciona una actualización en tiempo real de los procesos que están usando la CPU y la memoria.
 - ❖ **“pstree”:** Representación gráfica de la jerarquía de procesos.

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

- **Background (Segundo Plano):** El proceso que se ejecuta no está directamente interactuando con el usuario a través de la terminal. Puede ejecutarse en paralelo y no bloquea la terminal.
- **Foreground (Primer Plano):** Todo lo contrario al anterior, tiene el control sobre la interfaz de la terminal, la bloquea y requiere que el usuario interactúe directamente.

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

- Para ejecutar un proceso en segundo plano solo necesitamos utilizar el símbolo “&” luego de especificar un comando en la terminal. Para pasar un proceso **de foreground a background** solo debemos de ejecutar el comando “**bg %n**” siendo n el PID y **de background a foreground** “**fg %n**” siendo n el PID.

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

- **Pipe (|)**: Permite la comunicación entre procesos al redirigir la salida estándar de un proceso hacia la entrada estándar de otro.

Finalidades:

- ❖ **Procesamiento en tiempo real**: Permite el procesamiento en tiempo real de la salida de un comando antes de ser utilizada por otro.
- ❖ **Combinación de comandos**: Facilita la comunicación entre comandos para lograr tareas más complejas.

Contar las líneas de texto en un archivo:

```
joacogoonz@Debian:~/Desktop$ cat prueba_pipe.sh | wc -l
12
```

Concatenar dos archivos y comprenderlos en uno nuevo:

```
joacogoonz@Debian:~/Desktop$ cat prueba_pipe.sh prueba.sh | gzip pruebas.sh
```

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

- **Tipos de redirecciones:**
 - ❖ **Redirección de salida (>)**: Redirige la salida estándar de un comando hacia un archivos, si el archivo existe lo sobrescribe, sino, lo crea. **Comando > archivo**
 - ❖ **Redirección de salida añadiendo (>>)**: Similar a > solo que no sobrescribe, añade la salida a la última línea del archivo. **Comando >> archivo**

- ❖ **Redirección de entrada (<)**: Toma un archivo como entrada y lo pasa como entrada estándar para el comando. **Comando < archivo**
- ❖ **Redirección de línea de entrada (<<)**: Permite ingresar múltiples líneas de texto desde la terminal como entrada estándar de un comando. **Comando << líneas_de_texto**

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

- **“kill”**: Se utiliza para enviar señales a procesos en ejecución, permite la interacción con los procesos, como terminarlos de manera controlada o enviarles señales específicas. Requiere el uso del PID.

Señales que envía:

- ❖ **SIGTERM**: Si solo se especifica el PID del proceso a terminar, se envía esta señal que es una petición suave para terminar el proceso.
- ❖ **SIGKILL**: Si se utiliza el comando con el parámetro **“-9 PID”** se envía una señal fuerte de terminación que hará que se termine el proceso de inmediato sin darle chance de realizar acciones de limpieza o un cierre ordenado.

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

- **Comandos:**
 - ❖ **“killall”**: Permite terminar procesos basados en sus nombres, a diferencia de **kill** que requiere del PID.
- Parámetros:**
- **“-u”**: Especificar el nombre del usuario cuyos procesos se deben terminar.
 - **“-s”**: Especificar la señal que se enviará a los procesos.
 - **“-l”**: Lista todas las señales disponibles.
 - **“-g”**: Termina todos los procesos pertenecientes a un grupo de procesos.

- ❖ **“nice”**: Se utiliza para modificar la prioridad de ejecución de un proceso, esto puede afectar cuánto tiempo de CPU se le asigna en relación a otros procesos.

Parámetros:

- **“-n nivel comando”**: Especificar nivel de prioridad del proceso, si no se pone ningún valor en “nivel”, nos mostrará el nivel actual del comando especificado.

- ❖ **LOS DEMÁS COMANDOS FUERON EXPLICADOS ANTERIORMENTE.**

6. 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 se refiere a la acción de unir múltiples archivos y/o directorios en un solo archivo comprimido, esto lo logramos usando **tar**.

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?

- No voy a realizar el ejemplo pero la observación que se debería notar es que el tamaño del archivo empaquetado debería ser menor que la suma de los tamaños individuales de los cuatro archivos, ya que estos están comprimidos y optimizados para ahorrar espacio.

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

```
joacogoonz@Debian:~/Desktop$ tar -cvf prueba-tamano.tar prueba.sh prueba_pipe.sh tamanio.sh compactacion.sh
prueba.sh
prueba_pipe.sh
tamanio.sh
compactacion.sh
joacogoonz@Debian:~/Desktop$ gzip prueba-tamano.tar
```

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

- Sí, esto se ve en el ejemplo anterior, la única restricción es que todos los archivos a comprimir deben de estar en el mismo directorio.

e) Investigue la funcionalidad de los siguientes comandos:

- **Comandos:**

- ❖ **“tar”**: Comando que se utiliza para el empaquetado de uno o más archivos en un solo archivo que por convención tiene como extensión **“.tar”**

Opciones comunes:

- **“-c”**: Crea un nuevo archivo empaquetado.
- **“-x”**: Desempaqueta archivos de un archivo empaquetado.
- **“-v”**: Muestra información detallada del proceso.
- **“-f” archivo**: Especifica el nombre del archivo empaquetador a crear o empujar.
- **“-t”**: Muestra el contenido del archivo empaquetado sin extraerlo.
- **“-z”**: Además de empaquetar, comprime usando **gzip**.
- **Se combinan “-cvf” para crear y “-xvf” para extraer.**

- ❖ **“grep”**: Herramienta poderosa para buscar patrones de texto en archivos o en la salida de otros comandos.

Opciones comunes:

- **“-i”**: Ignora la diferencia entre mayúsculas y minúsculas.
- **“-c”**: Muestra solo el recuento de coincidencias.
- **“-v”**: Invierte la búsqueda para mostrar líneas que no coincidan con el patrón.

Sintaxis: `grep [opciones] patrón [archivo(s)]`

- ❖ **“gzip”**: Herramienta de compresión de archivos. Su función principal es reducir el tamaño de los archivos para ahorrar espacio en disco y acelerar la transferencia de archivos a través de la red. Por convención comprimimos los archivos con la extensión **“.gz”**.
- ❖ **“zgrep”**: Comando que combina la funcionalidad de **grep** con la capacidad de trabajar con archivos comprimidos en formato **gzip (.gz)**. Permite buscar patrones de texto dentro de archivos comprimidos sin necesidad de descomprimirlos previamente.
- ❖ **“wc”**: Se utiliza para contar líneas, palabras y caracteres en un archivo o en la entrada estándar.

Parámetros:

- **“-l”**: Cuenta el número de líneas en un archivo.
- **“-w”**: Cuenta el número de palabras en un archivo.
- **“-c”**: Cuenta el número de caracteres en un archivo.

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

```
ls -l > prueba
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
```

- **Análisis:**

- ❖ **ls -l > prueba:** Genera un listado de todos los directorios y archivos del directorio de inicio del usuario de forma detallada y lo guarda en un archivo llamado **prueba**.
- ❖ **ps > PRUEBA:** Guarda un listado con la información de los procesos que se están ejecutando actualmente en un nuevo archivo llamado **PRUEBA**.
- ❖ **chmod 710 prueba:** Cambia los permisos de acceso al archivo **prueba**, le da permisos de lectura, escritura y ejecución al propietario, permiso de ejecución a los participantes de su grupo y cero permisos a los demás usuarios.
- ❖ **chown root : root PRUEBA:** Intenta cambiar el propietario y el grupo al que pertenece el archivo **PRUEBA** pero no puede porque no es un usuario **root**.
- ❖ **chmod 777 PRUEBA:** Cambia los permisos de acceso al archivo **PRUEBA**, le da permisos de lectura, escritura y ejecución al propietario, a los participantes de su grupo y a los demás usuarios.
- ❖ **chmod 700 /etc/passwd:** El usuario intenta cambiar los permisos de lectura, escritura y ejecución al directorio **/etc/passwd** pero esta operación no es posible porque solo un usuario **root** puede administrar los permisos de los directorios almacenados en **/etc**.

- ❖ **passwd root:** El usuario intenta ver o cambiar la contraseña del usuario **root** pero esta acción no está permitida porque solo un usuario **root** puede cambiar la contraseña de otros usuarios.
- ❖ **rm PRUEBA:** Se elimina el archivo PRUEBA.
- ❖ **man /etc/shadow:** Esta acción no se puede realizar ya que el comando **man** no puede brindar un acceso a las páginas del manual del sistema ya que estas brindan información detallada sobre los comandos, utilidades, funciones del sistema y archivos del sistema operativo, no sobre directorios, además el permiso es denegado ya que intenta acceder al directorio **shadow** el cual contiene las contraseñas encriptadas de los usuarios.
- ❖ **find / -name *.conf:** Busca y lista a partir del directorio raíz todos los archivos y directorios que contengan la extensión **.conf**.
- ❖ **usermod root -d /home/newroot -L:** El Usuario intenta cambiar el directorio de inicio del Usuario **root** a el directorio **/home/newroot** y además intenta bloquear su contraseña, esta acción no está permitida ya que el usuario no posee los permisos necesarios para ejecutar el comando.
- ❖ **cd /root:** El usuario intenta acceder al directorio de inicio del usuario **root**, algo no posible ya que el usuario no posee los permisos necesarios.
- ❖ **rm *:** Se eliminan todos los archivos en el directorio donde está posicionado el usuario actualmente.
- ❖ **cd /etc:** El usuario se sitúa en el directorio de configuraciones del sistema.
- ❖ **cp * /home -R:** El usuario intenta copiar todos los archivos y directorios del directorio actual **/etc** al directorio **/home**, incluyendo subdirectorios de manera recursiva pero esto no es posible ya que solo los usuarios **root** tienen permisos para copiar el contenido del directorio **/etc**.
- ❖ **shutdown:** El usuario apaga el sistema. Se necesitan permisos **sudo** para hacerlo.

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

a) Terminar el proceso con PID 23.

- Se podría utilizar los comandos **“kill 23”** ó **“kill -9 23”**.

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

- El proceso **init** tiene el PID 1 ya que este es el proceso de arranque del sistema, pero si no se supiera esta información se podría realizar el comando **“ps -aux”** para ver información detallada de todos los procesos que se están ejecutando actualmente y en la primer línea de la salida del comando se verá el PID del proceso **init**, luego se podría intentar usar el comando **“kill -9 1”** pero no se obtendría ningún resultado ya que el proceso **init** es de suma importancia para el arranque del sistema y este no puede ser terminado.

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

- Se podría utilizar el comando **“find / -type f -name *.conf”** este realizará una búsqueda desde el directorio raíz de todos los archivos que contengan la cadena “.conf” en su nombre, solo buscará archivos ya que este fue el tipo especificado para la búsqueda.

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

- Se podría utilizar el comando **“ps >/home/usuario/procesos”**.

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

- **Usuario:** Lectura, escritura, ejecución.
- **Grupo:** Lectura, ejecución.
- **Otros:** ejecución.

- Se podría utilizar el comando **“chmod 751 /home/usuario/xxxx”**.
- f) Cambiar los permisos del archivo /home/<su nombre de usuario>/yyyy a:**
- Usuario: Lectura, escritura.
 - Grupo: Lectura, ejecución.
 - Otros: Ninguno.
- Se podría utilizar el comando **“chmod 650 /home/usuario/yyyy”**.
- g) Borrar todos los archivos del directorio /tmp**
- Se deberían de usar los comandos **“cd /tmp”** y luego **“rm *”**.
- h) Cambiar el propietario del archivo /opt/isodata al usuario iso2021**
- Se podría utilizar el comando **“sudo chown iso2021 /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.**
- Se podría utilizar el comando **“pwd >> /home/usuario/donde”**.

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

- a) Ingrese al sistema como usuario “root”**
- Se podría realizar usando el comando **“su”** ó **“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.**

- Comandos:
 - ❖ `sudo adduser jgonzalez.`
 - ❖ `sudo passwd jgonzalez 12345.`

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

- Se modificaron los archivos `/etc/passwd`, `/etc/shadow`, `/etc/group`, `/etc/gshadow` y se creó el directorio de inicio del nuevo usuario en `/home/jgonzalez`.

- d) Crear un directorio en `/tmp` llamado `cursada2021`

- Comandos:
 - ❖ `cd /tmp.`
 - ❖ `mkdir cursada2023.`

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

- Comandos:
 - ❖ `sudo cp /var/log/* > /tmp/cursada2023.`

- 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`.

- Se puede realizar con el comando `“sudo chown -R jgonzalez:users /tmp/cursada2023”`.

- 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.

- Se puede realizar con el comando `“chmod 723 /ruta_dir”`.

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

- Se puede realizar con el comando **“sudo login jgonzalez”**. Luego de realizar el comando login se nos pedirá la contraseña del usuario al que queramos cambiar, ver que solo se puede realizar esta acción con permisos **sudo**.
- i) **Una vez logueado con el usuario antes creado, averigüe cuál es el nombre de su terminal.**
- Se puede realizar con el comando **“tty”** que nos brinda el nombre del archivo asociado con la terminal que estamos usando o se puede también usar el comando **“ps -p 544”**.
- j) **Verifique la cantidad de procesos activos que hay en el sistema.**
- Se puede realizar con el comando **“ps -aux | wc -l”**.
- k) **Verifiqué la cantidad de usuarios conectados al sistema.**
- Se puede realizar con el comando **“who -q”**.
- l) **Vuelva a la terminal del usuario root, y envíele un mensaje al usuario anteriormente creado, avisándole que el sistema va a ser apagado.**
- **Comandos:**
 - ❖ **su.**
 - ❖ **wall “El sistema está por apagarse”**. El comando **wall** sirve para enviar un mensaje a todos los usuarios que están actualmente conectados al sistema, se necesita permisos **sudo** para usarlo.
- m) **Apague el sistema.**
- **“sudo shutdown -h now”**.

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 legajo”**. La acción no se puede realizar porque el sistema confunde la **“/”** con un desplazamiento de directorio. Sin la barra se puede crear.

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"**.

- **Comandos:**

- ❖ **“vi LEAME”**.

- ❖ Dentro del editor cambiar a modo de inserción usando la tecla **“i”**, escribir toda la información en el archivo y luego volver al modo comando usando la tecla **“Esc”**.

- ❖ Una vez en el modo comando escribir el comando **“: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.**
- **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?.

- **Comandos:**

- ❖ "cd /etc".

- ❖ "ls > /home/leame".

Esta acción se puede realizar debido a que GNU/Linux es case sensitive.

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.

- **Comandos:**

- ❖ Para buscar un solo archivo con find: "find / -name nombre_del_archivo".

- ❖ Para buscar archivos con características similares: "find / -name "*cadena_a_contener".

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".

- **Comando:** "find / -name "*.so" > /home/user/legajo/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 iso
cd ./iso; ps > f0
ls > f1
cd /
echo $HOME
ls -l && $HOME/iso/ls
cd $HOME; mkdir f2
ls -ld f2
chmod 341 f2
touch dir
cd f2
cd ~/iso
pwd >f3
ps | grep 'ps' | wc -l >> ../f2/f3
chmod 700 ../f2; cd ..
find . -name etc/passwd
find / -name etc/passwd
mkdir 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.

- **Acciones:**

- ❖ “`mkdir iso`”: Crea el directorio “iso” en su directorio inicial.

- ❖ **"cd ./iso ; ps > f0"**: Se sitúa en el directorio **"iso"** y crea el archivo **"f0"** con un listado de todos los procesos en ejecución en el sistema actualmente.
- ❖ **"ls > f1"**: Crea un archivo llamado **"f1"** en el directorio **"iso"** que contiene la lista de archivos y/o subdirectorios del directorio **"iso"**.
- ❖ **"cd /"**: Se sitúa en el **directorio raíz**.
- ❖ **"echo \$HOME"**: Muestra en la salida estándar la dirección del directorio inicial del usuario.
- ❖ **"ls -l &> \$HOME/iso/ls"**: Se crea un nuevo archivo llamado **"ls"** en el directorio **"iso"** ubicado en el directorio inicial del usuario que va a tener un listado detallado de los archivos y/o directorios del directorio raíz y además también la salida de error del comando realizado **"ls -l"**, esto se logra sumando a la redirección el símbolo **"&"**.
- ❖ **"cd \$HOME ; mkdir f2"**: Se sitúa en su directorio inicial y crea el directorio **"f2"**.
- ❖ **"ls -ld f2"**: Muestra información detallada del directorio **"f2"**. Se especifica que se trate al mismo como un directorio con el parámetro **"d"**.
- ❖ **"chmod 341 f2"**: Cambia los permisos de lectura, escritura y ejecución del directorio **"f2"**. Escritura y Ejecución para el dueño, Lectura para el grupo y Ejecución para los demás usuarios.
- ❖ **"touch dir"**: Crea un archivo llamado **"dir"** en el directorio inicial del usuario.
- ❖ **"cd f2"**: Se sitúa en el directorio **"f2"**.
- ❖ **"cd ~/iso"**: Se sitúa en el directorio **"iso"**.
- ❖ **"pwd > f3"**: Crea un archivo nuevo en el directorio **"iso"** que contiene la información de la ruta actual del usuario en ese momento **"~/iso"**.
- ❖ **"ps | grep " ps " | wc -l >> ../f2/f3"**: Se listan todos los procesos en ejecución y el **"grep"** recibe esa lista como entrada estándar, allí filtra los resultados por aquellos procesos que posean la cadena **"ps"**, ese nuevo listado filtrado es pasado como entrada estándar al comando **"wc"** el cuál cuenta la cantidad de líneas de ese listado, es decir, la cantidad de procesos que se encontraron con la cadena **"ps"** para luego

redirigir esa salida al archivo **"f3"** añadiendo la cantidad al final del mismo.

- ❖ **"chmod 700 ../f2 ; cd .."**: Se modifican los permisos de lectura, escritura y ejecución del directorio **"f2"**. Lectura, Escritura y Ejecución para el dueño y cero permisos para el grupo y otros usuarios. Luego al ejecutar **"cd .."** escala un nivel hacia arriba en la estructura de directorios, ahora situándose nuevamente en su directorio inicial.
- ❖ **"find . -name etc/passwd"**: Se muestra un mensaje de warning por el mal uso del comando **"find"**.
- ❖ **"find / -name etc/passwd"**: Muestra en pantalla el listado de los archivos y/o directorios de **"/etc/passwd"** ordenados por nombre.
- ❖ **"mkdir ejercicio5"**: Crea el directorio **"ejercicio5"**.

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 coméntelas en el archivo creado en el inciso a).



12. Cree una estructura desde el directorio /home que incluya varios directorios, subdirectorios y archivos, según el esquema siguiente. Asuma que “usuario” indica cuál es su nombre de usuario. Además deberá tener en cuenta que dirX hace referencia a directorios y fX hace referencia a archivos:

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".
- Haga lo mismo que en el inciso anterior pero el archivo de destino, se debe llamar "f7".
- Cree el directorio “copia” dentro del directorio usuario y copie en él, el contenido de "dir1".
- Renombre el archivo "f1" por el nombre “archivo” y vea los permisos del mismo.
- 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.
- Renombre los archivos "f3" y "f4" de manera que se llamen "f3.exe" y "f4.exe" respectivamente.
- 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.

- **Comandos:**
 - ❖ "mv f3 \$HOME".
 - ❖ "cp f4 \$HOME/dir1/dir11".
 - ❖ "cp f4 \$HOME/dir1/dir11/f7".
 - ❖ "mkdir copia ; cp -a dir1 copia".
 - ❖ "mv f1 archivo ; ls -ld archivo".
 - ❖ "chmod 617 archivo".
 - ❖ "mv f3 f3.exe ; mv f4 f4.exe".
 - ❖ "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.
 - "sudo mkdir /tmp/logs".
- b) Copie todo el contenido del directorio /var/log en el directorio creado en el punto anterior.
 - "sudo cp -a /var/log/* /tmp/logs".
- c) Empaquete el directorio creado en 1, el archivo resultante se debe llamar "misLogs.tar".
 - "tar -cvf misLogs.tar logs".
- d) Empaquete y comprima el directorio creado en 1, el archivo resultante se debe llamar "misLogs.tar.gz".
 - "tar -cvfz misLogs.tar.gz logs".
- e) Copie los archivos creados en 3 y 4 al directorio de trabajo de su usuario

- **"cp misLogs.tar logs misLogs.tar.gz logs \$HOME".**

f) Elimine el directorio creado en 1, logs.

- **"rm -r logs".**

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

- **"tar xvf misLogs.tar -C 1"**
"tar xvfz misLogs.tar.gz -C 2"