

1. Editor de textos:

(a) Nombre al menos 3 editores de texto que puede utilizar desde la línea de comandos.
VIM (vi (archivo)), NANO (nano (archivo)), para usar sus comandos apretar ctrl) y EMACS (emacs (archivo)).

(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 editores de texto tienen como propósito proporcionar una interfaz interactiva con la cual se puede crear, modificar y guardar archivos de texto, en cambio los comandos cat, more y less tienen como propósito mostrar el contenido de un archivo de texto.

Vi posee 3 modos de ejecución:

- Modo Insert (Ins o i).
- Modo Visual (v).
- Modo de Órdenes o Normal (Esc).

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

- **w**: escribir cambios
- **q o q!**: salir del editor
- **dd**: cortar
- **y**: copiar al portapapeles
- **p**: pegar desde el portapapeles
- **u**: deshacer
- **/frase**: busca "frase" dentro del archivo

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

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

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

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

- 1) Su función es cargar todos los subprocesos necesarios para el correcto funcionamiento del SO
- 2) El proceso init posee el PID 1 y se encuentra en /sbin/init
- 3) En SysV se lo configura a través del archivo /etc/inittab
- 4) No tiene padre y es el padre de todos los procesos (pstree)
- 5) Es el encargado de montar los filesystems y de hacer disponible los demás dispositivos

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

Los runlevels (niveles de ejecución) son estados predefinidos en los que un sistema operativo basado en Unix o Linux puede estar, y cada runlevel define qué servicios y procesos están activos en el sistema en un momento dado.

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

Según el estándar:

- 0: halt (parada).
- 1: single user mode (monousuario).
- 2: multiuser, without NFS (modo multiusuario sin soporte de red).
- 3: full multiuser mode console (modo multiusuario completo por consola).
- 4: no se utiliza.
- 5: X11 (modo multiusuario completo con login gráfico basado en X).
- 6: reboot.

Los runlevels se encuentran en el directorio `/etc/init.d`.

No todas las distribuciones respetan estos estándares.

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

Es un archivo de configuración utilizado en sistemas Unix y Linux tradicionales para definir cómo debía comportarse el sistema en los diferentes runlevels.

Se divide en 4 pasos: `id:runlevels:action:process`.

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

Se utiliza el comando que se utiliza es **sudo telinit** o **sudo init** seguido del número del runlevel al que se desea cambiar. El cambio no es permanente, solo se mantiene durante esa sesión.

- (g) Scripts RC. ¿Cuál es su finalidad? ¿Dónde se almacenan? Cuando un sistema GNU/Linux arranca o se detiene se ejecutan scripts, indique cómo determina qué script ejecutar ante cada acción. ¿Existe un orden para llamarlos? Justifique.

- Son scripts de shell que se ejecutan durante el proceso de inicio y apagado de un sistema GNU/Linux basado en SysV init. Su función es inicializar y detener servicios, así como realizar otras tareas de configuración necesarias para que el sistema entre en un runlevel específico.
- Se almacena en el directorio: **`/etc/init.d/`**
- Linux determina que runlevel ejecuta mediante scripts específicos almacenados en los directorios **`/etc/rcX.d/`**, donde **X** es el número del runlevel al que el sistema está cambiando.
- El orden de ejecución de los scripts está determinado por los números que siguen a las letras **S** o **K**. Los números indican la prioridad de ejecución, siendo los números más bajos los ejecutados primero.

3. SystemD.

- (a) ¿Qué es systemd?

Systemd es un sistema de gestión de servicios y un proceso de inicialización (init system) que se utiliza en muchas distribuciones modernas de GNU/Linux. (los runlevels son reemplazados con targets)

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

Hace referencia a las unidades de trabajo de tipo service, socket, target o snapshot, y pueden estar en dos estados: active o inactive.

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

Es la herramienta principal en systemd para gestionar servicios, controlar el estado del sistema y administrar el proceso de inicialización. systemctl permite a los usuarios interactuar con los servicios y otras unidades de systemd.

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

Un target es una unidad de systemd que agrupa y coordina otras unidades del sistema para alcanzar un estado específico del sistema.

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

Muestra una representación visual de la jerarquía de procesos en forma de un árbol. Esto permite observar cómo los procesos están relacionados entre sí y cómo se derivan unos de otros, facilitando la comprensión de la estructura de procesos en el sistema.

4. Usuarios:

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

Se almacena en varios archivos clave ubicados en el directorio /etc.

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

El UID es un identificador numérico único asignado a cada usuario en el sistema.

El GID es un identificador numérico único asignado a cada grupo en el sistema.

Cada UID debe ser único para garantizar que cada usuario se identifique de manera única. Esto significa que no pueden coexistir UIDs iguales en el mismo sistema. La misma regla se aplica a los GIDs; cada GID también debe ser único para garantizar la correcta administración de grupos.

(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 usuario **root** es el superusuario o administrador del sistema. Tiene el nivel más alto de privilegios y control sobre el sistema operativo.

Aunque el usuario root es único en el sentido de que tiene el mismo nombre de usuario y UID, es posible que diferentes instancias de usuarios tengan privilegios de administrador o superusuario. Sin embargo, estos usuarios no se llaman root y, por lo general, se les otorgan privilegios de administrador a través de mecanismos como **sudo**.

El UID del usuario root es **0**. Este UID especial está reservado para el usuario root y se utiliza para identificarlo en el sistema. Debido a su UID único, el sistema operativo concede al usuario root el máximo nivel de permisos.

(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 cátedra (si no existe, deberá crearlo). Luego, sin iniciar sesión como este usuario cree un archivo en su home personal que le pertenezca. Luego de todo esto, borre el usuario y verifique que no queden registros de él en los archivos de información de los usuarios y grupos.

Hecho.

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

Hecho.

5. FileSystem:

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

Cada archivo o directorio tiene una combinación de permisos para tres categorías de usuarios:

1. **Propietario:** El usuario que creó o es dueño del archivo o directorio.
2. **Grupo:** Un grupo de usuarios que puede tener permisos compartidos sobre el archivo.
3. **Otros:** Cualquier usuario que no sea el propietario ni miembro del grupo.

Tipos de Permisos

Hay tres tipos de permisos en GNU/Linux:

1. **Lectura (r)**: Permite leer el contenido del archivo o ver el listado de archivos en un directorio.
2. **Escritura (w)**: Permite modificar o eliminar el archivo o su contenido. En el caso de los directorios, permite crear, renombrar o eliminar archivos dentro del directorio.
3. **Ejecución (x)**: Permite ejecutar el archivo como un programa o script. En un directorio, permite entrar en él y acceder a sus archivos.

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

Hecho en el doc de comandos de 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?

La notación octal se forma sumando los valores de los permisos que desees asignar. Para cada categoría (propietario, grupo, otros), sumas los valores de lectura, escritura y ejecución para obtener un número entre 0 y 7. La referencia de cada valor está en el doc de comandos de linux para el comando `chmod`.

- (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` o los usuarios con privilegios `sudo` pueden acceder a archivos sin tener directamente permisos sobre ellos.

- (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 completa desde la raíz del sistema de archivos (`/`) hasta el archivo o directorio deseado.
- relative path name es una ruta que se especifica en relación al directorio de trabajo actual del usuario (el directorio en el que el usuario se encuentra en ese momento). Esta ruta no comienza desde el directorio raíz (`/`), sino desde el directorio donde estás posicionado.

- (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.

- Para saber el directorio actual utilizar: `pwd`
- Para ir directamente al directorio personal del usuario actual usar: `cd ~`
- Si, se puede acceder a otros directorios a partir del parámetro `~`: `cd ~/Documentos` por ejemplo.

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

Hecho en el docs de comandos de linux.

6. Procesos:

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

Un proceso en un sistema operativo GNU/Linux es una instancia en ejecución de un programa.

PID (Process ID):

- Es el identificador único de un proceso dentro del sistema operativo.

- Cada proceso en GNU/Linux tiene un número único que lo identifica, asignado por el sistema en el momento en que se crea el proceso.

PPID (Parent Process ID):

- Es el identificador del proceso padre (Parent Process ID).
- Cuando un proceso se crea, otro proceso (normalmente el shell o el init/systemd) lo genera, y este proceso "padre" tiene un PPID que identifica qué proceso lo creó.
- De esta manera, se puede rastrear la jerarquía de procesos en el sistema.

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

ps (process status): muestra una instantánea de los procesos en ejecución en el momento en que se ejecuta.

top: muestra una vista dinámica y en tiempo real de los procesos que están utilizando el sistema.

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

Los procesos pueden ejecutarse en **Background** (segundo plano) o en **Foreground** (primer plano). Estas dos formas de ejecución definen cómo interactúa el proceso con el usuario y los recursos del sistema.

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

Para ejecutar un proceso en **Background** simplemente añadir un **ampersand (&)** al final del comando.

Cuando un proceso se ejecuta en segundo plano, la terminal te devolverá inmediatamente el control y te mostrará un número entre corchetes (el **Job ID**) seguido del **PID** del proceso: [1] 12345

Con **fg % (JID)** pasas un proceso background a foreground.

Para pasar un proceso de foreground a background primero hay que detenerlo con **ctrl + z** y ejecutando **bg % (JID)**.

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

Es un operador que permite **redireccionar la salida de un comando como entrada de otro**. Esto facilita encadenar varios comandos para que trabajen juntos.

Ejemplos:

- **ps aux | grep firefox** (busca en los procesos que se están ejecutando actualmente, cuales coinciden con la palabra "firefox")
- **ls | wc -l** (cuenta la cantidad de directorios que hay en ls)

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

Permite manipular las **entradas** y **salidas** de los comandos. Su finalidad es redirigir los flujos de datos entre archivos, comandos y dispositivos, y se puede aplicar a la **salida estándar** (stdout / > / >>), **entrada estándar** (stdin / <) y **salida de error** (stderr / 2>).

Ejemplos:

- **ls > archivos.txt**
- **ls >> archivos.txt**
- **sort < lista.txt**
- **ls /directorio_inexistente 2> errores.log**

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

Sirve para enviar diferentes tipos de señales a los procesos para realizar acciones como detener, reiniciar o continuar la ejecución de un proceso.

Ejemplo: Forzar la terminación de un proceso con PID = 1234.

> **kill -9 1234**

(Los parametros estan en el doc de comandos de linux)

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

Hecho en el doc de comandos de Linux.

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

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

Hace referencia al proceso de **agrupar varios archivos y directorios en un único archivo**. Sin embargo, el empaquetado en sí **no comprime** los archivos, sino que solo los agrupa.

(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: tar grep gzip zgrep wc

Hecho en el doc de comandos de linux.

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

A. ls -l > prueba

No se puede, ya que prueba es un directorio.

B. ps > PRUEBA

No se puede, ya que prueba es un directorio.

C. chmod 710 prueba

El 7 le da permisos de lectura, escritura y ejecución al propietario.

El 1 le da permisos de ejecución al grupo.

El 0 no le da permisos para nada a otros.

D. chown root:root PRUEBA

Le transfiere la propiedad del archivo prueba al usuario root. En este caso no puedo hacerlo, ya que es necesario ser un superusuario para poder cambiar la propiedad de un archivo.

E. chmod 777 PRUEBA

Le da permisos de lectura, escritura y ejecución al propietario, grupo y otros.

F. chmod 700 /etc/passwd

Intenta cambiar los permisos de un archivo de texto passwd al cual no tiene permisos.

G. passwd root

Intenta eliminar/cambiar la contraseña del usuario root, no es posible ya que no posee los permisos.

H. rm PRUEBA

Intenta eliminar el directorio PRUEBA, no es posible ya que es un directorio y para poder hacerlo se necesita el parámetro -r o -rf.

I. `man / e t c / shadow`

No funcionara ya que más se usa para ver un manual de algún comando, sin embargo shadow no es un comando, es un archivo.

J. `find / -name * . c o n f`

No funciona ya que no existe el directorio -name.

K. `usermod r o o t -d /home/ newroot -L`

Intenta añadir al usuario root a un grupo, sin embargo no tengo permisos.

L. `cd / r o o t`

No puedo acceder al home del usuario root.

M. `rm *`

Intenta borrar todo, no se puede ya que prueba es un directorio.

N. `cd / e t c`

Accede correctamente al directorio /etc.

O. `cp * /home -R`

Intenta copiar todo lo que contiene /etc en el home del usuario, sin embargo no posee permisos para esto.

P. `shutdown`

Requiere privilegios de superusuario.

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

(a) Terminar el proceso con PID 23.

`kill 3 23.`

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

Se pudo hacer el killall systemd, me cerró la sesión con el usuario que ejecuto el comando.

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

`find /home -type f -name "*.conf"`

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

`ps -aux > /home/diego/Procesos/procesos.txt` (crea el archivo procesos.txt en el directorio Procesos con los procesos actuales según el parámetro -aux)

(e) Cambiar los permisos del archivo /home/<su nombre de usuario>/xxxx a: Usuario: Lectura, escritura, ejecución Grupo: Lectura, ejecución Otros: ejecución

`chmod 751 xxxx`

(f) Cambiar los permisos del archivo /home/<su nombre de usuario>/yyyy a: Usuario: Lectura, escritura.

Grupo: Lectura, ejecución Otros: Ninguno

`chmod 650 yyyy`

(g) Borrar todos los archivos del directorio /tmp

`sudo rm -rf /tmp/*`

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

`sudo chown iso2010 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/diego/Donde/posicion.txt`

10. 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 primera letra de su nombre seguida de su apellido. Asígnele una contraseña de acceso.

`sudo useradd -m DPingo`

`sudo passwd DPingo`

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

- En mi caso al usar el parámetro -m, se le creó un directorio en /home a DPingo.
- En el directorio /etc/passwd se agregó la información básica del nuevo usuario. (DPingo:UID:1001 GID:1001: home:/home/DPingo shell:/bin/bash)
- En el directorio /etc/shadow se agrega la información encriptada de las contraseñas del usuario.
- En el directorio /etc/group normalmente se crea un grupo con el mismo nombre que el usuario
- En el directorio /etc/gshadow se agrega información encriptada sobre las contraseñas de los grupos del usuario.

(d) Crear un directorio en /tmp llamado cursada2017

`mkdir /tmp/cursada2017`

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

`sudo cp -r /var/log/* /tmp/cursada2017` (-r es necesario para que copie todos los subdirectorios dentro de /log de forma recursiva)

(f) Para el directorio antes creado (y los archivos y subdirectorios contenidos en él) cambiar el propietario y grupo al usuario creado y grupo users.

`sudo chown -R DPingo:DPingo cursada2017` (-R especifica que todos los archivos y directorios dentro de cursada2017 también cambien de dueño)

(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.

`sudo chmod -R u+rw cursada2017`

`sudo chmod -R g+w cursada2017`

`sudo chmod -R o+wx cursada2017`

Asegurarse de también haber sacado recursivamente los permisos que previamente pudieron haber tenido.

(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.

`/dev/pts/0`

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

ps -aux | wc -l

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

who | wc -l

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

write tty3

(m) Apague el sistema.

sudo shutdown now

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

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

mkdir /home/diego/23159

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

touch LEAME

vi LEAME

(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

sudo 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?.

ls /etc > /home/diego/LEAME.txt

Pude hacer esto porque en este caso hice un .txt, el LEAME que había antes era un 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.

Para encontrar un archivo usaría locate -i (archivo o cadena de caracteres)

Si tuviera que buscar varios archivos usaria: find /home -name "*.config"

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

sudo find / -name "*.so" > /home/diego/23159/ejercicio_f

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

A. mkdir iso

Crea el directorio iso

B. `cd ./iso; ps > f0`

Crea un archivo de texto llamado f0 dentro del directorio iso, el cual contiene un listado de los actuales procesos ejecutándose.

C. `ls > f1`

Crea un archivo de texto llamado f1 que contiene un listado de los directorios dentro de ./iso.

D. `cd /`

Accede al directorio /.

E. `echo $HOME`

Imprime un texto en la terminal con el directorio principal del usuario.

F. `ls -l $> $HOME/iso/l s`

El \$ genera un error al juntarlo con -l.

G. `cd $HOME; mkdir f2`

Accede al directorio privado del usuario y crea un directorio llamado f2.

H. `ls -ld f2`

Imprime los permisos del directorio f2.

I. `chmod 341 f2`

Le asigna al usuario escritura y ejecución, al grupo lectura y a otros ejecución en el directorio f2.

J. `touch dir`

Crea un archivo de texto llamado dir.

K. `cd f2`

Accede al directorio f2.

L. `cd ~/iso`

Accede al directorio ./iso.

M. `pwd > f3`

Crea un archivo de texto con la ruta actual dentro.

N. `ps | grep 'ps' | wc -l >> ../f2 /f3`

Busca en los actuales procesos ejecutándose la cadena "ps" y retorna la cantidad de ocurrencias que tiene en un número decimal y agregarlo al final del archivo de texto f3.

O. `chmod 700 ../f2; cd ..`

Le da todos los permisos al usuario dueño del archivo f2, y accede al directorio privado del usuario.

P. `find . -name etc/passwd`

La Shell advierte que no declaraste una cadena de caracteres, así que es lo mismo que hacer un ls de todo lo que hay dentro de ./etc/passwd.

Q. `find / -name etc/passwd`

Intenta hacer la búsqueda, sin embargo el usuario no posee los permisos para acceder a los archivos, además al no declarar una cadena, retorna todo.

R. mkdir ejercicio5

Crea un directorio llamado ejercicio5.

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

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

```
mkdir /tmp/logs
```

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

```
sudo cp -R /var/log/* /tmp/logs
```

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

```
sudo tar -cvf misLogs.tar logs
```

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

```
sudo tar -czvf misLogs.tar.gz logs
```

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

```
sudo cp -r logs misLogs.tar.gz /home/diego
```

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

```
sudo rm -rf /tmp/logs
```

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

```
sudo tar -xzf misLogs.tar.gz -C /home/diego/misLogs
```