Introducción a los Sistemas Operativos / Conceptos de sistemas operativos

Práctica 2

Objetivo

El objetivo de esta práctica es que el alumno comprenda los aspectos principales acerca de la estructura del sistema Operativo GNU/Linux en lo que respecta a procesos, usuarios, filesystems, permisos, etc.

1. Editor de textos:

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

- VIM
- VI
- 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.

- ***Comandos***
  - **[CAT]:** Muestra el contenido de todo el archivo a la vez, y también puede conectar varios archivos para mostrar. A menudo se usa junto

con el símbolo de redirección y es adecuado para situaciones en las que el contenido del archivo es pequeño;

- **[MORE]** y **[LESS]** se utilizan generalmente para mostrar el contenido de un archivo con más de una pantalla y proporcionar la función de pasar páginas. **MORE** es más potente que **CAT** y proporciona funciones de visualización de paginación, y **LESS** es más potente que **MORE** , proporcionando comandos como cambio de página, salto y búsqueda.

- **Modos de operación que posee el editor de textos VI:**
  - **Modo de comandos**: En el modo de comandos, podemos desplazarnos dentro de un archivo y efectuar operaciones de edición como buscar texto, eliminar texto, modificar texto, etc. VI suele iniciarse en modo de comandos.
  - **Modo insertar**: En el modo insertar, podemos tipear texto nuevo en el punto de inserción de un archivo. Para volver al modo de comandos, presione la tecla Esc (Escape).
  - Estos dos modos determinan la forma de comportamiento del editor. Todo lo que se tipee en el modo insertar se considerará como texto a insertar en el archivo. Si intenta tipear un comando y no sucede nada o el carácter aparece debajo del cursor, seguramente olvidó presionar Esc para salir del modo insertar.

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

//COMANDOS VI

- **[i]**: Ingrese al modo *insertar* antes del carácter en la posición actual. Tipee el texto deseado y presione Esc para volver al modo de comandos. Use “i” para insertar texto al comienzo de la línea actual.
- **[a]**: Ingrese al modo insertar antes del carácter en la posición actual. Tipee el texto deseado y presione “Esc” para volver al modo de comandos. Use “***A”*** para insertar texto al final de la línea actual.
- **[c]**: Usw “C” para modificar el carácter actual e ingresar al modo insertar para tipear caracteres de reemplazo.
- **[o]**: Abra una línea nueva para insertar texto debajo de la línea actual. Use “O” para abrir una línea arriba de la línea actual.
- **[CW]**: Elimine el resto de la palabra actual e ingrese al modo insertar para reemplazarla. Use un conteo de repetición para reemplazar varias palabras. Use c$ para reemplazar hasta el final de la línea.
- **[DW]**: Igual a “cw” y “c$”, con la excepción de que no se ingresa al modo insertar.
- **[DD]**: Elimine la línea actual. Use un conteo de repetición para eliminar varias líneas.
- **[x]**: Elimine el carácter en la posición del cursor. Use un conteo de repetición para eliminar varios caracteres.
- **[P]**: Coloque el último texto eliminado después del carácter actual. Use P para colocarlo antes del carácter actual.
- **[XP]**: Esta combinación de x y p es una expresión muy útil que intercambia lugares entre el carácter en la posición del cursor y el que tiene a la derecha.

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 ejecuta el código del BIOS
2. El BIOS ejecuta 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 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 el 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

(***BIOS>POST>MBR>MBC>KERNEL load>INITRD>INIT>INITTAB>RUNLEVEL1>)***

(b) Proceso INIT. ¿Quién lo ejecuta?

- El ***Kernel*** ejecuta el proceso ***init***.
- En los sistemas tipo Unix, ***init*** (abreviatura de *initialization*) es el primer proceso en ejecución tras la carga del kernel y el que a su vez genera todos los demás procesos. Se ejecuta como demonio y por lo general tiene PID 1.

¿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) antes, ahora → systemd
5. Es el encargado de montar los filesystems y de hacer disponible los demás dispositivos

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

- PSTREE: Muestra información de los procesos en ejecución en forma de árbol. (Systemd /...)

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

- Los *runlevels* son el modo en el que arranca Linux .
- Cada *runlevel* es responsable de levantar (iniciar) o bajar (parar) una serie de servicios.

//VER PERMISOS ***UGO*** -> “ls-l” User Group O??

>> Permiso de lectura (read)

Si tienes permiso de lectura de un archivo, puedes ver su contenido.

>> Permiso de escritura (write)

Si tienes permiso de escritura de un archivo, puedes modificar el archivo. Puedes agregar, sobrescribir o borrar su contenido.

>> Permiso de ejecución (execute)

Si el archivo tiene permiso de ejecución, entonces puedes decirle al sistema operativo que lo ejecute como si fuera un programa. Si es un programa llamado «foo» lo podremos ejecutar como cualquier comando.
O un script (intérprete) que necesita permiso de lectura y ejecución, un programa compilado solo necesita ser lectura.

(e) ¿A qué hace referencia cada nivel de ejecución según el estándar?

- Existen 7, y permiten iniciar un conjunto de procesos al arranque o apagado del sistema.
- _0 - Halt (Parada)
- _1 - Single user mode (Modo monousuario)
- _2 - Multiuser, without NFS (modo multiusuario sin soporte de red)
- _3 - Full multiuser mode (modo multiusuario completo, console)
- _4 – No se utiliza
- _5 - X11 (modo multiusuario completo, con login gráfico basado en X)
- _6 - Reboot (Reiniciar)

¿Dónde se define qué Runlevel se ejecuta al iniciar el sistema operativo?

- En ***inittab*** se define que runlevel ejecutar.

¿Todas las distribuciones respetan estos estándares?

- *No todas las distribuciones respetan este estándar.*

(f) Archivo /etc/inittab. ¿Cuál es su finalidad?

- El archivo ***inittab*** se encarga de gestionar el inicio del sistema.

¿Qué tipo de información se almacena en el?

- En el archivo ***inittab*** se guarda la configuración del proceso init con respecto a que runlevels ejecutar, y en qué acciones consisten cada uno de ellos (donde buscar los procesos referentes a cada runlevel).

¿Cuál es la estructura de la información que en él se almacena?

- Cada línea en el archivo "***inittab***" consiste de cuatro campos delimitados por **":"** :

id:runlevels:action:process

Descripción de los campos:

- ***id*** (identification code) – Consiste en una secuencia de 1 a 4 caracteres que identifican su función.
- ***runlevels*** – Lista el "runlevel" que aplica a está entrada.
- ***action*** – El código en éste campo le dice a "init" cómo tratar el proceso:

  Valores posibles que incluye: initdefault, sysinit, boot, bootwait, wait, and respawn.
- ***process*** – Define el comando o secuencia de comandos que se ejecutará.

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

- Para cambiar de nivel de ejecución sólo hay que ejecutar ***init*** seguido del número del runlevel. Por ejemplo:

  init 0: Cambia al runlevel 0 (se apaga el sistema, equivalente al comando *halt*).

(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.

- Los **scripts RC** se guardan en los directorios *etc/rc**X**.d/* (donde **X** es un runlevel), *son enlaces a las funciones que crean los servicios que están en etc/init.d/*
- Los nombres en estos directorios empiezan por una letra ***(S o K***) seguidos de un número y el nombre del servicio. La letra **S** significa iniciar (S de start). La letra **K** significa acabar (K de kill). El número es de dos dígitos, de 00 a 99 e indica el orden en el que se arrancará el servicio. Por ej: S20lpd.

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

- ***Inserv*** se utiliza para manejar y actualizar el orden de los enlaces simbólicos de los RC de una forma más sencilla, en vez de tener que editar muchos archivos en los distintos runlevel para que sean enlaces simbólicos a mi servicio, solo tengo que modificar la cabecera de mi archivo, explicando en que runlevels quiero que se ejecute, y otras relaciones con los scripts rc.

Puede incluir:

- Default-Start / Default-Stop : Indica los runlevels en los que se debe iniciar/frenar el script
- Required-Start / Required-Stop : indica los servicios que son necesarios tener iniciados previo a ejecutar el script.
- *Es más rápido que un arranque tradicional*. La efectividad de esta técnica depende del número de servicios a iniciar, así como de la red de dependencia que haya entre ellos. Cuantos más servicios independientes haya, más se acelerará el arranque (más servicios podrán ser arrancados en paralelo).

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

- ***Upstart*** es un sistema de arranque compatible System five (System V) pero que ejecuta las tareas de forma asincrónica permitiendo así una mejora en las prestaciones, las tareas y servicios que son ejecutados ante eventos (arranque del equipo o inserción de un dispositivo USB), definidos como tareas o Jobs.
  En distribuciones como Fedora o Ubuntu se usa Upstart en vez de System V.

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

Linux adoptó el esquema ***System V***, Ubuntu y algunas otras distribuciones de Linux utilizan ahora ***upstart*** como reemplazo para el proceso de inicialización tradicionales y el motivo son las desventajas que posee Init en comparación con Upstart

- ***Upstart***
  - Trabaja de forma asíncrona supervisando las tareas mientras el sistema está arrancado. También gestiona las tareas y servicios de inicio cuando el sistema arranca y los detiene cuando el sistema se apaga.
  - Upstart es capaz de ejecutar scripts de sysvinit sin modificaciones. De esta manera se diferencia de la mayoría de reemplazos de init, que normalmente requieren una transición completa para funcionar correctamente y no son compatibles con un entorno mixto formado por métodos de arranque tradicionales y nuevos.
  - Como principal diferencia con sysVinit que es estrictamente sincrónico (dependency-based).

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

- Estos trabajos se denominan ***jobs***
- El principal objetivo de un **job** es definir servicios o tareas a ser ejecutadas por *init*
  - Son scripts de texto plano que definene las acciones/tareas (unidad de trabajo) a ejecutar ante determinados eventos

- Cada job es definido en el ***/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:

//JOBS https://geekland.eu/gestionar-trabajos-en-linux-terminal/

*# MySQL Service // **Comments begin with a '#' and continue until the end of the line.***
d e s c r i p t i o n "MySQL Server "
author " info autor "
s t a r t on (n e t −d e v i c e −up
and l o c a l − f i l e s y s t e m s
and r u n l e v e l [2 3 4 5])
s t o p on r u n l e v e l [0 1 6]
[...]
*exec / usr / sbin /mysqld //All job files must have either an **exec** or **script** stanza. This specifies what will be run for the job.*
[...]

(n) ¿Qué es systemd ?

- *Systemd es un sustituto para el Init de Linux*. Está hecho para proveer un mejor framework para expresar las dependencias del servicio, permite hacer más trabajo paralelamente al inicio del sistema y reducir la sobrecarga del shell.

- El nombre viene del sufijo **system daemon** (procesos en segundo plano) con la letra "**d**".
- Comparado con ***SystemV,*** init, que es utilizado por la mayoría de las distribuciones anteriores, ***systemd*** puede tomar ventaja de nuevas técnicas:
    - Los servicios de activación de sockets y la activación de buses, que conduce a una mejor paralelización de servicios independientes.
    - cgroups se utilizan para realizar un seguimiento de los procesos de servicio, en lugar de PIDs. Esto significa que los demonios no pueden "escapar" de systemd aunque estén doblemente-bifurcados.

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

- Systemd utiliza la activación de socket y D-Bus para iniciar los servicios y permite el inicio de los demonios bajo demanda, realiza un seguimiento de los procesos con el uso de los grupos de control de Linux.

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

- Los grupos de control (o ***cgroups***, como se les conoce comúnmente) son una característica proporcionada por el kernel de Linux para administrar, restringir y auditar grupos de procesos. En comparación con otros enfoques como el comando nice(1) o /etc/security/limits.conf, los cgroups son más flexibles ya que pueden operar en (sub) conjuntos de procesos (posiblemente con diferentes usuarios del sistema).

3. Usuarios:

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

- Los usuarios están almacenados en un archivo de texto en el sistema llamado ***passwd*** file. Este archivo está localizado 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 el identificador único de usuario, el **GID** es el identificador único de grupo, en versiones anteriores podía haber varios usuarios con el mismo UID, luego esto fue corregido, ya que de no hacerlo se podía modificar el UID de un usuario a 0 y así este tendría los mismos permisos que el root.(ya que este último tiene UID=0)

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

- El usuario root es el usuario que tiene permiso para manejar y modificar todas las cosas importantes, puede crear usuarios modificarlos, modificar los permisos de estos, modificar cualquier archivo ,etc, **su uid es 0**.No puede existir más de un root.

(d) Agregue un nuevo usuario llamado iso2021 a su instalación de GNU/Linux, especifique que su home sea creada en /home/iso_2021, 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.

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

- **useradd o adduser:** Agrega el usuario, modifica el archivo etc/passwd. (va a pedir todos los datos)
- o TMB “`su - nombreusuario`”

- **usermod:** <nombre usuario>
    - g modifica el grupo inicial modifica etc/passwd
    - G modifica grupos adicionales etc/group
    - d modifica el directorio home del usuario, modif etc/password

- **userdel** <nombre usuario> elimina el usuario
- **su** <nombre de usuario> Cambia al usuario nombre de usuario antes pidiendo su contraseña, si no se pone nombre de usuario se establece que se quiere cambiar a ser super usuario

- **groupadd** <nombre del grupo>crea un nuevo grupo
- //su -l
- //usermod -a -G catedra iso_2021

- **who :** muestra quien está logueado.

- **groupdel** <nombre del grupo>elimina el grupo

- **passwd** <nombre_usuario>asigna o cambia la contraseña, modifica los archivos etc/shadow

PARA VER INFO DEL GRUPO

cat /etc/group o more /etc/group

4. FileSystem:

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

- Se usa la notación octal:
    - **X = ejecución : 1**
    - **W = escritura : 2**
    - **R = lectura : 4.**

  Por ejemplo:
    - si hay 7 es que hay permiso para **lectura, escritura y ejecución**,
    - 5 que solo hay permiso de ejecución y lectura.
    - Ahora bien,como se le da permisos para el usuario(dueño),grupo al que pertenece, y otros?, se usa ***UGO*** por ejemplo UGO=755 significa que el dueño tiene todos los permisos y que tanto el grupo del dueño como otros usuarios solo pueden leer y ejecutar.

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

- **chmod** <permisos><objeto> : Modifica los permisos de acceso de ficheros.si se usa con **–R** cambia los permisos de los contenidos del directorio seleccionado.

- **chown:** Cambia el usuario y grupo propietarios de los ficheros.La extensión * cambia a todos los archivos y directorios del directorio actual.La extensión –R hace que también se aplique a archivos y subcarpetas.
- **chgrp:** Cambia el grupo al que pertenecen los ficheros.

(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?

- Esto significa que se representa mediante la suma de estos números los permisos que tiene el usuario o grupo, o otros según la hilera del ***UGO***,
  1=ejecución
  2=escritura
  3=ejecución y escritura
  4=lectura
  5=lectura y ejecución
  6=lectura y escritura
  7=todos.

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

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

- Una dirección relativa es, por ejemplo: estoy en ***/home/miUsuario*** y accedo a /**desktop**, ahí me moví sin decir la dirección entera, sino solo como me muevo a partir de donde estoy

- En cambio, una dirección completa sería **/home/miUsuario/desktop/**

(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.

- Con ***pwd*** sé en qué directorio estoy. Si estoy logueado como el usuario que soy simplemente con ***cd*** voy a mi directorio base ***/home/miUsuario***.si soy root me lleva a un directorio vacío.

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

- **cd <directorio> :**Cambia el directorio, cd solo lleva al directorio base del usuario.
- **umount :** Desmonta un filesystem.
- **mkdir:** Crea un directorio con ***–mkdir***, da la opción de crear los directorios padre.
- **du :** Estima el uso de espacio de ficheros(del fichero seleccionado).
- **rmdir :** Borra directorios vacíos con la opción ***–p*** borra primero los directorios vacíos internos hasta llegar al seleccionado ejemplo rmdir –p a borra a/b/c e orden inverso si son todos vacíos. Existe una opción que hace que borre directorios aunque no estén vacíos.
- **df :** informa de la utilización del espacio de disco en sistemas de ficheros.
- **mount:** monta un filesystem.
- **ln:** crea enlaces entre ficheros, por defecto enlaces duros, con la opción ***–s*** crea enlaces blandos.

- **ls:** lista los contenidos de los directorios, con –l lista los permisos y otros datos de los ficheros.
- **pwd :** dice cual es el directorio actual.
- **cp** <option>SOURCE DEST copia source a dest o muchas source(s) a el directorio dest.
- **mv** mueve o renombra ficheros o directorios desde un origen a un destino.

5. Procesos:

(a) ¿Qué es un proceso?

- Un proceso es un programa en ejecución.

¿A que hacen referencia las siglas PID y PPID?

- ***PID*** es el identificador único del proceso, ***PPID*** es el identificador único del proceso padre.

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

- Todos los procesos en gnu Linux tienen estos atributos:
  - p o PID: Process ID, número único o de identificación del proceso.
  - P o PPID: Parent Process ID, padre del proceso
  - U o UID: User ID, usuario propietario del proceso
  - t o TT o TTY: Terminal asociada al proceso, si no hay terminal aparece entonces un '?'
  - T o TIME: Tiempo de uso de cpu acumulado por el proceso
  - c o CMD: El nombre del programa o comando que inició el proceso
  - RSS: Resident Size, tamaño de la parte residente en memoria en kilobytes
  - SZ o SIZE: Tamaño virtual de la imagen del proceso
  - NI: Nice, valor nice (prioridad) del proceso, un número positivo significa menos tiempo de procesador y negativo más tiempo (-19 a 19).
  - C o PCPU: Porcentaje de cpu utilizado por el proceso
  - STIME: Starting Time, hora de inicio del proceso

- S o STAT: Status del proceso, estos pueden ser los siguientes :
  - R runnable, en ejecución, corriendo o ejecutándose.
  - S sleeping, proceso en ejecución pero sin actividad por el momento, o esperando por algún evento para continuar
  - T sTopped, proceso detenido totalmente, pero puede ser reiniciado
  - Z zombie, difunto, proceso que por alguna razón no terminó de manera correcta, no debe haber procesos zombies
  - D uninterruptible sleep, son procesos generalmente asociados a acciones de IO del sistema
  - X dead, muerto, proceso terminado pero que sigue apareciendo, igual que los Z no deberían verse nunca

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

- Se puede usar ***ps*** que lista los procesos que se están ejecutando.

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

- **Background**: Linux permite iniciar una tarea en segundo plano y seguir haciendo otras cosas en la línea de comandos. Con un (&) al final de un comando, se puede iniciar en segundo plano y obtener la línea de comandos de vuelta de inmediato.

- **Foreground**: Un proceso en primer plano es diferente de un proceso de fondo de dos maneras:
  - 1. Algunos procesos en primer plano muestran al usuario una interfaz, a través de la cual el usuario puede interactuar con el programa.

    - 2. El usuario debe esperar a que un proceso en primer plano termine antes de ejecutar otro.

(d) ¿Cómo puedo hacer para ejecutar un proceso en Background?

- Para llevar un proceso de foreground a background se utiliza el comando bg , se utiliza tipeando bg %326 (siendo 326 el PID), o usamos ctrl +z.

¿Como puedo hacer para pasar un proceso de background a foreground y viceversa?

- Para llevar un proceso de background a foreground se utiliza el comando fg, de la misma manera que el bg.

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

- El pipe “|” de Linux permite tomar la salida de un proceso y mandarla como entrada a otro proceso.
    - Ejemplo: head -5 archivo | tail -1 permite quedarse con la 4 línea de archivo.

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

- La redirección tiene como objetivo poder concatenar las funciones de varios procesos consiguiendo un resultado en conjunto.
- > Envía la salida a un archivo
- >> Anexa la salida a un archivo
- cmd1 | cmd2 La salida de cmd1 se usa como argumento de cmd2
- < archivo Toma de "archivo" los argumentos
- Ejemplos:
- $ ls > ejercicio.txt

- $ cat /etc/fstab >> ejercicio.txt
- $ ordena < numAleatorios.

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

- El comando kill (un Proceso) envía una señal a un proceso que por lo general implica que este pare, o se termine de ejecutar, según el contexto. Es de la forma kill –opción –pid. Con la opción -9 se fuerza la terminación. Con -19 o -20 se detiene momentáneamente, con -18 se continúa la ejecución, con -1 se obliga a releer los archivos de configuración.

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

- **ps :** Despliega información de los procesos en ejecución.
- **kill:** envía una señal a un proceso que por lo general implica que este pare, o se termine de ejecutar.
- **pstree :** Despliega información de los procesos en ejecución en forma de árbol.
- **killall:** funciona como el kill pero en vez de pasarle el *PID* hay que poner el nombre.
- **top :** Da información en tiempo real de las tareas que se están ejecutando, especificando además el % de Cpu y Memoria que están utilizando, sus IDs, usuarios que lo están ejecutando, etc.
- **nice:** Setea la prioridad del proceso a ejecutar, mientras menor el número mayor la prioridad, o devuelve la prioridad del proceso.

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

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

- ***TAR***: Su función es la de incluir los archivos seleccionados en el mismo archivo, conservando los permisos y las estructuras de directorios de los mismos.

(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?

- El tamaño del nuevo archivo es menor, más o menos por la mitad.

(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:

// PÁGINA CON TAR INFO y uso

https://www.hostinger.com.ar/tutoriales/como-usar-comando-tar-linux#Como-verificar-un-archivo-tar-en-Linux

- **tar:** Crea y modifica archivos empaquetados, sus opciones básicas son:
  - cf: crea un nuevo archivo.tar a partir de un directorio
  - A: concatena archivos tar
  - c: crea un nuevo archivo.
  - delete: borra archivos
  - r: concatena files normales
  - t: lista los contenidos del archivo
  - x: extrae los archivos

- **grep :** Busca las líneas que contengan un determinado patrón que se pasa como parámetro,por defecto imprime estas líneas.

  opciones disponibles:

  - −**i**: no diferenciará entre mayúsculas y minúsculas.
  - −**w**: fuerza que sólo encuentre palabras concretas.
  - −**v**: selecciona las líneas que no coinciden.
  - −**n**: muestra el número de la línea con las palabras solicitadas.
  - −**h**: elimina el prefijo del nombre del archivo Unix en la salida.
  - −**r**: busca directorios recursivamente.
  - −**R**: como -r pero sigue todos los enlaces simbólicos.
  - −**l**: muestra sólo nombres de archivos con las líneas seleccionadas.
  - −**c**: muestra sólo una cuenta por archivo de las líneas seleccionadas.
  - **–color**: muestra los patrones coincidentes en colores.

- **gzip :** Es de la forma gzip [option][file] y comprime descomprime archivos,con gunzip se descomprime.
- **zgrep :** Invoca a grep para archivos comprimidos.
- **wc:** Imprime datos de los archivos como las líneas las palabras y el conteo de bytes para cada archivo, además de un total si especifica más de uno.

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

- **ls −l > prueba**
- **ps > PRUEBA**
- **chmod 7 1 0 p r u e b a**
- **chown r o o t : r o o t PRUEBA**
- **chmod 7 7 7 PRUEBA**
- **chmod 7 0 0 / e t c / p a s s w d**

- **passwd r o o t**
- **rm PRUEBA**
- **man / e t c / shadow**
- **f i n d / −name ∗ . c o n f**
- **u s e r m o d r o o t −d / home / n e w r o ocp t −L**
- **cd / root**
- **rm ∗**
- **cd / etc**
- **c p ∗ / home −R**
- **shutdown**

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

(a) Terminar el proceso con PID 23.

- > kill 23

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

- >killall init >>no se permite la operación

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

- >find /home .conf , no existe el fichero.

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

- **>**usaría ps> /home/miusuario/procesos, crea el archivo con los procesos que se están ejecutando.

(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

- rm –R /tmp

(h) Cambiar el propietario del archivo /opt/isodata al usuario iso2021.

- 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.

- pwd> /home/<nombreUsuario>/donde

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

// log en otra term ctrl alt f4

// commandos msg y < habilitas los mensajes entre usuarios.

>>write unUsuario

>> elMensaje

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

- a)su, pide la contraseña root y se la ingresa.
-

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

- b)lo llamé LM

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

- Se creó el directorio **/home/LM**, se actualizaron el **/etc/shadow** y el **/etc/passwd**, además se creó el nuevo grupo que lo contiene.

(d) Crear un directorio en /tmp llamado cursada2021

- **d)**cd /tmp mkdir cursada2021

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

- e) cp /var/log/* /tmp/cursada2021

(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 LM /tmp/cursada2010 chgrp users /tmp/cursada2010

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

- **g)** chmod –R 723 /home/LM

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

- **HECHO**

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

- VER

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

- ps -l

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

- who –q

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

- Utilizaremos “mensaje” > destinatario. Aunque con otros comandos como wall y write también podrían enviarse mensajes, mediante ficheros.

(m) Apague el sistema.

- shutdown [option] tiempo [mensaje]

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 "LÉAME".

(c) Cambie los permisos del archivo LÉAME, de manera que se puedan ver reflejados los siguientes permisos:
Dueño: ningún permiso Grupo: permiso de ejecución Otros: todos los permisos.

- Lo llame de dd hice chmod 017

(d) Vaya al directorio /etc y verifique su contenido. Cree un archivo dentro de su directorio personal cuyo nombre sea léame 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 "LÉAME.en este directorio?.

- Puedo crear el archivo léame ya que LÉAME es distinto de léame ya que Linux es case sensitive.

(e) ¿Qué comando utilizaría y de qué manera si tuviera que localizar un archivo dentro del filesystem?

- Usaría el comando find

¿Y si tuviera que localizar varios archivos con características similares? Explique el concepto teórico y ejemplifique.

- Si quiero encontrar varios archivos que cumplen con un patrón usaría grep, podría combinarlos con | para conseguir el resultado buscado.

(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".

- seria find / -name "*so" --ejercicio_f

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

- **mkdir i s o**
- **cd ./ 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.

(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 qué 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 los 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 del 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" "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**

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.

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

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

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

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

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

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