

## Practica 2 ISO:

1.- shift + q me lleva al modo comandos. q! salir del editor

a) existen vi, emacs, ed.

b) se diferencian en que con los comandos solo puedo visualizar el texto o concatenarlo, en cambio con los editores puedo hacer muchas mas modificaciones.

c) El editor de texto vi tiene tres modos de operación:

### 1 Modo comandos

Vim empieza en modo comando, también conocido como modo normal. En este modo se pueden emplear combinaciones de teclas para, por ejemplo, copiar líneas y trabajar en el formato del texto. Éste es el *modo central*, desde el que se cambia a los otros modos. Si no se sabe qué se está haciendo, pulsando dos veces la tecla Escape siempre se puede volver al modo normal.

### 2 Modo inserción

En modo inserción cuando se pulsan las teclas se edita el texto como en otros editores. Se puede cambiar del modo comandos al modo inserción pulsando la tecla i. Hay un gran abanico de comandos para pasar al modo inserción, que difieren sustancialmente, pues permiten por ejemplo editar al final de la línea, en un punto concreto del texto, editar borrando una palabra, entre muchas otras. Un usuario experto puede sacar un gran provecho de la existencia de esta variedad de órdenes.

Cambiando al modo comandos para realizar ciertas tareas se incrementa en gran medida la eficiencia en la edición, y se puede aprovechar la potencia completa de Vim.

### 3 Modo línea de órdenes

A este modo se accede pulsando la tecla dos puntos :. Tras los dos puntos se pueden introducir órdenes complejas, como por ejemplo buscar y reemplazar con expresiones regulares. Pulsando la tecla Esc se puede volver al modo órdenes. Las búsquedas se pueden realizar con la orden / (hacia adelante) y ? (hacia atrás). También se pueden filtrar líneas mediante !.

1 d)

**dd:** Borra una línea entera.

Puede insertar texto (pasar a modo edición) con varias teclas:

**i:** Inserta texto antes del carácter sobre el que está el cursor.

**a:** Inserta texto después del carácter sobre el que está el cursor.

**I:** Inserta texto al comienzo de la línea en la que está el cursor.

**A:** Inserta texto al final de la línea en la que está el cursor.

**o:** Abre espacio para una nueva línea después de la línea en la que está el cursor y permite insertar texto en la nueva línea.

**O:** Análogo al anterior, pero abre espacio en la línea anterior.

**ESC:** Abandonar el modo de inclusión de texto para volver al modo de comandos; también se usa para cancelar comandos. (Usarlo en caso de duda)

**CntI-F:** Avanzar una página hacia adelante.

**Cntl-B:** Avanzar una página hacia atrás.

**Cntl-L:** Refrescar la pantalla.

**G:** Cursor al final del fichero.

**1G:** Cursor al principio del fichero.

**\$:** Cursor al final de la línea.

**0 (cero):** Cursor al principio de la línea.

Para pasar de modo edición a modo de comandos se emplea la tecla ESC, para desplazarse sobre el archivo se emplean las teclas **j**(abajo), **k** (arriba), **h** (izquierda) y **l** (derecha).

2.-

a) se ejecuta el código de la bios que realiza un chequeo general, el hardware lee el mbr, se carga el mbr (gestor de arranque), se carga el kernel, se monta el sistema de archivos raíz, se ejecuta el proceso init, lee el /etc/inittab, se ejecutan los scripts del runlevel 1 y le indica que vaya al runlevel por defecto, se ejecutan los scripts del runlevel por defecto.

b) Al proceso init lo ejecuta el kernel, su función es cargar todos los procesos para el correcto funcionamiento del sistema operativo. Como por ejemplo montar los filesystem, y hacer disponibles los demás dispositivos.

c) Los runlevels son el modo en el que arranca Linux siempre que sea system 5, cada runlevel es responsable de levantar o bajar una serie de servicios.

d) **\_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)

no todas las distribuciones respetan este estándar.

en inittab se define que runlevel ejecutar.

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

### **Id:runlevels:action:process**

Niveles\_ejecucion: el/los nivel de ejecución en los que se realiza la acción.

Acción: describe la acción a realizar:

Wait: Se inicia cuando se entra al runlevel e init espera a que termine initdefault

Ctrlaltdel: se ejecutará cuando init reciba la señal SIGINT  
off, repawn, once, boot, bootwait, powerwait, otras...

Proceso: el proceso exacto que será ejecutado

Niveles\_ejecucion: el/los nivel de ejecución en los que se realiza la acción.

Acción: describe la acción a realizar:

Wait: Se inicia cuando se entra al runlevel e init espera a que termine

initdefault

Ctrlaltdel: se ejecutará cuando init reciba la señal SIGINT

off, repawn, once, boot, bootwait, powerwait, otras...

Proceso: el proceso exacto que será ejecutado

f) En la línea de comandos se permite cambiar de un nivel de ejecución a otro con la herramienta *init*. 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*).

g) los scripts RC se guardan en los directorios *etc/rcX.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 un 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.

j) *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 run-levels 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).

k) *Upstart* es un sistema de arranque compatible System five pero que ejecuta las tareas de forma asincrónica permitiendo así una mejora en las prestaciones, las tareas y servicios 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.

### 3.-

A) Se usan:

*-etc/passwd*, para guardar el UID, el GID su directorio de usuario su nombre, etc. Tiene el siguiente formato: **nombre\_de\_cuenta :**

**contraseña : numero\_de\_usuario : numero\_de\_grupo : comentario : directorio :**

**programa\_de\_inicio**

*-etc/group* para guardar los datos del grupo como su GID sus miembros. Su formato es **nombre\_de\_grupo : campo\_especial : numero\_de\_grupo: miembro1, miembro2**

Con frecuencia, el campo especial está vacío. El número de grupo corresponde al número del vínculo entre los archivos `/etc/group` y los archivos [`/etc/passwd`](#)

`-etc/shadow` para guardar la contraseña de forma encriptada, junto con la fecha de expiración.

b) 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) 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) `adduser` : agrega el usuario, modifica el archivo `etc/passwd`. (va a pedir todos los datos)  
`passwd <nombre usuario>` asigna o cambia la contraseña, modifica los archivos `etc/shadow`  
`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  
`groupdel<nombre del grupo>` elimina el grupo  
`groupadd<nombre del grupo>` crea un nuevo grupo  
`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.

`who` muestra quien está logueado.

4)

a) se usa la notación octal: `-ejecucion:1 -escritura:2 -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) `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) 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=lectura y escritura, 4=lectura, 5=lectura y ejecución, 6=lectura y escritura, 7=todos.

d) el root puede acceder a todo aunque no le hayan dado permiso. ¿??

e) una dirección relativa es por ejemplo estoy en /home/miusuario y accedo a desktop, ahí me movi sin decir la dirección entera, sino solo como me muevo a partir de donde estoy, en cambio una dirección completa sería /home/miusuario/desktop/prueba

f) con pwd se ve en que 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) cd directorio: cambia el directorio, cd solo lleva al directorio base del usuario.

-mkdir crea un directorio con -mkdir da la opción de crear los directorios padre.

-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.

-mount monta un filesystem.

-umount: desmonta un filesystem.

-du: estima el uso de espacio de ficheros (del fichero seleccionado)

-df: informa de la utilización del espacio de disco en sistemas de ficheros.

-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 cuál 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) Un proceso es un programa en ejecución. PID es el identificador único del proceso, PPID es el identificador único del proceso padre. 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 <ul style="list-style-type: none"> <li>• R runnable, en ejecución, corriendo o ejecutándose</li> <li>• S sleeping, proceso en ejecución pero sin actividad por el momento, o esperando por algún evento para continuar</li> <li>• T sTopped, proceso detenido totalmente, pero puede ser reiniciado</li> <li>• Z zombie, difunto, proceso que por alguna razón no terminó de manera correcta, no debe haber procesos zombies</li> <li>• D uninterruptible sleep, son procesos generalmente asociados a acciones de IO del sistema</li> <li>• X dead, muerto, proceso terminado pero que sigue apareciendo, igual que los Z no deberían verse nunca</li> </ul>

b) se puede usar ps que lista los procesos que se están ejecutando.

C) Proceso en 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.

Proceso en Foreground: Un proceso en primer plano es diferente de un proceso de fondo de dos maneras:

1. Algunos procesos en primer plano muestra al usuario una interfaz, a través del 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)

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.

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

e) 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) 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) 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 –opcion –pid. Con la opción -9 se fuerza la terminación. Con -19 o -20 se detiene momentaneamente, con -18 se continua la ejecución, con -1 se obliga a releer los archivos de configuración.

h)

- ps : despliega información de los procesos en ejecución.
- pstree: despliega información de los procesos en ejecución en forma de árbol.
- 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.
- killall: funciona como el kill pero en vez de pasarle el pid hay que poner el nombre.
- nice: setea la prioridad del proceso a ejecutar, mientras menor el numero mayor la prioridad, o devuelve la prioridad del proceso.

6.

a) Los archivos TAR no son archivos comprimidos sino empaquetados. Tar es un empaquetador, algo mas parecido a un compresor, pero sin la compresión. 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) El tamaño del nuevo archivo es menor, mas o menos por la mitad

c) primero empaqueto los archivos con tar, y ahora \*\*\*\*\*ver como zipearlos

d) ???

e) -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

-gzip: es de la forma gzip [opcion][file] y comprime descomprime archivos, con gunzip se descomprime.

-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 mas de uno.

-grep: busca las líneas que contengan un determinado patrón que se pasa como parametro, por defecto imprime estas líneas.

Es de la forma grep [opcion]patrón [fichero]

Tiene varias opciones:

grep -E interpreta el patrón como una expresión regular extendida.

grep -F interpreta el patrón como una lista de strings, separadas por nuevas líneas, una

coincidencia con cualquier línea, da como resultado un acierto.

-zgrep invoca a grep para archivos comprimidos.

7. a) ls-l> prueba: lista los permisos del archivo y otros datos en el archivo prueba

b)ps> prueba : crea un archivo llamado PRUEBA y guarda ahí los datos de los procesos en ejecución.

c)chmod 710 prueba: cambia los permisos de prueba, quedando todo para el propietario, ejecución para el grupo, nada para otros.

d)chown root: root Prueba intenta cambiar el propietario de PRUEBA al root, pero es no se puede ya que no se está invocando desde un root y solo éste puede utilizar chown.

e)chmod 777 PRUEBA cambia los permisos de PRUEBA haciendo que todos puedan leer ejecutar y escribir.

f)chmod 700 etc/passwd , no lo permite ya que no es el root y no puede dejar a otros usuarios sin permisos para lo que no es su directorio.

g)passwd root no me deja manipular la contraseña porque no soy el root.

h)rm PRUEBA elimina el archivo prueba.

i)

man /etc/shadow: intenta abrir shadow (su manual) pero no lo permite porque es un archivo del root.

j) **find / -name \*.conf**: Busca todos los archivos en / con terminación “.conf” y no encuentra ninguno.(entre los que tenga permiso)

k)usermod root-d /home/newroot -L no encuentra la orden

l) cd /root quiere ir al directorio /root permiso denegado.

m) **rm \***: remueve todos los archivos pero no los directorios.

n)cd /etc: va al directorio /etc

o) cp \* /home -R home-R no es un directorio- intenta copiar todos los archivos de configuración, pero tiene El acceso denegado a todos ellos.

p)shutdown no hace nada no encuentra la orden.

8.

a) kill 23

b) killall init no se permite la operación

c)find /home .conf , no existe el fichero.

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

e)chmod 751 procesos

f)chmod 650 procesos

g) rm -R /tmp

h)chown iso2010 /opt/isodata

i)pwd> /home/duilio/donde

9.

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

b)lo llame dr

c)se creo el directorio /home/dr, se actualizaron el /etc/shadow y el /etc/passwd, además se creo el nuevo grupo que lo contiene

d)cd /tmp mkdir cursada 2010

e)cp /var/log/\* /tmp/cursada2010

f)chown dr /tmp/cursada2010 chgrp users /tmp/cursada2010



g)chmod -R 723 /home/duilio

h)

i) dr@lihuen

j)ps -l

k)who -q

l) Utilizaremos **echo "mensaje" > destinatario**. Aunque con otros comandos como **wall** y **write** también podrían enviarse mensajes, mediante ficheros.

**m)shutdown [opcion] tiempo [mensaje]**

10)lo llame de dd hice chmod 017

d) puedo crear el archivo leame ya que LEAME es distinto de leame ya que Linux es case sensitive.

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

f)seria find / -name "\*so" --consultar

reveer punto e

11.1) crea el directorio iso

2)no se puede ir a un directorio inexistente, ./ no existe, ps> f0 guarda el estado de los procesos activos en f0.

3) ls > f1 guarda en f1 los subdirectorios y archivos del directorio de usuario.

4) cd / se posiciona en el directorio /

5) echo \$HOME imprime en pantalla el directorio base del usuario, por lo general /home/nombreusuario.

6) ls-l > \$HOME/iso/ls guarda la lista en detalle de los directorios contenidos en / junto con sus permisos etc, en un nuevo archivo llamado ls dentro de el directorio /home/miusuario/iso

7) cd \$HOME; mkdir f2 crea el directorio f2 en el directorio de usuario.

8) ls -ld f2; no se puede porque f2 no es un archivo, sino un directorio.

9)chmod 341 f2 cambia los permisos de f2 para si mismo escritura y ejecución, para el grupo lectura, y para otros solo ejecución.

10)touch dir actualiza la fecha de acceso de el directorio dir.

11)cd f2 se posiciona en f2

12) cd ~/iso se posiciona en el directorio ISO en ~ se tiene lo mismo que en \$HOME?

13)pwd > f3 guarda el directorio actual en f3

14)ps