

“linux es sensible a las minusculas y minusculas, es diferente una mayuscula que minuscula”

el directorio /dev contiene dispositivos (enlaces a lispositivos del sistema (teclado. Discos etc)

“el directorio /etc guarda la configuracion del sistema”

“cat /etc/passwd :estan todos los usuarios del sistema”

“cat /etc/group : nos muestra los grupos del sistema”

“el directorio /bin son ejecutables de los comandos”

“cuando un programa falla deja un archivo en /var/crash”

para eliminarlos “sudo rm /var/crash/\*”

“ls -l /sbin/init : nos muestra el ejecutable que arranca la maquina”

“prioridades en linux van desde -20 al +19

**-20 tiene mayo prioridad**

**“**sudo fdisk -l : este comanod te muestra los discos de la computadora”

los comando sinternos de una shell son los que no tienen ejecutable, los externos son losque tienen ejecutable en el sistema (la terminal(cmd) es una shell bash)

type comando: nos dice si el comando es interno y externo si es externo nos dice deonde esta su ejecutable

los ejecutables de linux son .sh

en el shell scritp poner como header #!/bin/bash

#!/bin/sh es un enlase simbolico a ‘dash’

“sudo passwd” : para cambiar contrasena de root

“ssh [nombre\_usuario@numero\_ip](mailto:nombre_usuario@numero_ip)” : asi se hace acceso por ssh

COMANDOS RASPBERRRY

“sudo raspi-config” : entre a la configuracion de las raspbery

“sudo idcdetect -y 1” : muestra las direcciones de los i2c de la raspberry

**CREAR OTRA CUEN LINUX**

**1.** su: entrar como usuario root (si no hay root usar “sudo su”, contrasena es la de la compu)

2.useradd -m -d /nombre nombre: ponemos el nombre del nuevo usuario,

useradd -m nombre (usar solo este por lo general)

el -m crea su propio directorio

el -d le da un directorio diferente

3.passwd nombre: nos pedira la contrasena del usuario nombre dos veces

4.userdel nombre: borrar al usuario nombre

**DESACTIVAR FIRAWALL**

sudo ufw disable: desactiva el firewall

sudo ufw enable: activa el firewall

**COMANDOS DE LINUX**

pwd: avisa en que librería estamos.

Man: nos da una ayuda sobre el comando que va despues

ls -l: listado detallado

ls -r: ordena los ficheros de forma descendente según orden alfabetico

ls -d: nos muestra los nombres de los directorias sin su contenido

ls -a: nos muestra los ficheros ocultos tambien

los ficheros ocultos empiezan con un “.”

ls -al: es la combinacion de los dos anteriores

ls -S: ordena por tamano del ficheros

ls -t: ordena por tiempo

man -f \_\_\_: nos da una breve descripcion del comando siguiente

file \_\_: nos dice el tipo de fichero es el siguiente

WILDCARDS (metaracteres en ficheros)

\*: sustituye a un grupo de caracteres( ls b\*: nos muestra too los ficheros que empieza por b)

?:solo sustituye a **UN** carácter

[ ]: es “o” (ls [asd]\*:mostrara los archivos que empiecen con a, s o d)

(ls \*[0-2]: los ficheros que terminen en caracteres entre 0 y 2)

(ls \*[a-c]: los ficheros que terminen en caracteres entre a y c)

INFORMACION SOBRE EL FICHERO

-cuando hacemos el comando “ls -l” en la primera columna nos da las espicificaciones del archivos

el primero es el tipo

d: es un directorio

-: son ficheros normales(texto,ejecutable,binario)

l: son enlaces o acceso directo a otro archivo

estos solo estan en el directorio /dev

b: son tipo bloque

c: tipo carácter

los siguientes son los permisos que tienen los ficheros

habla de lo que hay dentro

si es 1 es un archivo (tipo -)

si es tipo – y el numero es mayor restamos 1 y el resultado es el numero de enlaces (fisico o simbolico) apuntandole

si es 2 es un directorio (tipo d)

si es tipo d y el numeor es mayor restamos 2 y el resultadoe s el numero de directorios dentro del mismo

el siguiente es el propietario

es el grupo

feche de modificacion o creacion

nombre

**CREAR FICHEROS**

mkdir: crea dirctorios

mkadir – p dir1/dir2: donde estemos, crea un directorio dir1 y dnetro de este crea dir2

rmdir: borra directorios

rm: borra ficheros

tambien funciona con meta caracteres

rm -r dir1:borra los direcctorios con sus contenidos

touch: crea ficheros si no existe, y si existe cambia la fecha de creacion

echo hola: manda la palabra hola a la salida estandar (pantalla)

echo -ne hola : imprime hola pero no hace enter (backslash \) de salto de lines y lo siguiente se imprime alado

echo -ne ‘hola \r’ : el \r hace que el cursor vuelva al principio

echo prueba > f1.txt: crea el fichero f1 si no existe y **REESCRIBE** en el la palabra “prueba”

echo prueba >> f1.txt: **AUMENTA**  al fichero la palabra prueba

cat f1.txt: muestra el contenido del fichero

cat -n f1.txt: nos muestra enumeando las filas

tail f1.txt: muestra las 10 ultimas lineas del ficheros

tail -n 20 f1.txt: muestra las 20 ultimas lineas

tail -f f1.txt: se mantiene funcionando y muestra todo lo que se vaya anhadiendo al ficheros

head f1.txt: muestra las primeras 10 lineas

head -n 20 f1.txt: muestra las primera 20 lineas

cp (fichero origen) /(directorio destino)/(fichero destino): copia un fichero si se le da ruta de destino, o si se le da otro nombre solo lo copia con el nuevo nombre

cp -i: pregunta si quiere sobreescribir en caso de ya existir un fichero con ese nombre

cp f\* (destino): copia todos los ficheros que enpiecen con “f” al destino

funciona con los metacaracteres

cp -r (directorio origen) (directorio destino) : con -r copia todo un directorio y lo que esta dentro

mv: mover o renombrar ficheros

\*: tambien funciona con metacaracteres

**EDITOR DE TEXTO**

less f1..txt: entra en un editor de texto

ES SOLO DE LECTRURA

q:sale de less

**INODOS**

son numeros que tienen los datos de un ficheros

tamanho

direccion de datos

usuario

grupo

ls -li f1.txt: nos dice el numero de inodo

enlace fisico (hard):es como una copia con otro nombre del mismo ficheros

el fichero sigue vivo si esq no se borran todos los enlaces

los enlaces se van actualizando al mismo tiempo

los enlaces creados son ficheros de tipo normal (-)

ln f1.txt f1\_enlace.txt: crea un enlace tipo fisico que se cambia uno de ellos

enlace sibolico (soft): es como un acceso director

si se borra el original, los enlaces ya no funcionan

los enlaces se van actualizando al mismo tiempo

los enlaces creados son de tipo link (l)

este enlace creado adquiere un nuevo inodo

ln -s f1.txt f1\_enlace.txt: crea un enlace tipo simbolico

**TAMANO DE FICHEROS**

du dir: te da el tamano de los subdirectorios dentro de dir

du -a dir:saca el tamano de cada uno de los ficheros dentro de dir

du -s dir: saca el tamano de los directorios directos sin entrar en cada uno de ellos

du -am -t 1m:pregunta por todos los ficheros que sean mayor a 1mega

**USAR MAS DE UN USUARIO (**terminales virtuales)

ctrl + alt + F3: entrar a otro usuario

ctrl + alt + F2: volver al usuario original

who:nos dice quien esta conectado a nuestro usuario en ese momento

**EDITOR DE TEXTO**

gedit nombre.txt: crea un archivo de texto con el nombre.txt

**BUSCAR FICHEROS EN EL PAD DEL SISTEMA**

which fichero: nos da la ruta a la ubicación de este fichero en el sistema

whereis fichero: nos avisa donde esta el ejecutable, manual y las librerias que usa

**CREAR COMANDOS**

alias nombre=’comando’: crea un comando con el nombre que hace lo que se ponga en comandos

alias nombre: nos dice que comandos hace el alias nombre

**TIEMPO**

time comando: nos dice cuanto tiempo le toma el proceso del comando

**calendario**

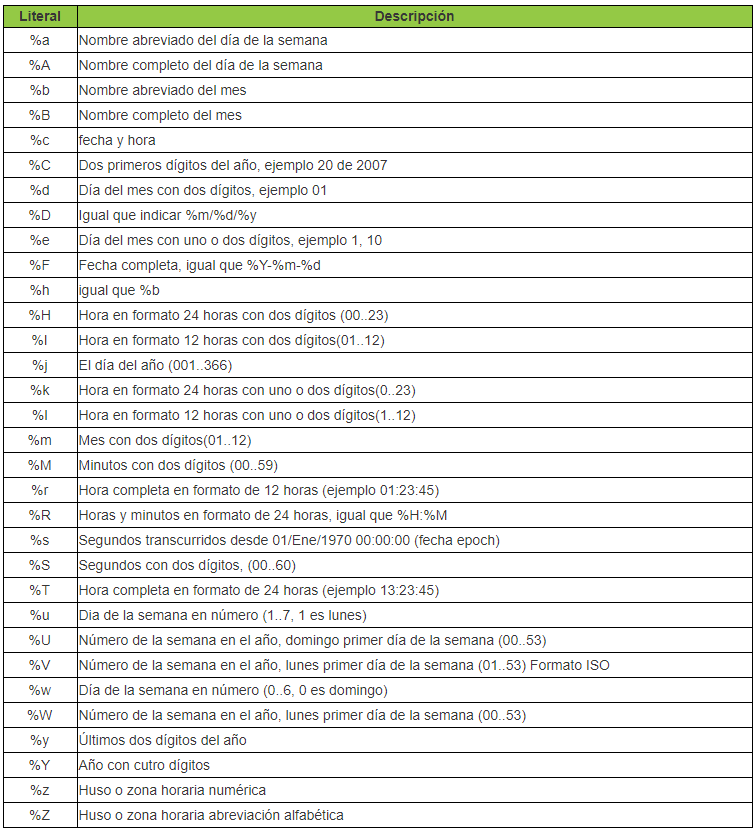
cal: nos da la fecha

cal -m 8 1999: nos da el mes 8 del ano 1999

ncal: formato anglosajon (desde lunes)

date: nos da dia, hora, fecha

date +'hoy es el dia %d del mes %B del ano %Y y la hora de %k:%M:%S'



uptime : nos da datos del funcionamiento

uptime -p: tiempo q paso desde que se arranco la maquina

uptime -s: momento esacto q paso desde que arranco la maquina

**ENTRADAS Y SALIDAS**

STDIN(0)=entrada desde el teclado

STDOUT(1)=salida en la pantalla

STDERR(2)=salida de errores

tr a o:cambia la “a” por la “o”

tr aeiou AEIOU f1.txt : sustituya la ‘a’ po ‘A’, ‘b’ por ‘B’ y asi carácter por carácter

tr a-z A-Z f1.txt : convierte todo en mayuscula

tr -d a f1.txt : borra todas las a del fichero

tr -s a f1.txt : borra todas las a repeticas

tr -c a o f1.txt : todo lo que **no** es a sustituye por o

REDIRECCIONAR ENTRADA

tr a o < f1.txt: cambia las “a” por la “o” en el fichero f1.txt

tr a o << final : recibe palabras del teclado y sustituye las “a” por “o” cuando en lee “final”

en lugar que la entrada sea el teclado, la entrada es un ficheros

wc -l < f1.txt : cuenta cuantas lineas hay en el fichero f1.txt

wc -l << final : cuenta lineas hasta que aparesca la palabra “final”

REDIRECCIONAR SALIDA

echo hola > f1.txt: manda hola a la salida que es el fichero f1.txt

date > f1.txt:manda la fecha a fichero f1.txt

en lugar que la salida sea la pantalla, la salida es un fichero

ping 8.8.8.8 | tee ping.txt : todo lo que salga de ping lo envia a la pantalla **Y** al fichero ping.txt

salida de errores

error 2> error.txt : ponemos el 2 antes del > para q guarde el mensaje de error

error > salidanormal.txt 2> error.txt : la salida estandar lo manda a salinormal.txt y el erro a error.txt

error &> error.txt : manda tanto la salida estandar como el error a error.txt

error > /dev/null :lo borra el mensaje

para guardar un error pon

**TUBERIAS**

cat f1.txt | tr a o : dice que la salida del primer comando lo manda como entrada para el segundo comando

**FILTROS**

uniq f1.txt : filtra todas las lineas **consecutivas** iguales y las vuelve 1 linea

uniq -c f1.txt: nos avisa cuantas veces se repite cada linea

uniq -d f1.txt: nos avica que lineas estan repetidas y nos entrega esas lineas

wc f1.txt: nos cuenta

wc -l f1.txt: nos dice cuantas lineas hay

wc -w f1.txt: cuantas palabras hay

wc -m f1.txt: cuantos caracteres

wc -c f1.txt: cuantso bytes hay

grep palabra f1.txt :busca palabra en fichero f2.txt no puede entrar en directorios y tira warning

grep -s palabra f1.txt: no lansa los errores de los directorios y solo lanza resultados

grep -l palabra f1.txt: solo nos dice en que ficheros hay la palabra

grep -c palabra f1.txt: nos idce cuantas vecces aparece la palabra en cada ficheros

grep -i palabra f1.txt: para no separe mayusculas de minusculas

sort f1.txt:ordene resultados por orden alfabetico

sort -r f1.txt:ordena en forma alfabetica ascendente

sort -n f1.txt: ordena por numeros

sort -k2 : ordena según lo que haya en el campo 2

cut :busca espacios de forma de TAB

cut -f1 fichero.txt: estamos diceindo que nos devuelva el primer campo

cut -d ‘:’ -f1 fichero.txt: nos da el primer campo del fichero.txt

cur -d ‘:’ -f1,6 fichero.txt : nos saca el campo 1 y 6

cut -c 25-29 fichero.txt : le decimos que nos extraiga los datos que estan entre la posicion 25 y 29

diff

**PROCESOS**

jobs: nos muestra los procesos que se estan llevando acabo en el bash actual

jobs -l : nos muestra el identificador del proceso

ps: muestra los procesos que se estan llevando a cabo en el sistema

“espacio PID : es el identifacador del proceso”

“ espacio TTY: es la terminal de la cual se esta ejecutnado”

“espacio CMD : que comadno se esta corriendo”

ps -f : nos muestra los procesos ejecutandose en el bash actual(en esa ventana abierta

“espacio PPID: proceso padre el cual ejecuto este proceso hijo”

“espacio NI: nos dice la prioridad

ps -l : nos muestra mas detalles

“espacio F : nos dice los permisos que tiene el procesos”

“espacio S : nos dice el estado del proceso “

“espacio PRI: prioridad”

ps -ef: nos muestra too los procesos del sistema

ps -ef | grep 1234 : podemos buscar los proecesos por us identificador hasta llegar al padre original

ps -o pid, ni ,comm: nos mues los espacios pid ni y comm

ps -uxf: procesos que se estan ejecutando con el usuario curso

pstree: nos muestra en formato arbol todo los procesos de padre a hijo

pstree -p : nos muestra con los identificadores

pstree -s 123: muscamos toda la jerarquia del proceso con el identificador 123

PRIORIDADES DE PROCESOS

top : nos muestra los procesos en tiempo real

despues de dar top

^M : ordena por memoria cual proceso consume mayor memoria

^P : por consumo de CPU

c : muestra comando completo

CAMBIAR DE PRIORIDADES A PROCESOS

(sleep 5: hace que la shell se quede esperando 5 segundod)

nice -10 sleep 10 & : asigna una prioridad al momento de correr

renice -n 10 1234 : el proceso con el ID 1234 cambia a prioridad 10

MATAR PROCESOS

para matar procesos mandamos senales a un procesos

kill -l: nos muestra las senales que hay en el sistema

man signal: para ver mas comandos de senales

kill -9 1234 : es para matar un procesos 1234 bruscamente

kill -15 1234: termina un proceso, no siempre termina, depende de prioridades

guarda lo necesario y termina el proceso mas limpio

PONER PROCESOS EN SEGUNDO PLANO

sleep 1000 &: pone el proceso sleep en segundo plano

nos entrega el numero de proceso entre corchetes [1] y el identificador del proceso

con procesos que muestran cosas en la pantalla por mas que lo pongamos en backgrpound sige lanzando cosas a la pantalla

du -am /: lanza los tamanos de todos los directorios de todo el sistema

du -am / > f1.txt 2> error.txt & :

du -am / &> f1.txt &

ponemos en segundo plano, y todos los mensajes y errores se guardan en ficheros y ya no nos lanza nada en la pantalla

fg 1 : nos pone en primero plano el proceso con el numero entre corchetes que puesimos en background

**USUARIOS**

id usuario: nos muestra las carateristicas del usuario

**PERMISOS**

acl: permisos de usuarios para usuariso

selinux: permisos para usuarios mas complejos



espacio d : nos dice el tipo de archivos

los 9 siguientes nos dice los permisos del fichero

los 3 primeros: son permisos de usuarios

los 3 siguientes son permisos de grupo

los 3 siguientes son permisos al mundo

son 3 tipos d permiso de **ficheros**

r (read): el fichero se puede leer copiar o vizualizar

w (write): se puede modificar, borrar o cambiar permisos

x (ejecucion): si el fichero es ejecutable

permisos para **directorios**

r : se puede ver el contenido (se le puede hacer ls)

w : se puede crear, borrar o modificar directorios o ficheros dentro de el

lo que este dentro de el puede que no se pueda modificar

x : si puedo entrar en el directorio

CAMBIAR PERMISOS

se peude cambiar de dos formas por carácter o octal

FORMA CARACTER

chmod u+w directorio: cambia los permisos + para para aumentar el premiso y – para quitar

chmod u+x directorio: da permiso de usuario de ejecucio

chmod g+x directorio: cambia permisos para el grupo

chmod o+x directorio: cambia permisos para otros

chmod a+x directorio : cambia permisos a todos

FORMA OCTAL

0=- - -

1=- - x

2=- w -

3=- w x

4=r - -

5=r - x

6=r w -

7=r w x

chmod 124 directorio: nos pones usuario ejecucion, grupo escritura y otros lectura

CAMBIAR PERMISOS POR DEFECTO

umask:nos muestra el umask de nuestra maquina

directorios

maximo valor octal:777

umask:022 (defecto)

777 – 022 = 755= r w x r – x r - x=el octal por defecto para directorios

ficheros

maximo valor octal: 666

umask:022 (defeccto)

666 – 022 = 644= r w – r - - r - -

sacamos umask nuevo para la nueva configuracion que querramos

umask 002: cambia el umask por defecto (022) a 002

CAMBIAR PROPIETARIO DE FICHEROS

sudo chown ususario fichero : cambia nuestro fichero al propiedad del usuario

sudo chgrp grupo fichero : cambia de nuestro fichero al grupo que pongamos

**VARIABLE DE ENTORNO**

printenv : nos muestra las variables de entorno

set : nos muestra las variables de entorno y las de shell

echo$PWD : con el signo de dólar decimos que nos muestra el valor de la variable PWD

$PWD : guarda la direccion donde estamos con la shell

$DISPLAY

$HOME : direccion de home

$SHELL : la tipo de shell que se esta guardando

$LANG : el lenguaje de la

$PATH : muestra los directorios donde encontrar comandos (ejecutables

solo en estas direcciones busca variables

cuando creamos ejecutables los tenemos que agregar la direccion de la carpeta donde esta los ejecutables q creamos

set PATH=$PATH :nueva direcciones

a la variable PATH le guardamos lo anterior juentp nuestra nueva direccion

$PS1 : muestra el pront del bash

$USER : muestra el usuario que somos

FICHEROS QUE CARGAN LOS EJECUTABLES

estas son variables para todo el sistema ( todos los usuarios pueden verlos)

/etc/profile : carga las variables de login al sistema

/etc/bash.bashrc : carga las variables de entorno de bash

/*etc*/bash\_profile : en otros sistemas se puede llamar asi

estas variables solo las puede ver un usuario en especifico,ya que tienen punto al inicio estan icultos

.profile : son las variables de entorno

se ejecuta solo cuando se inicia el sistema

.bashrc : son las variables de bash

se ejecuta cada vez que se abre un bash

**BUSCAR FICHEROS EN EL SISTEMA**

locate f1.txt : busca en una base de datos

/*var/*lib/mlocate/mlocate.db : es una base de datos que actualiza cada dia con los ficheros del sistemas

sudo updatedb : actualiza esta base de datos

locate -b f1.txt : solo busca en ficheros

locate -c f1.txt : nos dice cuantos ficheros o directorios hay con el nombre f1.txt

locate \f1 : busca los nombres solo con el texto f1

find : busca en todo los sistemas de directorios, entonces puede tardar mas que locate

find /directorio\_inicio -name f’1.txt’ : busca el fichero f1.txt a partir del directorio\_inicio

find /directorio\_inicio -not -name ‘f1.txt’ : busco todos los ficheros que no tengan la palabra f1.txt

find /directorio\_inicio -type -f: busca los que son ficheros

-d : directorio

-l : enlaces

c : tipo caracteres

monitor, teclado

b : tipo bloque

disco duro, memoria

find /directorio\_inicio -atime : fecha mas reciente acceso

-mtime : fecha mas reciente de creacion o modificacion

find /directorio\_inicio -atime +2 : hace mas de dos dias

1 : hace un dia esactamente

find /directorio\_inico -size +5k : busca archivos cn mas de 5 kilobytes

c : en bytes

k : en kilobytes

M : en megabytes

G : en gigabytes

b : trozos de 512 bytes

find /directorio\_inicio -user usuario : busca los ficheros pertenecioente al usuario

find /directorio\_inicio -group grupo : busca ficheros pertenecientes al grupo

find /directorio\_inicio -perm 642 : los ficheros que tengan esos permisos en octal

**EMPAQUETADOR**

tar cvf paquete.tar \*: sol empaqueta , pone varios ficheros dentro de uno, pero no comprime

tar cvfz paquete.tar.gz fichero : lo empaqueta todo y ademas comprime con el gzip

tar xvf paquete.tar.gz : descomprimimos y desempaquetamos un fichero que sea .tar.gz

tar tvf paquete.tar fichero : nos lista el contenido del .tar

tar xvf paquete.tar fichero : estraemos le contenido del .tar

con el \* significa que empaquete todo lo que este en nuestro directorio actual

c :crear

x : extraer

l : para listar

v : ver el avance

f : file

t: lista el contenido del .tar

**COMPRESOR**

gzip fichero : comprime **UN** fichero y borra el original y entrega un .gz

gzip -k fichero: comprime y **NO** borra el original

gzip -l fichero : muestra el porcentaje de comprecion

gunzip fichero : descomprime el fichero

gunzip -l fichero.gz : listo el contenido del .gz sin descomprimirlo

bzip2 fichero : comprime ficheros mas grandes, trabaja igual que gzip entrega un .bz2

bunzip2 fichero.bz2 : descomprime el fichero bzip2

zip compresion.zip fichero1 fichero2 fichero3 : comprime todo y no borra los originales

unzip compresion.zip : descomprime el .zip donde estemos

unzip -l compresion.zip : lista el contenido del .zip sin descomprimirlo

**ADMINISTRADOR DE TAREAS**

(free : permite ver la memoria libre)

watch -n 2 ‘comando’ : corre de nuevo el comando que pongamos cada 2 seg

CRONTAB

crontab: configurar trabajos o proyectos en un momento dado en el tiempo

cat /etc/crontab : muestra el archivo crontab donde esta la configuracion de que y cuando ejecutar

este dichero solo lo puede modificar root , nosotros nos tenemos que crear otro crontab

crontab -l : nos muestra los usuarios del crontab

crontab -e crea y modifica el crontab y nos lo guarda en el lugar ocrrecto

crontab fichero\_creado : este comando nos manda el fichero en formato crontab q tengamos

crontab -ri : borra el crontab y nos pregunta si queremos borrarlo

2 variables

SHELL: que shell utilizara para ejecutar

PATH: donde esta el ejecutable a usar

espacio a llenar

m : minuto en el que se ejecutara el script/comando

va del 0 a 59 si ponemos 1,2,3,4 lo hace al minuto 1 2 3 4 si ponemos 1-4 lo hace al minuto

h : la hora del dia que se ejecutable

va del 0 a 23 horas

dom : dia del mes a ejecutarse

mon : mes a ejecutarse

va del 1 al 12

dow : dia de la semana a jecutarse

va del 0 al 6 donde 0 es domingo

user : usuario que ejecutara

solo ponemos si somos root

command : comando o script que se ejecutara

FORMATO PARA EL CRONTAB

# m h dom mon dow user command

\* \* \* \* \* date >> ~/Escritorio/dir1/prueba\_crontab

**GESTION DE USUARIOS Y GRUPOS**

cat /etc/passwd : cada linea es por usuario

campo 1: nombre de usuarios

campo 2 : contrasena encriptada

campo 3 : identificador de usuario (UID)

campo 4 : identificador de grupo (GID)

campo 5 : comentario

campo 6 : directorio de carpeta de usuario personal

campo 7 : identifica la shell del usuario

sudo cat /etc/shadow :

campo 1 : nombre de usuario

campo 2 : contrasena encriptada

campo 3 : caundo se cambio la contrasena por ultima vez

el numero que sale son los dias pasados desde el 1 del 1 de 1970

campo 4 : cuantos dias tengo que esperar para cambiar la contrasena

campo 5 : dias que dura las contrasena

campo 6 : cuantos dias antes que caduque la contrasena nos avisa

campo 7 : dias pasados de la caducidad de la cuenta para que esta se de de baja

campo 8 : fecha de caducidad (poner dias desde el 1-1-1970 para poner fecha)

chage -l usuario : nos muestra los datos de ese usuario legible para el humano

cat /etc/group :

campo 1 : nombre de grupo

campo 2 : contrasena del grupo

campo 3 : identificador del grupo

campo 4 : usarios perteneciente al grupo separados por coma (,)

CREAR GRUPOS

sudo groupadd -g 3000 nombre\_grupo : creamos un grupo llamado nombre\_grupo y con identificador 3000

no menor de 1000

MODIFICAR GRUPO

groupmod -n nuevo\_nombre -g 123 nombre\_grupo

-p : poner contrasena

-g : modificar el identificador del grupo

-n : cambiar nombre del grupo

BORRAR GRUPO

sudo groupdel grupo : borrar grupo

-f: si un usuario tiene este grupo como principal le asigna otro y borra el que queremos

CREAR USUARIO

sudo useradd -u 3000 -d /home/usuario -g 2000 -s /bin/bash -c ‘usuario normal’ nombre\_usuario

-d : decimos cual es el directorio personal del usuarios

-u : identificador del usuario

-g : grupo principal

-G : grupos adicionales

-s : consola shell que suara

-c : comentario

-p : contrasena

passwd usuario: cambia la contrasena del usuario, te pedira la vieja y la nueva contrasena

-l : bloquea al usuarios

-u : desbloquea la contrasena

-s : estado de la contrasena

MODIFICAR USUARIO

sudo usermod : modificar usuario

se pone en lugar de useradd y ponemos lo espacios que queremos cambiar y al final el nombre del usuario del que queremos cambiar

recomendación ; no modificar el id del usuario

BORRAR USUARIOS

sudo userdel usuario: borra el usuario

-r : borrar los directorio y correo

**COMANDO SUDO**

el comando sudo nos permite ejecutar comandos como otro usuario

cat /etc/group | grep sudo : nos muestra los usuarios que pertenecen al grupo sudo y los que estan en el pueden ejecutarse como root

sudo -u usuario2 comando:ejecuta este comando com el usuario

su – usuario2 :crea un nuevo entorno y entras como usuario

su usuario2 : entra como el usuario2 peor arrastra

**EDITOR VI**

distintos tipos de trabajo

modo comando:

ctrl + g : nos muestra el nombre del fichero que estamos y mas cosas

poner numero antes de un comando repite ese numero de veces el comando

moverse con flechas

h: izquierda

j: abajo

k: arriba

l:derechas

w: moverse de palabra en palabra hacia delante

b: palabra a palabra hacia atrás

$ : final de la linea

^ : inicio de linea

ctrl + f: pagina adelante

ctrl + b : pagina atrás

ctrl + d : media paguina adelante

ctrl + u : media pagina atrás

modo edicion:

edita texto

cuadno esty en una linea en blacon

i : entro en modo edicion a la izquierda del cursor

a : entra en modo edicion a la derecha del cursor

escape (esc) : sale del modo edicion

shift + i : entra edicion al principio de la linea

shift + a: entra en modo edicion al final de la linea

o : aumenta linea en blanco debajo donde estemos e inicia edicion ahi

shift + o : aumenta linea en blanco arriba donde estemos e inicia edicion ahi

x : borra hacia delante

shift + x : borra hacia atras

dw : borra desde donde estemos hasta el final de la palabra, para luego pegar

dd :suprime toda la linea, para luego pegar

u : deshace la ulitma operación que hicimos

r +letra: sustituye la letra donde estemos por la letre que ponemos en el comando

shift + r : sobre todo lo que este por lo que escribamos depues

salimos con escape (esc)

yy: copia toda la linea

yw: copia desde donde estamos hasta el final de la palabra

yb ; copia desde donde estemos hasta el inicio de la palabra

p : pega lo copiado justo debajo donde estemos

shift + p : pega encima de donde estemos

se abre en modo comando

/ palabra: ponemos lo que queremos buscar y damos a enter y el cursor se va a la primera coninsidencia que encuentre con el texto que pusimos

n : va a la siguiente coinsidencia

shift + n : busca la anterior coinsidencia

MODO XE : se entra con :

me situo en la liena que quiero sustituir y apreto : y salgo con u

:s/texto original/texto nuevo/

sustituye la primera coinsidencia en la linea

:s/texto original/texto nuevo/g

sustituye todas las coincidencias en toda la linea

:%s/texto original/texto nuevo/

sustituye la primera coincidencia de cada linea en todo el texto

:%s/texto original/texto nuevo/g

sustiruye toda las coincidencias en todo el texto

:r fichero: nos vamos a una linea en especifico del fichero acutal y hacemos este comando

inserta el contenido del fichero desde desde la linea

vi -O fichero fichero2 fichero3 : abre los tres ficheros en la misma ventana, separandolos por una linea bertical

vi -o fichero fichero2 fichero3 : abre los tres ficheros en la misma ventana, separandolos por una linea horizontales

ctrl + w + flecha : cambia de fichero a fichero

:w

graba todas las modificacione

:q

sale del VI si no tenemos modificaciones por guardar

:q!

Salir de VI sin guardar

:wq

guarda y sale de VI

:wq nuevo nombre

guarda con el nuevo nombre y sale de VI

:x

guarda y se sale de VI

RECUPERAR FICHEROS

cuando el vi se sale corruptamente, por fallos u otro error sin guardar

se genera un “.fichero.swp” con el mismo nombre del fichero que perdimos pero oculto

si intento volver a entrar al fichero nos lanza una advertencia y nos ofrece recuperar la ultima version **guardada** de ese fichero

:recover

recupera la ultima **modificacion** hecha en el fichero

despues es recomendable guardarlo una vez recuperado

despues tendemos que borar el .fichero.swp para que no haya problemas despues

OTROS COMANDOS EN VI

:set nu

activa la numeracion de lineas

:set nonu

desactiva la numeracion de lineas

:123

va a la linea 123

shift + g

va a la linea final del fichero

shift + j

la linea de abajo la pone despues de la que estamos

.

repite el ultimo comadno ejecutado

**SHELL SCRIPT**

**. ./**script.sh : el punto espacio path de un script hace que recuerde las variables del script despues de ejecutar el mismo, y no las borre y olvide al terminar de ejecutarse dicho script

# : comentar una linea

#! : se pone esto en la primera linea para especificar el tipo de shell que correra el programa

#!/bin/bash : el programa se correra en la shell bash

#!/bin/ksh : con la korn shell

#!/bin/csh : con c shell

CAMBIAR LA VARIABLE $PATH

vamos al directorio principal (ponemos cd,nada mas)

ls -l .profile :

este es el fichero de ejecucion del usuario especifico de variables del sistema que se ejecuta 1 vez cuando se enciende la maquina

entramos a este fichero y **al final de todo el fichero** ponemos **:**

PATH = $PATH:(nueva doreccion)

hace que se aumente la nueva direccion a lo que ya hay

guardamos y salimos del fichero .profile y para que se ejecutenestos cambios tenenmos que reiniciar la maquina.

si solo actualizamos la variable $PATH como comando en la terminal, esta variable se guardara asi solo para esa ventana de terminal

es necesario anadir la ubicación (path) de donde estan los programas de mayor uso de lo contrario para correr un programa tendria que:

opsion 1:

poner la direccion completa del ejecutable para correrlo

/home/escritorio/programas/ejecutable.sh

opsion 2:

ir al directorio donde esta el ejecutable y poner

./ejecutable.sh

export variable= valor

para que el valor de la variable se reconosca en todo el sistema ponemos export

echo ${variable}hola

si quiero imprimir algo alado de la variable (sin espacio) ponemos la variable entre corchetes para que reconosca la variable

expr 1 + 2

hace que lo que esta a lado es numero pero tenenemos que separar cada parte por espacio

poner con espacios

let v3=10+20

guarda en v3 el resultado numerico de la operación

v1=10

v2=20

echo $v1 + $v2

no funciona porque linux lo reconoce como caracteres

imprime: 10 + 20

echo $(($v1+$v2))

poner asi para que reconosca a las variables como numeros

x=$(ls -l /var/lib)

guarda todo el resultado del comando en la variable x

x=`ls -l /var/lib`

guarda todo el resultado en el comando en la variable x

echo $? : nos muestra 0 si el **programa inmediato anterior ejecutado** es ejecuto bien

nos devuelve 2 si el comadno anterior se ejecuto mal

resultado=$?

Se lo puede guaradar en ua variable

CASE

case valor in

patron1)

comandos

;; : esto significa salte del case

patron2)

comandos

;;

\*) : esto significa default (q si no se cumple ninguna de las anteriores viene aquí

comandos

;;

esac

COSAS PARA PONER EN patron

[ABC]) : se refiere el valor es alguno de los caracteres A B o C

en esta caso puede entrar las entradas A B o C

[0-9]) : entra a este caso cuando la entrada esta entre 0-9

??) : en este caso estamos diciendo que es una **palabra de dos letras**

\*a) : entra en esta opsion cuando la entrada sea lo que sea que termine en ‘a’

x|X) : es cuando la entrada es ‘x’ o ‘X’

CONDICIONES

#

if [condicion] test

then

comando a ejcutar

elif [ codicion ]

then

comandos a ejecutar

else

comando alternativa

fi

#

test

-d fichero : si el fichero existe y es un directorio devuelve true

-e fichero : si el fichero existe, y es el tipo que sea devuelve true

-f fichero : si dichero existe y es un fichero normal devuelve true

-r fichero : si fichero existe y se puede leer devuelve true

-s fichero : si fichero existe y tiene tamano mayor a cero devuelve true

-w fichero : si fichero existe y se puede escribir en el devuelve true

-x fichero : si fichero existe y es ejecutable (en permisos) devuelve true

-L fichero : si fichero existe y es enlace simbolico devuelve true

COMPARACION DE CADENAS

test condicion : si queremos comprar

test string1 = string2

esto debuelve 0 si es verdadero y 1 si es falso

string : el string es nulo (\*\*)

-z string : la longitud del string es 0

-n string : la longitud del estring no es 0

string1 = string2 : los estring son iguales

string1 == string2

string1 != string2 : los string son distintos

ENTEROS

int1 -ep int2 :enteros son iguales

int1 -ne int2 : enteros son distintos

int1 -gt int2 : int1 es mayor que int2

int1 -ge int2 : int1 es mayor o igual que int2

int1 -lt int2 : int1 es menor que int2

int1 -le int2 : int1 es menor o igual que int2

OPERADORES LOGICOS

operation test []

and -a &&

or -o ||

not ! !

LEER DEL TECLADO

read variable : recibe del teclado y lo guarda en una variable

read -p ‘entroduce el valor de la variable’ variable

imprime un mensaje y recibe valores del teclado

read -d: var1 : este comando lee hasta encontrar el carácter ‘:’ o el que se ponga

read -s var1 : hace que lo que se escribe no se muestre en pantalla

**BUCLE FOR**

for variable in lista

do

comandos que utilizan variables

done

QUE PONER EN LISTA

\* : hace el comando ‘ls’ y usa esa lista

s\* : pone en la lista todos los ficheros con nombres que empiecen en ‘s’

{1..5} : la lista son numeros del 1 al 5

{a..z} : la lista son letras desde la a hasta la z

$(seq 1 4) : para listas numericas poner asi

$(seq 1 $(($variable))) : cuando un limite es una variable

**BUCLE WHILE**

while condicion

do

comandos

done

**BUCLE UNTIL**

until condicion

do

comandos

done

**PARAMETROS**

$1 : este el primer parametro que se pone al momento de correr el programas

$2 $3 : y asi en orden según se ponga los parametros

$0 : es un parametro reservado, en esta variable se guarda la direccion del programa (path completo)

$# : aquí se guarda la cantidad que le damos al programa

**CURSOS DE LINUX DE NETACAD**

aptitude : aptitude es un gestor de paquetes

aptitude moo : nos muestra easter eggs

EXPRESIONES REGULARES

. : (es un punto) cualquier individual

[ ] : cualquier carácter especificado

[^ ] : no el carácter especificado

\* : cero o mas d ellos caracteres previos

^ : si ^ es el primer carácter en el patron, entonces el patron debe estar al principio d el alinea que debe coincidir, de lo contrario solo un literal ^

grep

$ si $ es el ultimo carácter en el patron, entonces el patron debe estar al final de la linea que debe coincidir, de lo contrario solo un literal $

EXPRESIONES EXTENDIDAS

+ : uno o mas d ellos patrones previos

? : el patron anterior es opcional

{ } : especificar las coincidencias minimas, maximas o exactas del patron anteior

| : (es una barra vertical) alternancia

( ) : permite crear grupos

COMANDO :dd

dd : este comando es muy poderosa y delicada

**leer mas al respecto**

sudo shutdown : este comando apaga la computadora

shutdown [opciones] hora [mensaje]

**de manera segura**

avisara a todos los usuarios que la maquina se apagara en 5 min

si o si hay que poner un argumetno de hora

shotdown now “adios a todos”

ifconfig : da detalles de red

hostame : nos da el nombre de hostname

hostaname -I : nos da nuestra ip

iwconfig : se especializa en las conecciones inalambricas

ADMINISTRADORES DE PAQUETES

ej.-

ubuntu, debian, red hat

apt-cache search palabra\_clave

busca dentro de los paquetes con

apt-get install nombre\_paquete

este comando instala el paquete en caso de que no lo tengamos

en caso de que ya tengamos el paquete **lo actualiza**

apt-get update

actualiza los cache de los paquetes

es decir actualiza **la lista de los paquetes que tenemos disponibles para descargar**

esta lista se actualiza con servidores de internet

a la version mas reciente

tambien nos muestra donde conseguirlos

apt-get upgrade

actualiza todos los paquetes y sus dependencias a su version mas reciente disponibles

apt-get remove nombre\_paquete

**elimina un paquetes**

deja sus archivos de configuracion en el sistema

apt-get purge nombre\_paquete

**depura el paquete completo del sistema**

elimina sus archivos de configuracion del sistema

apt-get remove –purge nombre\_paquete

esto es util cuando arruinaste la configuracion de una aplicación y quieres iniciar desde cero con la misma aplicación

APRENDIDO POR EXPERIENCIA

“cat /etc/services” : este comando te muestra los puertos usados

**PREGUNTAR PUERTOS**

less /etc/services

-este comando nos muestra que puertos estan asignados a que servicio

sudo netstat -plut

-este comando nos permite supervisar la conexiones de red tanto entrates como salientes

-p :estamos diciendo que nos muestre los procesos que estan utlizando las conexiones

-l : muestra las conexiones en estado de escucha

-u : muestra las conexiones usando UDP

-t : muestra las conexiones usando TCP

sudo ss -plut

-este se usa ahora, **netstat a quedado en desuso**

-los argumentos funcionan igual en ss que en netstat

sudo ufw deny 8000/udp

-con este comando cerramos puertos

sudo ufw allow 800/tcp

-con este comando abrimos puertos al protocolo especificado

sudo ufw deny http

-niega todas las entradas con servicio http

-se puede usar esto tambien en allow

**LIMITADOR DE TRAFICO EN PUERTO**

iptables: es un servicio de linux que nos brinda la posibilidad de administrar el trafico en los puertos

sudo iptables -L

-nos muestra los puertos que esta administrando iptables

sudo iptables -F

-borra todas las reglas que ayamos puesto con iptables

sudo iptables -A INPUT -p tcp –dport 25 -j ACCEPT

-con este comando ponemos una reglas de entrada para el puerto 25 y a los paquetes que quieran entrar

-con este comando estamos diciendo que todos los **paquetes tcp que ENTREN por el puerto 25 ceran aceptados (**ACCEPT)

-las reglas que se aplicaran al puerto especificado ya sea de entrada o de salida se espicifica en el ultimo comando:

-DROP : seran rechazados

-REJECT: rechaza el paso y le avisa al amisor que se le rechazo

-QUEUE: introduce el paquete a una cola dentro d ella biblioteca **libipq**

-RETURN el paquete vuelve a su origen y deja de cricula por la cadena

-LOG: crea un registro d ellos paquetes que circulan por la cadena

-DNAT: modifica la direccion de destino y puerto tambien

-SNAT: modifica la direccion de origen y su puerto tambien

-MASQUERADE: es una forma de SNAT y permite mas restricciones

sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT

-este comando dice que todos los paquetes que entren por eth1 saldran por eth0

**VER IP PUBLICA**

sudo apt\_get update

sudo apt install curl

-necesitamos esta librería

curl ifconfig.me

-este comando nos devolvera en la ip publica