

SISTEMAS OPERATIVOS

TEMA 1

1.2

Usermod: modificar cuenta de usuario

Userdel: eliminar una cuenta de usuario (por defecto no borra el hombre)

Newusers: crea cuentas de usuario utilizando la información de un archivo de texto.

Este debe tener el formato de /etc/passwd

system-config-users: herramienta en modo gráfico

Utilizando el directorio /etc/skel --> archivos de configuración del shell --> en el HOME asignado a la cuenta creada.

- bash_profile: se ejecuta al hacer login (alias , variables , configuración del entorno)

- bashrc : se ejecuta cada vez que se ejecuta una shell

- bash_logout: al salir del sistema (script, programas , etc) .

1.3

El rastro que dejan los usuarios eliminados es el directorio HOME.

En el directorio encuentro como archivos ocultos los archivos anteriores.(.bash_profile , etc)

1.4

(nombre cuenta, password , user ID, group ID , user id info, directorio home, shell)

El propietario es el root , todos pueden leerlo y el root además puede escribir.

1.5

El etc/shadow tiene esos permisos para evitar que alguien pudiese sacar tu contraseña mediante la fuerza bruta. Es decir, teniendo la encriptación de tu password y teniendo un buen hardware podría probar con miles de contraseñas hasta conseguir la misma combinación de caracteres en la encriptación y por tanto averiguarla sin problemas.

1.6

La información que devuelve **id** en modo root es:

(userid)uid= 0

(groupid)gid = 0

grupos a los que pertenece = 0 (root)

TEMA 2

2.1

`fdisk /dev/loop0`

comandos: m(muestra el interfaz), n(nuevo), w(escribe y sale)

`mke2fs -L LABEL_ext3 /dev/loop0` //añade la etiqueta al volumen

`mkfs.ext3` y `mkfs.ext4` añadiremos la ext3 o ext4 al volumen creado anteriormente.

Utilizamos `tune2fs -l /dev/loopX` para mostrar los SA.

2.4

`mkdir /mnt/SA_ext3`

`mount -rt ext3 /dev/loop0 /mnt/SA_ext3`

2.5

`vi /etc/fstab`

//En los primeros 4 campos metemos en orden:

`LABEL_ext3 /mnt/SA_ext3 ext3 ro,auto`

2.7

Listar todos los paquetes instalados y disponibles -> `yum list installed` y `yum list` (para los disponibles)

Eliminar el paquete instalado que te indique el profesor de prácticas -> `yum remove`

<nombre> Reinstalar -> `yum install <nombre>`

2.8

2.8.1: información general y lista los archivos que contiene un paquete ya instalado con `rpm -qa` podemos ver los paquetes instalados, en este ejemplo:

`openssh-5.5p1-24.fc14.2.i686`

`rpm -qi a openssh* | grep openssh* | sort`

TEMA 3

3.1

1. ¿cuanto lleva en marcha el sistema?

```
[root@localhost ~]# uptime  
12:32:49 up 2 min, 1 user, load average: 0.00, 0.00, 0.00
```

Lleva 2 min (segundo campo)

2. Usuarios trabajando

1 usuario trabajando (Tercer campo) uptime -p

3. Carga media en los ultimos 15 minutos

```
[jota@cvi074148 ~]$ uptime  
18:37:52 up 13 min, 1 user, load average: 0,25, 0,25, 0,16
```

carga media del sistema en los ultimos 15 minutos 0.16

3.2

```
[jota@cvi074148 ~]$ time ps  
PID TTY      TIME CMD  
2908 pts/1    00:00:00 bash  
3467 pts/1    00:00:00 ps
```

```
real    0m0,014s  
user    0m0,002s  
sys     0m0,012s
```

En el siguiente ejemplo hemos ejecutado time con el programa ps, y como podemos observar, primero ejecuta ps y después nos muestra los valores de los distintos tiempos: real, user y sys:

Ejercicio 3.2

a)

```
#!/bin/bash
```

```
i=0
```

```
aux=0
```

```
while [ $i -lt $1 ]; do
```

```
let aux=56*47
```

```
let i=i+1
```

```
done
```

```
echo "El valor de la variable es $i"
```

b)

Se inicia con prioridad 0 en mi caso con renice “valor [-19,20] PID

PR (80) Y NI(0) si le pones 20 PRI(99) NI(19)

si le pones -20 PRI(60) NI(-20)

Ejercicio 3.3

b) que no tienen una terminal asociada

TEMA 4

4.1

USUARIO , TERMINAL ASOCIADO, QUIEN ES SU PADRE

service crond start

service atd start

ps -a -F --> (ALL FORMAT)

	UID	PID	PPID	C	SZ	RRS	PSR	STIME	TTY	TIME	CMD
crond	root	907	1	0	n	m	0	10:27	?	00:00:00	usr/sbin/crond -n
atd	root	903	1	0	n	m	0	10:27	?	00:00:00	usr/sbin/atd -f
	(user)		(padre 1 = root)								(no tiene asociada)

4.2

#!/bin/bash

ls /home > listahome-\$(date +%Y-%j-%T-\$\$)

4.3

• **atq**: lista los trabajos pendientes de usuario, excepto si el usuario es superusuario. En este caso todos los trabajos son listados. Formato de salida: 1 linea por trabajo: jobn number, date , hour , queue , username.

• **atrm**: delete de jobs , identificados por el job number

a)a media noche

at -f archivo.sh 00:00am

[jota@cvi074141 practicas]\$ at -f genera-apunte 00:00pm
warning: commands will be executed using /bin/sh
job 2 at Wed Oct 11 12:00:00 2017

b) un minuto despues de la medianoche de hoy

at -f archivo.sh 00:01am

warning: commands will be executed using /bin/sh

job 4 at Thu Oct 12 00:01:00 2017
[jota@cvi074141 practicas]\$

c) a las 17 horas y 30 minutos d mañana

SI HAN PASADOS LAS 17:30 HOY SEGUIR EL PASO ANTERIO.

SI NO:

at -f archivo.sh 17:30 + 1day

[jota@cvi074141 practicas]\$ at -f genera-apunte 17:30 + 1day
warning: commands will be executed using /bin/sh
job 5 at Thu Oct 12 17:30:00 2017

d) a la misma hora en que estamos ahora pero del día 25 de diciembre del presente año

[[CC]YY]MMDDhhmm[.ss]

at -f archivo.sh -t 12250903

[jota@cvi074141 practicas]\$ at -f genera-apunte -t 12250903
warning: commands will be executed using /bin/sh
job 8 at Mon Dec 25 09:03:00 2017

e) a las 00:00 del 1 de enero del presente año

[[CC]YY]MMDDhhmm[.ss]

[jota@cvi074141 practicas]\$ at -f genera-apunte -t 01010000
warning: commands will be executed using /bin/sh
job 9 at Sun Jan 1 00:00:00 2017

4.4 y 4.5

*NOTA: al lanzar la ejecución asíncrona de una tarea con la orden at NO estamos creando un proceso hijo de nuestra shell, este nuevo proceso NO tendrá como entrada estándar, salida estándar y salida de error estándar los asociados a nuestra consola de terminal.

1. Si el usuario que no tiene permisos de ejecución al mandarlo a la cola de trabajos con at, no se ejecuta, es decir, que hereda los permisos de Escritura, Lectura, Ejecución del padre.

2. Cambiado desde root los permisos, lo dejamos de nuevo en la cola de trabajo, en este caso

genera el archivo, podemos comprobar como no tiene una terminal asignada(?), Y su padre “absoluto” es el root, es decir, no es su padre directo, pero acaba heredando de él(modos de ejecución sudo su Y TAMBIÉN USUARIO). EN AMBOS CASOS EL PADRE DIRECTO ES “SH”

3. La máscara de creación que tiene es heredada del padre, ya que al mandarlo a la cola de forma root hereda el : - rw- r-- r-- y de forma usuario(jota) hereda el: rw-rw-r--

4.6

```
////////////////////////////////////  
#!/bin/bash //  
// //  
//find /home/jota -mtime -1 //  
////////////////////////////////////
```

at -f “nombre del script” at -f archivo.sh horaactual tomorrow

4.8

3 script del tipo: echo Soy el proceso 1,2 y 3 y con el comando: at -f “script” -q c,d,e hh:mm

4.13

Script

```
#!/bin/bash
```

```
path= `/home/SO`  
./$path/recducelista
```