Apuntes Módulo 1 Prácticas

Archivos */etc/passwd* y */etc/shadow*.

En el archivo **passwd** encontramos informacion relevante a los usuarios creados en el sistema y cierta información sobre estos de la forma:

Usuario(Login) : Contraseña : UID : GID : Nombre de usuario : Directorio de trabajo : Intérprete de shell

En el archivo **shadow** se encuentran codificadas cada una de las contraseñas de los usuarios que aparecen en **passwd**, de forma que en el archivo passwd aparecerá una x en el campo de contraseña, indicando que esta se encuentra codificada en **shadow**.

Administración de usuarios.

Es posible añadir, modificar y eliminar usuarios mediante las órdenes:

- **useradd**: Añadir usuarios (Podemos añadir las opciones -p para modificar la contraseña o -d para elegir su /home)
- **userdel**: Eliminar usuarios
- **usermod**: Modificar la configuración de un usuario existente (comparte opciones con *useradd* y añade otras como elegir el nuevo nombre de login con -l)
- **newusers**: Lee un archivo y procede a la creación de usuarios en tanto a la información suministrada.

Si no se especifican las opciones se tomarán los valores por defecto que se encuentan en los archivos */etc/default/useradd* y */etc/login.defs*.

Cualquier usuario puede modificar su propia contraseña con la orden **passwd** ya que no podrá usar las opciones anteriores si no es root

Administración de grupos.

Se pueden administrar los grupos, y obtener información sobre estos mediante las órdenes:

- **groupadd, groupdel, groupmod**: Para añadir,borrar y modificar grupos
- **gpasswd group** : Cambia la contraseña de un grupo
- **gpasswd -a user group** : Añade un usuario a un grupo
- **groups user, id user**: Proporciona los grupos a los que pertenece “user”, además id proporciona su identificación

Diferencia entre *fstab*, *mtab*, *filesystems* y *mounts*.

Los cuatro archivos citazos son archivos FHS que guardan diferente información relevante al mismo tema, la primera diferencia es donde se encuentran los archivos. 1. *fstsab* y *mtab* se encuentran en la carpeta *etc*. 2. *filesystem* y *mounts* se encuentran en la carpeta *proc*.

La informacion que encontramos en cada uno de estos archivos es:

- **Fstab**: Es una lista de los sistemas de archivos que se inician con el arranque del equipo, con una serie de información sobre cada uno de ellos. Aqui es donde el usuario puede editar información.
- **Mtab**: Es una lista de los sistemas de archivos que se encuentran montados. Este archivo no se edita.
- **Filesystems**: Es una lista de todos los sistemas de archivos que se encuentran disponibles.
- **Mounts**: Es una lista de todos los sistemas de archivos que se encuentran montados ya sea manualmente o con el arraque del sistema.

Trabajo con sistemas de archivos

Muchas veces, nos interesará crear una partición, un dispositivo virtual. Para ello, hacemos la orden:

```
mknod /dev/nombre
```

El nombre asignado suele ser /loop?/ con ? un número natural.

Luego, para crear un archivo de X megas, hacemos la orden:

```
dd if=/dev/zero of=/root/nombrearchivo bs=Xk count=10000
```

Ojo con la posición de la X, podemos luego asignar un S.A. a ese archivo, pero antes debemos asociar el dispositivo que habiamos creado con el archivo que hemos creado, para ello realizamos:

```
losetup /dev/loop? /root/nombrearchivo
```

Más tarde podemos comprobar la configuración del disco virtual mediante:

```
fdisk -l /dev/loop?
```

Asignar un sistema de archivos a una partición

Para crear un sistema de archivos en una partición (ext3,ext4) tenemos que usar la orden **mke2fs**. Tiene varias opciones interesantes como:

- **-l** para hacerlo sobre un archivo
- **-L** para establecer una etiqueta a la partición
- **-t *nombre particion*** para establecer si queremos ext2,ext3,ext4.

- **-T *tipo de uso*** Para establecer qué tipo de uso queremos darle al Sistema de archivos.

Configurar parámetros del Sistema de Archivos

Para ajustar algunos parámetros de nuestro S.A. usamos la opción **tune2fs**.

- **-l *dispositivo*** nos da un listado con la información relevante de un S.A.
- **-c max-mount-counts *dispositivo*** Establece el número máximo de montajes que se pueden realizar sin que se realice una comprobación de consistencia del S.A.
- **-L *etiqueta dispositivo*** Pone una etiqueta al S.A.
- **-r *numero*** indica el numero de bloques del sistema de archivos que queremos reservar para uso exclusivo de un usuario
- **-u *username*** indica el usuario que tendrá acceso a ese número de bloques asignado con **-r**

Orden Mount

Nos permite montar un sistema de archivos, con una amplia gama de opciones, de las que hemos utilizado:

- *-o ro*: Monta en sistema de archivos en solo lectura.
- *-o rw*: Monta el sistema de archivos en lectura y escritura.

Un ejemplo del formato utilizado para estas ordenes sería:

```
mount /dev/loop0 -o ro mnt/SA_ext3
```

Debemos tener en cuenta que para que esos sistemas de archivos se ejecuten en el arranque debemos añadirlos a /etc/fstab. añadiendo una linea de la siguiente manera:

```
/dev/loop0 /mnt/SA_ext3 ext3 ro 0 0
```

Ahora, podemos también usar esto para obtener paquetes o archivos de nuestro ordenador(recordemos que estamos en una máquina virtual. Para ello, utilizamos la opción:

```
mount none /mnt -t hostfs -o [carpeta de nuestro PC.]
```

Ahora, para instalar un paquete debemos hacer:

```
rpm -i /mnt/[nombrepaquete]
```

Orden Yum

Los archivos de configuracion de Yum se encuentran en /etc/yum/yum.conf. Yum dispone de una amplica cantidad de acciones distintas, entre ellas estan:

```
#   yum erase [paquete]      Permite eliminar un paquete.
#   yum install [paquete]    Permite instalar un paquete.
#   yum update [paquete]     Permite actualizar un paquete.
```

Con la orden "*yum list installed*" podemos ver aquellos paquetes que se encuentran instalados. Utilizando "*yum erase [paquete]*" podemos eliminar el paquete que queramos y luego utilizando "*yum install [paquete]*" podemos volver a instalarlo.

Orden Rpm

Instalador de paquetes que se encuentran descargados en el equipo. Ordenes útiles:

- *-qa* Lo usamos para listar los paquetes
- **-qli paquete** muestra la información de un paquete
- *rpm* ***-i*** *[direccion paquete]* o *rpm –install [direccion paquete]*, podemos instalar el paquete que deseemos.
- Para postrar la maxima información posible utilizaremos *-v* (muestra informacion relevante)
- *-h* (muestra hash marks). De forma que para instalar el paquete:

```
# rpm -ivh [direccion del paquete]
```

Para desinstalarlo utilizaremos:

```
# rpm -e -v [paquete]
```

Uso de quota

Para activar el uso de quota tenemos que usar:

```
quotaon -a
```

Ahora, si queremos ponerle una cuota a un usuario usamos:

```
edquota nombreusario
```

```
edquota -t
```

Estableciendo así el periodo de gracia para el límite soft (están los límites soft y hard).

- Usando **repquota** podemos ver la estadística de las cuotas para todos los usuarios

Orden uptime, orden w

Las opciones muestran salidas similares. Dan información sobre quién está conectado al sistema y qué están haciendo.

- Con **uptime** sólo da la cabecera, es decir, la hora actual, el tiempo que lleva en marcha la máquina, el nº de usuarios conectados y la caga media del sistema en los últimos 1,5 y 10 mins.
- Con **w** se muestra lo que muestra *uptime* mas qué usuarios están conectados y qué está haciendo cada uno, da más detalles sobre el sistema
- Usando **-h** deja de mostrar lo que mostraba *uptime*

Orden time

Mide el tiempo de ejecución de un programa y muestra un recurso del uso de los recursos del sistema. Mide el tiempo de ejecución real, de usuario y de supervisor, asi que haciendo

T_espera = real - usuario - supervisor

Hallamos el tiempo de espera

Órdenes nice y renice

1. Con **nice** establecemos el valor de prioridad de un proceso a un valor distinto del que tiene por defecto. El rango es [-20-19]. Solo puede hacerlo el usuario root.
2. Con **renice** alteramos el valor de la prioridad de uno o más procesos en ejecución.

Orden pstree

Muestra los procesos en ejecución dibujados en forma de árbol. Opciones:

- **-a** muestra los argumentos de la línea de órdenes
- **-A** dibuja el árbol con caracteres ASCII
- **-h** resalta el proceso actual y sus antepasados. Con **-H** especifica el proceso
- **-l** usa un formato largo
- **-n** ordena los procesos por PID. Con **-p** no muestra prioridad

- **-u** muestra el uid de un proceso hijo si es distinto al de su padre.
- **-Z** muestra información de seguridad

Orden ps

Nos da la siguiente información de los procesos: *User, PID, PPID (identificador del proceso padre), %CPU, %MEM, VSZ (tamaño virtual del proceso en KB), RSS (memoria real usada en KB), TTY(terminal asociado con el proceso), STAT (estado del proceso).*

Se suele ejecutar con las operaciones **-ef**. Con la e especificamos todos procesos en el sistema y f permite mostrar la información completa.

Orden top

Muestra la actividad del procesador en tiempo real, y sobre los procesos que aparecen podemos realizar varias opciones:

- **r** Cambiar la prioridad del proceso
- **k** matar o enviar señal
- **N** ordena por PID
- **P** ordena por CPU
- **A** ordena por tiempo
- **n** cambia el numero de procesos mostrados
- **q** salir

Orden mpstat

Muestra estadísticas del procesador junto con la media de datos mostrados. *sysstat* debe estar instalado en el sistema para poder ejecutarlo. Su sintaxis es(aunque se puede usar solo mpstat y funciona):

mpstat [intervalo] [numero]

Control y gestión de memoria

Orden free

Orden muy ligera que visualiza el uso actual de memoria. Informa de la memoria RAM de la computadora y de la memoria SWAP disponible

- **-b** en bytes, **-k** en kilobytes, **-m** en megabytes, **-g** en gigabytes.
- **-w** Produce un resultado más ancho
- **-c**

- **-l** muestra más detalles de alta y baja memoria

Orden watch

Esta orden ejecuta un comando repetidamente y muestra la salida y los errores. Por defecto, se ejecuta cada dos segundos

- **-n [intervalo]** cambia el intervalo de ejecución. No puede ser menor que 0.1 segundos.
- **-g** Sale cuando la salida del comando cambia
- **-e** se congela cuando hay un error en el comando y para salir se pulsa una tecla

Orden *vmstat*

Supervisa el sistema mostrando información de memoria, de procesos, de E/S y de CPU. El porcentaje de tiempo de CPU del usuario viene en la columna *us*, el de tareas del sistema viene en la columna *sy* y el de no hacer nada en absoluto en la columna *id*

- La columna *r* muestra cuantos procesos hay en cola de ejecución

Ampliación de ls

Con ls, buscaremos ahora mostrar ciertos metadatos de los archivos. Para ello, usaremos las opciones.

1. Opciones de formato de listado largo

- **-l** Larga lista de metadatos de archivo para cada archivo
- **-n** larga anterior pero sin los usuarios ni los grupos.
- **-la** igual que *-l* pero no ignora los archivos ocultos
- **-li** Larga lista incluyendo el campo: número de inodo()
- **-lh** lista larga de los metadatos del archivo con los tamaños listados en Kbytes,Mbytes o Gbytes

2. Opciones extras para el listado de formato largo.

- **-X** ordena alfabeticamente por extensión
- **-t** ordena por fecha de modificación
- **-u** ordena por fecha de acceso
- **-c** ordena por fecha de modificación de los metadatos

También se pueden combinar algunas letras para ofrecer distintas salidas según lo que queramos.

Delante de los permisos, aparece una letra. Esta letra indica:

- **-** es un archivo regular
- **d** es un Directorio
- **l** es un enlace simbólico
- **b** es un archivo especial de dispositivo de bloques
- **c** archivo especial de dispositivo de caracteres
- **p** archivo FIFO para comunicaciones entre procesos

Orden df

Esta orden permite visualizar para cada Sistema de Archivos montado, información sobre su capacidad de almacenamiento total, el usado para almacenar y el espacio libre restante y el punto de montaje en la jerarquía de directorios.

- **-i** sirve para visualizar la información sobre los inodos de cada SA montado.

Orden du

Sirve para poder ver el espacio en disco que gasta un directorio de la jerarquía de directorios y todo el subárbol de la jerarquía que comienza en él.

- La última línea de la salida muestra la cantidad total de bloques de disco que utiliza el subárbol.
- du contabiliza el número de bloques de disco asignados estén o no ocupados.

Creación de enlaces simbólicos o duros

Para crear estos enlaces, previamente usamos la orden:

```
ln [nombre archivo] [nombre enlace]
```

Se crean por defecto enlaces duros. Otras opciones son:

- –**symbolic** crea un enlace simbólico
- **-F** intenta crear un enlace duro a un directorio, aunque seguro que falla incluso con root.
- –**P** hace enlaces duros directamente simbólicos.
- **-v** imprime el nombre de los archivos que tienen un enlace.

Los números que aparecen en la columna anterior al *username* son los valores del contador de enlaces, el número de enlaces duros a archivos para liberar el inodo cuando los nombres de archivo que usan ese inodo se eliminen.

Demonios atd y cron

- El demonio **atd** provoca la ejecución de una orden en un momento de tiempo especificado
- El demonio **cron** sirve para hacer estas ejecuciones pero de forma periódica

Para conocer los PIDs de estos demonios, recordamos que tenemos que hacer respectivamente:

```
ps aux | grep [proceso]
```

Y luego, para conocer información específica (como el padre del proceso, qué terminal tienen asociado, etc) utilizamos

```
ps -p [PID] -f
```

sabiendo su PID con la orden anterior o con *pidof nombreproceso*.

Orden at

La sintaxis completa de esta orden es

```
at [-q queue] [-f <script>] [-mldbv] TIME
```

Al iniciarlo se entra en el menú y es necesario poner una orden y redirigirlo, pues no tiene una terminal de salida asociada. Ejemplo de ejecución

```
at 17:10
at> ls ~ > listahome
```

Para ejecutarlo dentro de X tiempo, podemos poner at now, tomorrow, 3 days... y sumar otro tiempo. Por ejemplo:

```
at tomorrow + 3 days
at now + 1 minutes
```

Si lo ponemos en una hora ya pasada, se ejecuta automáticamente.

- Con la opción **-v** se mostrará la hora en la que será ejecutado el trabajo.
- Para ejecutar un script usamos:

  ```
  at -f <script> TIME
  ```

Tenemos una serie de utilidades para ver los procesos que tenemos hechos con at:

1. **aqt** lista los procesos que tenemos para hacer en cola
2. **atrm** elimina órdenes que se vayan a ejecutar más tarde.

Las salidas estándard(1) y de error (2) pueden ser redirigidas a donde quieras al usar at. Lo hacemos de la siguiente forma dentro del prompt de at:

```
[orden que queramos] . 1>>[directorio salida] 2>[directorio errores]
```

Además, at trabaja con colas de prioridad de la a a la z, podemos mandar un proceso a cualquier cola con *[-q queue]* como ya hemos visto.

Los archivos de configuración /etc/at.deny y /etc/at.allow determinan qué usuarios pueden usar at. El de allow tiene una lista de los usuarios habilitados

Orden batch

Esta orden equivale a *at* pero no especificamos la hora de ejecución, sino que se hará cuando la carga de trabajos del sistema esté bajo cierto umbral que se especifica al lanzara el demonio atd.

El demonio cron

Sirve para ejecutar de forma periódica órdenes en el sistema.

La especificación de las tareas a realizar se hacen con la orden **crontab** y archivos en formato crontab.

Formato de archivos crontab:

Cada línea puede tener el formato:

```
minuto hora dia(mes) mes dia(semana) orden
```

Cada uno de los campos puede tener:

- Asterisco, indicando cualquier valor posible
- Número entero
- Dos enteros separados por un guión(un rango de valores)
- Serie de enteros o rangos separados por coma, activando cualquier valor que aparece en la lista

Orden crontab

Instala, desinstala o lista trabajos que procesará el demonio cron. La sintaxis es:

```
crontab <archivo>
```

Opciones útiles:

- **crontab -r** elimina los procesos que crontab fuera a ejecutar periódicamente
- **crontab -l** da el contenido del archivo cron que se está ejecutando
- **crontab -e** nos lleva a un editor para poder editar el archivo cron.

Variables de entorno

Cron tiene asignadas algunas variables de entorno que podemos usar como se usa una variable en los script normales. Algunas son:

- *SHELL* establecida a /bin/bash
- *LOGNAME*
- *HOME*
- *PATH*

En estas variables no se pueden hacer sustituciones de otras variables, así que no podríamos realizar algo del estilo:

```
PATH=$HOME/SO:$PATH
```