

EXAMEN PRÁCTICAS MÓDULO 1 - ALEON - 2019/20

EJERCICIO 1. [2.5 PUNTOS]

1.1 Crea un usuario nuevo en el sistema con la orden `adduser`, con `username=user1` y directorio `home=home-user1`. Al crear el nuevo usuario, ¿se le ha asignado automáticamente un *password*? Muestra el archivo que demuestra si la respuesta es verdadera o falsa, indicando la línea (registro) correspondiente al usuario. Si la respuesta es falsa, indica la orden que utilizas para asignar el *password* y muestra cómo queda la línea tras la asignación. Indica cuál es el archivo de configuración en el que se especifican los parámetros de envejecimiento del *password*. Cambia el máximo número de días que puede estar el usuario sin cambiar la contraseña a 7 días. Muestra el cambio en el archivo pertinente.

La orden que nos permite hacer lo pedido en primera instancia es:

```
> adduser user1 --home /home/home-user1
```

Para ver que hemos creado el usuario de manera correcta, consultemos el archivo `/etc/passwd`:

```
> cat /etc/passwd | grep user1
user1:x:505:508::/home/home-user1:/bin/bash
```

La `x` del segundo campo de los parámetros asociados al usuario `user1` nos indica que no hay contraseña por defecto, y para ver esta información más detalladamente podemos acudir al archivo `/etc/shadow`:

```
> cat /etc/shadow | grep user1
user1:!!:18190:0:99999:7:::
```

La aparición del símbolo `!!` en el campo de la contraseña nos muestra lo que hemos dicho anteriormente. No hay ninguna por defecto, aunque no podremos acceder desde este usuario sin utilizar permisos de superusuario `su`.

Para cambiar la contraseña del usuario, podemos utilizar la siguiente orden:

```
#> passwd user1
Changing password for user user1.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Los parámetros de envejecimiento de la cuenta se encuentran `/etc/shadow`. Mantiene la misma estructura por campos como ya hemos visto que el archivo `/etc/passwd`. Aquí se muestra el estado actual para `user1`:

```
> cat /etc/shadow | grep user1
user1:dfawef9sdfoasidfhsaodhfaee:18190:0:99999:7:::
```

Para cambiar el número máximo de días que puede estar el usuario sin cambiar la contraseña utilizamos la orden `chage`:

```
> chage -m 7 user1
> cat /etc/shadow | grep user1
user1:dfawef9sdfoasidfhsaodhfaee:18190:7:99999:***7***::: (marcado con *** la solución)
```

1.2 Suponiendo que eres un usuario normal del sistema, ¿sobre qué directorios `<directorios>` de un sistema de archivos conforme con el FHS realizarías la siguiente orden para poder saber si el sistema dispone de la utilidad `wc`?

Nos iríamos a los siguientes directorios:

- **/bin** : Programas del sistema que no son críticos para el arranque.
- **/usr/bin** : Archivos con programas, librerías, código y documentación de paquetes del sistema.
- **/opt** (EN CASO DE POSTERIOR INSTALACION): Carpetas con programas añadidos o configurados posteriormente a la instalación del sistema y que no forman parte de la distribución, también se suele usar como carpeta "otros".

Vamos a realizar la búsqueda:

```
$> ls -l /bin | grep "wc"
(NO ENCONTRADO)

$> ls -l /usr/bin
-rwxr-xr-x 1 root root 18504 ene 22 2018 showconsolefont
***-rwxr-xr-x 1 root root 43200 ene 18 2018 wc*** (AQUI ESTÁ)

$> ls -l /opt
(NO ENCONTRADO, YA ESTABA EN /usr/bin)
```

1.3 ¿Qué significan las opciones `defaults` y `noatime`, las cuales se pueden incluir en el campo opciones de `/etc/fstab`? Muestra las opciones tal y como se describe el manual en línea.

Buscamos en el manual la orden `mount`:

```
$> man mount
#DENTRO DE MAN HACEMOS /defaults Y /noatime. ASÍ, NOS APARECERÁ LA INFORMACIÓN ASOCIADA A ESAS OPCIONES
```

- **defaults** : Use the default options: rw, suid, dev, exec, auto, nouser, and async. Note that the real set of all default mount options depends on kernel and filesystem type. See the beginning of this section for more details.
- **noatime** : Do not update inode access times on this filesystem (e.g. for faster access on the news spool to speed up news servers). This works for all inode types (directories too), so it implies nodiratime.

EJERCICIO 2. [3 PUNTOS]

2.1 Crea un archivo especial de dispositivo de bloques `/dev/loop0` con un tamaño de 25 MB tal como se hizo en la sesión 2. Realiza el formato del dispositivo con `fdisk` indicando una única partición y muestra la información de la partición una vez realizado el formato.

Para crear el dispositivo de bloques utilizamos `mknod`:

```
> mknod /dev/loop0 b 7 0
```

Ahora creamos un archivo de 25MB escribiendo en él ceros:

```
> dd if=/dev/zero of=/root/EXAMEN_25MB bs=2500 count=10000
10000+0 records in
10000+0 records out
25000000 bytes *** (25 MB) copied***, 0.304657 s, 82.1 MB/s
```

Asociamos el dispositivo especial de bloques a este archivo creado

```
> losetup /dev/loop0 /root/EXAMEN_25MB
```

Para crear la partición usamos `fdisk`:

```
> fdisk /dev/loop0
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF
disklabel
Building a new DOS disklabel with disk identifier 0xe46d9a17.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4, default 1): 1
First sector (2048-48827, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-48827, default 48827):
Using default value 48827

Command (m for help): p

***Disk /dev/loop0: 24 MB, 24999936 bytes
255 heads, 63 sectors/track, 3 cylinders, total 48828 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xe46d9a17
```

Device	Boot	Start	End	Blocks	Id	System
/dev/loop0p1		2048	48827	23390	83	Linux***

```
# (LA INFORMACION ENTRE *** ES TEMPORAL, AHORA VAMOS A GUARDARLA)
```

```
Command (m for help): w
The partition table has been altered!
```

Ahora comprobamos finalmente lo que hemos hecho:

```
fdisk -l /dev/loop0

Disk /dev/loop0: 24 MB, 24999936 bytes
11 heads, 3 sectors/track, 1479 cylinders, total 48828 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0xe46d9a17

    ***Device Boot      Start         End      Blocks   Id  System***
   ***/dev/loop0p1        2048       48827       23390    83   Linux**
```

2.2 Crea un sistema de archivos `ext3` en el dispositivo del apartado 1. Copia todos los metadatos de SA que seas capaz de visualizar. Indica los siguientes metadatos en especial:

- **Número de Grupos de Cilindros (o de Bloques).**
- **Número de Inodos totales, inodos por grupo de cilindros y tamaño de inodo.**
- **Tamaño de bloque, número de bloques libres total y número de bloques libres por grupo de cilindros.**

Para crear el S.A tipo `ext3` vamos a usar la orden `mkfs` (específica para S.A de tipo extended).

```
> mkfs.ext3 /dev/loop0
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
6120 inodes, 24412 blocks
1220 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=25165824
3 block groups
8192 blocks per group, 8192 fragments per group
2040 inodes per group
Superblock backups stored on blocks:
    8193

Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 30 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Parte de la información pedida se encuentra justo en lo que nos ha mostrado la orden:

- *Número de Grupos de Cilindros (o Bloques)* : **24412 bloques** que se se estructuran en **3 grupos** de **8192 bloques** cada grupo.
- *Número de inodos totales* : **6120 inodos** con **2040 inodos por grupo**.
- *Tamaño de bloque* : **1024 bytes**.

El resto de la información la encontramos con `tune2fs`:

```
> tune2fs -l /dev/loop0

tune2fs 1.41.12 (17-May-2010)
Filesystem volume name:   <none>
Last mounted on:         <not available>
Filesystem UUID:         39953597-1afe-4c9e-b1a2-63f52ec83879
Filesystem magic number: 0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:     has_journal ext_attr resize_inode dir_index filetype
sparse_super
Filesystem flags:         signed_directory_hash
Default mount options:   (none)
Filesystem state:        clean
Errors behavior:         Continue
Filesystem OS type:      Linux
***Inode count:          6120*** ((TOTAL INODOS))
Block count:             24412
Reserved block count:    1220
***Free blocks:         22403*** ((BLOQUES LIBRES))
Free inodes:             6109
First block:             1
***Block size:          1024*** ((TAM BLOQUE))
Fragment size:          1024
Reserved GDT blocks:     95
Blocks per group:        8192
Fragments per group:     8192
***Inodes per group:    2040*** ((INODOS POR GRUPO))
Inode blocks per group:  255
Filesystem created:      Mon Oct 21 03:28:53 2019
Last mount time:         n/a
Last write time:         Mon Oct 21 03:28:53 2019
Mount count:             0
Maximum mount count:     30
Last checked:            Mon Oct 21 03:28:53 2019
Check interval:         15552000 (6 months)
Next check after:        Sat Apr 18 03:28:53 2020
Reserved blocks uid:     0 (user root)
Reserved blocks gid:     0 (group root)
First inode:             11
***Inode size:          128*** ((TAM INODO))
Journal inode:           8
Default directory hash:  half_md4
Directory Hash Seed:     c99be435-e3cd-4584-8617-8136ee971fc0
Journal backup:         inode blocks
```

(ENTRE *** SE ENCUENTRA TODO LO PEDIDO)

- *Tamaño de inodo* : **128 bytes**
- *Bloques libres* : **22403 bloques**

2.3. Monta el sistema de archivos recién creado en el directorio /mnt/ con las siguientes opciones: solo lectura y toda la E/S sobre el sistema de archivos de realizarse de forma síncrona.

Usaremos `mount` con sus opciones de montaje:

```
> mount /dev/loop0 -r /mnt/ -o dirsync
[ 3334.020000] EXT3-fs: barriers not enabled
[ 3334.020000] kjournald starting. Commit interval 5 seconds
[ 3334.020000] EXT3-fs (loop0): mounted filesystem with writeback data mode

#-r: SOLO LECTURA
#dirsync: TODA LA E/S SOBRE EL SA DEBE REALIZARSE DE FORMA SINCRONA
```

EJERCICIO 3. [4.5 PUNTOS]

3.1 Indica el significado de los siguientes campos mostrados por la orden `top`: 'us', 'ni' y 'si'. Además indica para un determinado proceso de ejemplo cuáles son las columnas que muestran: el tamaño de la memoria RAM que realmente utiliza el proceso y el tamaño de memoria que puede compartir con otros procesos.

Para ello consultaremos el manual

```
$> man top

us : time running un-niced user processes
ni : time running niced user processes
si : time spent servicing software interrupts

# VEAMOS EL EJEMPLO CON UN PROCESO

  PID  USUARIO   PR  NI   VIRT   **RES**   **SHR** S  %CPU %MEM   HORA+
ORDEN
  6850 danielsp  20   0   46872   **4120**   **3432** R  11,8  0,0   0:00.04 top

#LAS COLUMNAS PEDIDAS SON RES (MEMORIA RAM UTILIZADA) Y SHR (MEMORIA COMPARTIDA)
((PUESTO ENTRE **))
```

3.2 Indica la orden (y la información que muestra) que podrías utilizar para conseguir saber el uso de memoria del sistema. ¿Qué orden usarías para saber el uso de memoria de un determinado proceso? Muestra la información para un proceso ejemplo. ¿Qué orden usarías para poder saber la memoria *buffered* y la memoria *cached* por separado?

La orden pedida es `free`, ya que nos permite ver la memoria usada por el sistema:

```
$> free

              total        usado        libre        compartido        búfer/caché
disponible
Memoria:    16332372    1719752    12592884         414728        2019736        13901972
Swap:        2097148           0         2097148
```

Para el uso de memoria de un determinado proceso utilizamos `ps`

```
$> ps -p <PID>
```

Para saber la memoria *buffered* y la memoria *cached* por separado volvemos a utilizar `free` pero con la opción `-w`:

```
$> free -w
```

	total	usado	libre	compartido	búferes	caché
disponible						
Memoria:	16304424	1234612	13545920	366484	67564	1456328
	14421136					
Swap:	2097148	0	2097148			

3.3 Crea un *script* que devuelva exclusivamente la información sobre la cantidad de espacio libre y ocupado tanto de la memoria principal como del espacio de intercambio de tu sistema (Ubuntu).

Nota: Este script ha sido creado por Alejandro Villanueva Prados ([EX MO1 UNDERGROUND](#)).

```
#!/bin/bash
free | awk '/total/{ print "\t" $2 "\t" $3 } /Mem:|Swap:/{ print $1 "\t" $3 "\t" $4 }'
```

La salida del script es:

	used	free
Mem:	4057948	831332
Swap:	492	1358468

3.4 Crea un archivo llamado `log` en tu directorio `home` y consigue que todos los Viernes a las 2 horas y 5 minutos durante los meses de Marzo, Abril y Mayo, se ejecute el guión del apartado anterior y se guarden los resultados en el archivo creado (`log`), de tal forma que se pueda observar la evolución de la memoria. Para verificar que funciona pruébalo especificando "cada minuto".

El archivo pedido es un script de `crontab` como el siguiente:

```
#Nombre: log
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

#minuto hora día-del-mes mes , día-de-la-semana orden
5 2 * 3-5 5 ~/script.sh >> ~/log 2> /dev/null
```

La ejecución de este script sería:

```
$> crontab log
```

