

Laboratorio 4



Apellidos: Moreno Vera

Nombres: Felipe Adrian

Código: 20120354I

Asignatura: Administración de Redes (CC481)

2016 - I

Indice

Actividad 1	(3)
Actividad 2	(4)
Actividad 3	(6)
Actividad 4	(8)
Actividad 5	(9)
Actividad 6	(10)
Actividad 7	(15)
Actividad 8	(20)
Actividad 9	(25)

Actividad 1

1. Información de la cuenta. Entrar en el sistema (gráfico). Obtener la información de la cuenta mediante el comando id (identificador numérico de usuario UID, y grupo GID; así como los grupos a los que pertenece).

Para el entorno gráfico, hacemos en root ... yum groupinstall yum groupinfo 'X Window System' y con el yum update.

```
fapCentOSuser@localhost:/root
[fapCentOSuser@localhost root]$ id
uid=500(fapCentOSuser) gid=500(fapCentOSuser) grupos=500(fapCentOSuser) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

2. Cambiar de cuenta de usuario. Para cambiar de cuenta de usuario se utiliza el comando su:

1. Consultar su página de manual especialmente la opción -l (ó el equivalente -)

```
-, -l, --login
Provide an environment similar to what the user would expect had
the user logged in directly.

When - is used, it must be specified as the last su option. The
other forms (-l and --login) do not have this restriction.
```

2. Cambiar al usuario root. Comprobar la información de este usuario con id

```
fapCentOS 2.0 [Corriendo] - Oracle VM VirtualBox
root@localhost ~]# id
uid=0(root) gid=0(root) grupos=0(root) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

3. Volver al usuario suyo saliendo de la Shell. Comprobar la diferencia en el entorno si se usa su y su - (ó su -l) para cambiar a root.

```
root@localhost:~
[fapCentOSuser@localhost ~]$ su root
Contraseña:
[root@localhost fapCentOSuser]# exit
exit
[fapCentOSuser@localhost ~]$ su -l root
Contraseña:
[root@localhost ~]#
```

su -l, cambia de sesión hacia root, su solamente hace una referencia al usuario root.

3. Otros comandos. Para ver qué usuarios están en el sistema tenemos el comando `w`, además hay algunas variantes de `id` como `whoami`. Probar estos comandos.

```
[fapCentOSuser@localhost ~]$ whoami
fapCentOSuser
[fapCentOSuser@localhost ~]$

[fapCentOSuser@localhost ~]$ w
13:07:55 up 7 min,  2 users,  load average: 0,02, 0,14, 0,11
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU WHAT
root      tty1     -               13:00    2:41   6.25s   6.20s /usr/bin/python
root      pts/0    10.10.3.10      13:04    0.00s   0.01s   0.00s w
```

Actividad 2

1. Comprobar los atributos de los ficheros del directorio `home` de su usuario, `ls -la`. Las propiedades son: <tipo><rw_x_propietario><rw_x_grupo><rw_x_resto>:

```
[root@localhost ~]# ls -la
total 80
dr-xr-x---.  3 root root  4096 ene 21 13:40 .
dr-xr-xr-x. 24 root root  4096 ene 21 15:10 ..
-rw-----.  1 root root  1184 ene 21 12:02 anaconda-ks.cfg
-rw-----.  1 root root  1051 ene 21 15:09 .bash_history
-rw-r--r--.  1 root root    18 may 20  2009 .bash_logout
-rw-r--r--.  1 root root   176 may 20  2009 .bash_profile
-rw-r--r--.  1 root root   176 sep 22  2004 .bashrc
-rw-r--r--.  1 root root   100 sep 22  2004 .cshrc
drwxr-xr-x.  2 root root  4096 ene 21 13:40 fapMeow
-rw-r--r--.  1 root root 24747 ene 21 12:02 install.log
-rw-r--r--.  1 root root  7345 ene 21 12:02 install.log.syslog
-rw-r--r--.  1 root root   129 dic  3  2004 .tcshrc
-rw-----.  1 root root     0 ene 21 13:02 .Xauthority
```

1. - fichero; d directorio; l enlace; c dispositivo caracter; b dispositivo bloque; p FIFO; s socket. Explique cada una de ellas muy resumidamente.

-l muestra un listado en el formato largo, con información de permisos, número de enlaces asociados al archivo, usuario, grupo, tamaño y fecha de última modificación además del nombre.

-d Indica si es directorio.

-c Indica que es un caracter especial file, son un tipo de ficheros que pueden representar puertos seriales, etc.

-s Indica que es un socket un archivo especial de comunicación entre procesos que se diferencia claramente de las pipes por ser bidireccional.

-p Indica que es un pipe, es decir una tubería.

-b Indica que es un block special file, son un tipo de ficheros que pueden representar las particiones de los discos duros, etc.

- indica que es un fichero.

2. r: lectura (4); w: escritura (2); x: ejecución (1) Comprobar los permisos del directorio /etc/sudoers.d (ls -ld) e intentar cambiar a ese directorio. Explique que ocurre.

```
[fapCentOSuser@localhost ~]$ ls -ld /etc/sudoers.d
drwxr-x---. 2 root root 4096 jun 22 2012 /etc/sudoers.d
```

Intentando cambiar ...

```
[fapCentOSuser@localhost ~]$ cd /etc/sudoers.d
bash: cd: /etc/sudoers.d: Permiso denegado
```

No nos permitió cambiar debido a que esa carpeta tiene permisos para ser visto solo por el propietario(que es root) y para el grupo, solo de ejecución.

2. Escribir un script que imprima la frase (“Curso Administración”) que llamaremos mi_echo. Para poder ejecutarlo añadir permisos de ejecución con chmod +x mi_echo.sh

```
[fapCentOSuser@localhost ~]$ nano mi_echo.sh
[fapCentOSuser@localhost ~]$ ls
mi_echo.sh
[fapCentOSuser@localhost ~]$ chmod +x mi_echo.sh
[fapCentOSuser@localhost ~]$ ls
mi_echo.sh
[fapCentOSuser@localhost ~]$
```

3. Los permisos se pueden otorgar de forma selectiva usando la notación octal o la simbólica. Ejemplo, probar las siguientes órdenes (equivalentes):

1. chmod 540 mi_echo.sh

2. chmod u+rx,g+r-wx,o-wxr mi_echo.sh

```
[fapCentOSuser@localhost ~]$ chmod 540 mi_echo.sh
[fapCentOSuser@localhost ~]$ ls -l
total 4
-r-xr-----. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
[fapCentOSuser@localhost ~]$ chmod u+rx,g+r-wx,o-wxr mi_echo.sh
[fapCentOSuser@localhost ~]$ ls -l
total 4
-r-xr-----. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
```

¿Cómo se podrían fijar los permisos rw--w--wx, de las dos formas?
Expóngalas.

Easy, como se ve a continuación:

```
[fapCentOSuser@localhost ~]$ chmod 623 mi_echo.sh
[fapCentOSuser@localhost ~]$ ls -l
total 4
-rw--w--wx. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
[fapCentOSuser@localhost ~]$ chmod u+rw-x,g+w-rx,o+wx-r mi_echo.sh
[fapCentOSuser@localhost ~]$ ls -l
total 4
-rw--w--wx. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
```

4. Crear un directorio y quitar los permisos de ejecución para usuario, grupo y otros. Intentar cambiar al directorio. Para que un usuario pueda cambiar un directorio tiene que tener permisos de ejecución.

```
[fapCentOSuser@localhost ~]$ mkdir meow
[fapCentOSuser@localhost ~]$ ls
meow mi_echo.sh
[fapCentOSuser@localhost ~]$ ls -l
total 8
drwxrwxr-x. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:28 meow
-rw--w--wx. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
```

Luego cambiamos permisos, quitando ejecución para todos, grupo y el usuario. No pudimos acceder.

```
[fapCentOSuser@localhost ~]$ chmod 666 meow/
[fapCentOSuser@localhost ~]$ ls
meow mi_echo.sh
[fapCentOSuser@localhost ~]$ ls -l
total 8
drw-rw-rw-. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:28 meow
-rw--w--wx. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
[fapCentOSuser@localhost ~]$ cd meow/
bash: cd: meow/: Permiso denegado
[fapCentOSuser@localhost ~]$
```

Actividad 3

1. Hay dos permisos de ejecución especiales: set uid, SUID y set gid, SGID. Si un fichero tiene activados esos permisos se ejecutan con la identidad del propietario (o grupo propietario) en lugar del usuario que invoca la ejecución:

1. Listar las propiedades de la utilidad /usr/bin/passwd

```
[fapCentOSuser@localhost ~]$ ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 25980 feb 22 2012 /usr/bin/passwd
```


2. Los permisos SUID se pueden añadir con +s o en el byte más significativo un 4.
Ejemplo añadir los siguientes permisos al script (u+rws,g+rx ó 4750). NOTA:
Aunque los permisos se fijan Linux no permite la ejecución de scripts con SUID.

```
[fapCentOSuser@localhost ~]$ chmod u+rws,g+rx mi_echo.sh
[fapCentOSuser@localhost ~]$ ls -l
total 8
drw-rw-rw-. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:28 meow
-rwSrwx-wx. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
[fapCentOSuser@localhost ~]$ chmod 4750 mi_echo.sh
[fapCentOSuser@localhost ~]$ ls -l
total 8
drw-rw-rw-. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:28 meow
-rwsr-x---. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
```

2. El permiso SGID sobre directorios tiene un significado especial, los archivos creados heredan la propiedad del grupo:

1. Crear un directorio y dar los permisos SGID (g+wrxs, 2770), un 2 en el byte más significativo.

```
[fapCentOSuser@localhost ~]$ mkdir fapMeow
[fapCentOSuser@localhost ~]$ ls
fapMeow meow mi_echo.sh
[fapCentOSuser@localhost ~]$ chmod g+wrxs fapMeow/
[fapCentOSuser@localhost ~]$ ls -l
total 12
drwxrwsr-x. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:37 fapMeow
drw-rw-rw-. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:28 meow
-rwsr-x---. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
[fapCentOSuser@localhost ~]$ chmod 2770 fapMeow/
[fapCentOSuser@localhost ~]$ ls -l
total 12
drwxrws---. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:37 fapMeow
drw-rw-rw-. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:28 meow
-rwsr-x---. 1 fapCentOSuser fapCentOSuser 45 ene 21 13:22 mi_echo.sh
```

2. Cambiar a root (su root) y crear un fichero; ver sus atributos

```
fapCentOSuser@localhost:~
[root@localhost ~]# mkdir fapMeow
[root@localhost ~]# ls -l
total 48
-rw-----. 1 root root 1184 ene 21 12:02 anaconda-ks.cfg
drwxr-xr-x. 2 root root 4096 ene 21 13:40 fapMeow
-rw-r--r--. 1 root root 24747 ene 21 12:02 install.log
-rw-r--r--. 1 root root 7345 ene 21 12:02 install.log.syslog
```

3. Volver al usuario, ¿puede borrar el fichero de root?

```
[fapCentOSuser@localhost ~]$ rm -rf /root/fapMeow
rm: no se puede borrar «/root/fapMeow»: Permiso denegado
```

No se puede, no tiene permisos, solo de ejecución.

3. Finalmente el sticky bit (1 en el byte más significativo, ó `chmod +t`) sirve para permitir únicamente al propietario eliminar un fichero. Suele emplearse en directorios compartidos, e.g./tmp. Comprobar que a pesar de poder escribir en el directorio /tmp no podemos borrar ficheros de otros usuarios.

```
[fapCentOSuser@localhost tmp]$ ls
yum.log
yum_save_tx-2016-01-21-12-26zsYg5J.yumtx
yum_save_tx-2016-01-21-13-05g5e4I5.yumtx
[fapCentOSuser@localhost tmp]$ nano meow
[fapCentOSuser@localhost tmp]$ ls
meow      yum_save_tx-2016-01-21-12-26zsYg5J.yumtx
yum.log   yum_save_tx-2016-01-21-13-05g5e4I5.yumtx
[fapCentOSuser@localhost tmp]$ rm yum.log
rm: ¿borrar el fichero regular vacío «yum.log» protegido contra escritura? (s/n) s
rm: no se puede borrar «yum.log»: Operación no permitida
```

```
[fapCentOSuser@localhost tmp]$ rm meow
[fapCentOSuser@localhost tmp]$ ls
yum.log
yum_save_tx-2016-01-21-12-26zsYg5J.yumtx
yum_save_tx-2016-01-21-13-05g5e4I5.yumtx
```

Actividad 4

1. La orden `umask` muestra los permisos que no se otorgan a un fichero o directorio cuando se crea. Comprobar la máscara por defecto del usuario, crear un archivo y comprobar los permisos con los que se crea.

```
[fapCentOSuser@localhost ~]$ umask
0002
[fapCentOSuser@localhost ~]$ umask -S
u=rwx,g=rwx,o=rx
[fapCentOSuser@localhost ~]$ umask -p
umask 0002
```

Haciendo `touch meowFile` (creando el archivo)

```
-rw-rw-r--. 1 fapCentOSuser fapCentOSuser 0 ene 21 13:51 meowFile
```


2. Modificar la máscara de forma que no se de ningún permiso a “otros” ni permisos de modificación al propio grupo. Comprobar el resultado.

Una máscara establecida a `u=rwx,g=rwx,o=` implica que los nuevos archivos tendrán los permisos `rw-rw----`, y los nuevos directorios tendrán los permisos `rw-rwx---`.

```
[fapCentOSuser@localhost ~]$ umask u=rwx,g=rwx,o=
[fapCentOSuser@localhost ~]$ mkdir foo
[fapCentOSuser@localhost ~]$ touch bar
[fapCentOSuser@localhost ~]$ ls -l
total 20
drwxrwxr-x. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:54 alsa
-rw-rw----. 1 fapCentOSuser fapCentOSuser    0 ene 21 13:56 bar
drwxrws---. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:37 fapMeow
drwxrwx---. 2 fapCentOSuser fapCentOSuser 4096 ene 21 13:56 foo
```

Verificar el archivo nuevo bar, y el directorio nuevo foo.

Actividad 5

1. El superusuario puede cambiar el propietario de un fichero (chown) y del grupo propietario (chgrp):

1. Cambiar a root y crear el directorio `/home/prueba`
2. Fijar el propietario y grupo propietario a su usuario
3. Comprobar el funcionamiento

NOTA: con `chown` se puede fijar ambos usando `<usuario>:<grupo>`,
e.g. `chown root:root /tmp`

```
[root@localhost ~]# mkdir /home/prueba
[root@localhost ~]# ls -l /home/prueba
total 0
[root@localhost ~]# ls -ld /home/prueba
drwxr-xr-x. 2 root root 4096 ene 21 15:25 /home/prueba
[root@localhost ~]# chown fapCentOSuser /home/prueba
[root@localhost ~]# ls -ld /home/prueba
drwxr-xr-x. 2 fapCentOSuser root 4096 ene 21 15:25 /home/prueba
[root@localhost ~]# chown fapCentOSuser:fapCentOSuser /home/prueba
chown: grupo inválido: «fapCentOSuser:fapCentOSuser»
[root@localhost ~]# chown fapCentOSuser:fapCentOSuser /home/prueba
[root@localhost ~]# ls -ld /home/prueba
drwxr-xr-x. 2 fapCentOSuser fapCentOSuser 4096 ene 21 15:25 /home/prueba
```

Actividad 6

1. Abrir el fichero `/etc/passwd` y observar su estructura:

`nombre_usuario:x:uid:gid:información:home:shell`

El campo `x`, sirve para indicar que la información de la contraseña está en el fichero `shadow`. Usando el contenido del fichero `password` y las utilidades de UNIX (práctica 1):

```
dbus:x:81:81:System message bus:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/cache/rpcbind:/sbin/nologin
ntp:x:38:38:/:etc/ntp:/sbin/nologin
saslauth:x:499:76:"Saslauthd user":/var/empty/saslauth:/sbin/nologin
postfix:x:89:89:/:var/spool/postfix:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
abrt:x:173:173:/:etc/abrt:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
oprofile:x:16:16:Special user account to be used by OProfile:/home/oprofile:/sbin/nologin
fapCentOSuser:x:500:500:/:home/fapCentOSuser:/bin/bash
Tiene correo nuevo en /var/spool/mail/root
[root@localhost ~]#
```

1. Listar el nombre (sólo el nombre de la cuenta) de los usuarios definidos.

```
root@localhost:~
[root@localhost ~]# cut -f1 -d ":" /etc/passwd | tail
saslauth
postfix
haldaemon
rpcuser
nfsnobody
abrt
tcpdump
sshd
oprofile
fapCentOSuser
```

2. Determinar el número total de usuarios en el sistema.

```
[root@localhost ~]# wc /etc/passwd
 29   50 1393 /etc/passwd
[root@localhost ~]# fapCentOS
```

2. Observar la shell especial nologin (deshabilitar cuenta 1):

```
[fapCentOSuser@localhost root]$ cat /etc/passwd |head
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
```

Podemos ver usuarios con cuenta bloqueada.(no hay acceso a ellas) se puede bloquear o desbloquear cuentas de esta manera: Para bloquear usuario: `passwd -l username`
Para desbloquear usuario: `passwd -u username`

3. Ejecutar directamente ese comando en un terminal.

```
root@localhost fapCentOSuser]# nologin
his account is currently not available.
root@localhost fapCentOSuser]#
```

1. Copiar el fichero /etc/passwd a /etc/passwd.bck

```
[root@localhost fapCentOSuser]# cp /etc/passwd /etc/passwd.bck
[root@localhost fapCentOSuser]#
```

2. Cambiar la cuenta de usuario para que tenga como shell nologin.

Usar la orden `vipw`

Al usar `vipw`, nos permite modificar el fichero `passwd`, Creamos un usuario provisional, `user1` y le cambiamos a `nologin`

```
n/nologin
fapCentOSuser:x:500:500::/home/fapCentOSuser:/bin/bash
user1:x:501:501::/home/user1:/bin/nologin
```

~
INSERT

3. Intentar entrar en otro terminal (Ctrl_Dcho + F2).

Al intentar entrar a `user1`, nos impide el acceso. Y nos regresa al login.

```
[root@localhost ~]# su user1
su: /bin/nologin: No existe el fichero o el directorio
[root@localhost ~]#
```

4. Restaurar la copia del fichero `passwd`.

```
[root@localhost ~]# cp /etc/passwd /etc/passwd.bck_
```

4. Por defecto en CentOS/RHEL cada usuario se asigna a un grupo propio. Abrir el fichero `/etc/groups` y observar su estructura: `nombre_grupo:x:gid:miembros` separados por “,”

El fichero es `/etc/group`.

```
[fapCentOSuser@localhost root]$ cat /etc/group | head
root:x:0:
bin:x:1:bin,daemon
daemon:x:2:bin,daemon
sys:x:3:bin,adm
adm:x:4:adm,daemon
tty:x:5:
disk:x:6:
lp:x:7:daemon
mem:x:8:
kmem:x:9:
```

5. Cada grupo (usando el sistema de permisos que veremos) implementa un rol. Por ejemplo, el grupo `wheel` tradicionalmente se asocia al grupo de administradores (por su acceso a la orden `su` y configuraciones de `sudo`, más adelante):

1. Añadir nuestro usuario al grupo `wheel`. Usar la orden `vigr`

Usando `vipgr`, modificamos

```
wheel:x:10:fapCentOSuser
```

2. Abrir un nuevo terminal y comprobar el cambio con el comando `id`

```
[root@localhost fapCentOSuser]# id
uid=0(root) gid=0(root) grupos=0(root) context=unconfined_u:unconfined_r:unconfi
ned_t:s0-s0:c0.c1023
```

6. Comprobar los permisos de los ficheros `/etc/passwd` y `/etc/shadow`. ¿Por qué está separada la información en dos ficheros?

Porque a diferencia de `passwd`, en `shadow` están las contraseñas encriptadas de los usuarios.

```
fapCentOSuser@localhost:/home/fapCentOSuser
[root@localhost fapCentOSuser]# ls -l /etc/passwd
-rw-r--r--. 1 root root 1393 ene 21 17:14 /etc/passwd
[root@localhost fapCentOSuser]# ls -l /etc/shadow
-----. 1 root root 957 ene 21 17:10 /etc/shadow
[root@localhost fapCentOSuser]#
```

7. Abrir el fichero `/etc/shadow` y observar su estructura:

`nombre:6sal$hash:ultimo_cambio:min:max:inactiva:deshabilitada`

NOTA: `1` usa MD5, `5` usa SHA-256 en RHEL 5 y `6` usa SHA-512 en RHEL6. La “sal” se añade a la contraseña antes de encriptarla para dificultar diversos ataques.

NOTA: min es el mínimo número de días que debe conservarse la contraseña, max el máximo sin cambiar y deshabilitada el número de días en los que se deshabilitará la cuenta después de que caduque la contraseña.

```
opProfile!!!:16821:!:!:!:!  
fapCentOSuser:!!!:16821:0:99999:7:::  
user1:$6$SrN3jJvP$QQk0ZDmFtzKrZLgzIJxPQir/zLmRI2sYhHUKbHTRfD34npkvvt0pYzqybukrG  
YFWbh2tqeVDDGDw8Qk4XaUg/:16821:0:99999:7:::  
[root@localhost fapCentOSuser]#
```

8. El campo contraseña puede tener algunos significados especiales:

1. En blanco (::), sin contraseña
2. (!,!!,*) cuenta bloqueada (sin contraseña, contraseña caducó, bloqueada)

Hacer una copia de seguridad del fichero /etc/shadow y probar las combinaciones anteriores. Una vez terminado restaurar su contenido.

```
CentOS release 6.7 (Final)  
Kernel 2.6.32-279.el6.i686 on an i686  
  
localhost login: user2  
[user2@localhost ~]$_
```

```
localhost login: user3  
Password:  
Login incorrect  
  
login: user4  
Password:  
Login incorrect  
  
login: user5  
Password:  
Login incorrect
```

Users, con su respectivo cambio en la contraseña.

```
fapCentOSuser:!!!:16821:0:99999:7:::  
user1:$6$SrN3jJvP$QQk0ZDmFtzKrZLgzIJxPQir/zLmRI2sYhHUKbHTRfD34npkvvt0pYzqybukrG  
YFWbh2tqeVDDGDw8Qk4XaUg/:16821:0:99999:7:::  
user2:::16821:0:99999:7:::  
user3!:!:16821:0:99999:7:::  
user4:!:!:16821:0:99999:7:::  
user5*:!:16821:0:99999:7:::
```

9. La configuración y valores por defecto para el mecanismo shadow, se configura en el fichero /etc/login.defs. Abrir el fichero y observar su contenido, especialmente:

1. FAIL_DELAY, LOGIN_RETRIES

No está presente, pero el man indica que FAIL_DELAY es el tiempo de espera en segundos cuando nos muestra un intento de login después de un intento errado; y LOGIN_RETRIES es el máximo número de intentos después de ingresar mal la contraseña.

2. PASS_MAX_DAYS, PASS_MIN_DAYS

PASS_MAX_DAYS : Número máximo de días que una contraseña podrá ser usada.

PASS_MIN_DAYS : Número mínimo de días permitido entre cambio de contraseña.

```
#
PASS_MAX_DAYS    99999
PASS_MIN_DAYS    0
```

3. UID_MIN, UID_MAX

Mínimo y máximo de los rangos de la selección de UIDs

```
UID_MIN          500
UID_MAX          60000
```

4. GID_MIN, GID_MAX

Mínimo y máximo de los rangos de la selección de GIDs

```
#
GID_MIN          500
GID_MAX          60000
```

```
fapCentOSuser@localhost:~
bash: gedit: no se encontró la orden
root@localhost ~]# cat /etc/login.defs
#
# Please note that the parameters in this configuration file control
# behavior of the tools from the shadow-utils component. None of
# tools uses the PAM mechanism, and the utilities that use PAM (
# passwd command) should therefore be configured elsewhere. Refer
# /etc/pam.d/system-auth for more information.
#
# *REQUIRED*
# Directory where mailboxes reside, _or_ name of file, relative
# home directory. If you _do_ define both, MAIL_DIR takes pre
# QMAIL_DIR is for Qmail
#
#QMAIL_DIR      Maildir
#MAIL_DIR       /var/spool/mail
#MAIL_FILE      .mail
#
# Password aging controls:
#
#      PASS_MAX_DAYS    Maximum number of days a password may be
#      PASS_MIN_DAYS    Minimum number of days allowed between p
#      PASS_MIN_LEN     Minimum acceptable password length.
```


Actividad 7

1. El comando `useradd` crea una cuenta y añade las entradas necesarias en `passwd`, `shadow` y `group`, además del directorio de usuario. Opciones importantes para definir la cuenta (ver `man useradd`):

- `-c` comentario (sección información)
- `-e` fecha de expiración
- `-f` días para que se bloquee la cuenta después de que caduque la contraseña
- `-g` grupo principal (por defecto creará uno, ver `USERGROUPS_ENAB` en `login.defs`)
- `-G` grupos adicionales
- `-m` crea el directorio home del usuario
- `-s` shell

Crear varias cuentas de usuario con diferentes opciones. Comprobar el contenido de `passwd`, `groups` y `shadow`. Explique la salida de cada una de ellas.

```
fapCentOSuser@localhost:~  
[root@localhost ~]# useradd -c "usuario 6 xd" -e 2016/01/22 -f 2 -G bin,wheel -s /bin/bash user6
```

El usuario `user6` tiene fecha de expiración el 22 de enero del 2016, 2 días en la opción `-f`, con información de usuario “usuario 6 xd”, pertenece a los grupos `bin` y `wheel` además del grupo principal por defecto `user6` y como shell hacia `bash`.

```
[root@localhost ~]# useradd -c "usuario 7" -f 5 -G wheel -s /bin/sh user7  
[root@localhost ~]# useradd -c "usuario 8" -m -d /home/meow -s /sbin/nologin user8
```

El usuario `user7` tiene como shell por defecto a `sh`.

El usuario `user8` tiene como carpeta home `/home/meow` y shell a `nologin`.

Viendo el `/etc/passwd`

```
fapCentOSuser@localhost:~  
[root@localhost ~]# cat /etc/passwd | tail  
oprofile:x:16:16:Special user account to be used by OP  
fapCentOSuser:x:500:500::/home/fapCentOSuser:/bin/bash  
user2:x:501:502::/home/user2:/bin/bash  
user3:x:502:503::/home/user3:/bin/bash  
user4:x:503:504::/home/user4:/bin/bash  
user5:x:504:505::/home/user5:/bin/bash  
user6:x:505:506:usuario 6 xd:/home/user6:/bin/bash  
user7:x:506:507:usuario 7:/home/user7:/bin/sh  
user8:x:507:508:usuario 8:/home/meow:/sbin/nologin  
user9:x:508:500:usurio 9:/home/user9:/bin/bash  
[root@localhost ~]#
```

Viendo /etc/group

```
bin:x:1:bin,daemon,user6
wheel:x:10:fapCentOSuser,user6,user7
useradd: user user9 already exists
[root@localhost ~]# groups user9
user9 : fapCentOSuser
```

Viendo /etc/shadow

```
user6:!!:16821:0:99999:7:2:16822:
user7:!!:16821:0:99999:7:5::
user8:!!:16821:0:99999:7::
user9:!!:16821:0:99999:7::
```

2. Las contraseñas se pueden asignar con el comando passwd. Un usuario puede cambiar su propia contraseña:

- **cambiar la contraseña de un usuario con ese mismo usuario (passwd, sin opciones).**

Vamos al user 6, y cambiamos la contraseña con el mismo user 6.

```
[root@localhost user6]# passwd user6
Cambiando la contraseña del usuario user6.
Nueva contraseña:
CONTRASEÑA INCORRECTA: Es demasiado corta.
CONTRASEÑA INCORRECTA: es demasiado sencilla
Vuelva a escribir la nueva contraseña:
passwd: todos los tokens de autenticación se actualizaron exitosamente.
[root@localhost user6]#
```

- **poner una contraseña a las cuentas creadas en el ejercicio anterior.**

```
fapCentOSuser@localhost:~
[root@localhost ~]# passwd user7
Cambiando la contraseña del usuario user7.
Nueva contraseña:
CONTRASEÑA INCORRECTA: Es DEMASIADO corta.
CONTRASEÑA INCORRECTA: es demasiado sencilla
Vuelva a escribir la nueva contraseña:
passwd: todos los tokens de autenticación se actualizaron exitosamente.
[root@localhost ~]# passwd user8
Cambiando la contraseña del usuario user8.
Nueva contraseña:
CONTRASEÑA INCORRECTA: Es DEMASIADO corta.
CONTRASEÑA INCORRECTA: es demasiado sencilla
Vuelva a escribir la nueva contraseña:
passwd: todos los tokens de autenticación se actualizaron exitosamente.
[root@localhost ~]# passwd user9
Cambiando la contraseña del usuario user9.
Nueva contraseña:
CONTRASEÑA INCORRECTA: Es DEMASIADO corta.
CONTRASEÑA INCORRECTA: es demasiado sencilla
Vuelva a escribir la nueva contraseña:
passwd: todos los tokens de autenticación se actualizaron exitosamente.
[root@localhost ~]#
```

Comprobar los cambios en el fichero shadow

```
user6:$6$Ag/U8QoG$00LU5znyXXCQySDHrYs4LDtRAwVhZnuhouSKbZWcfLk8hAmA8os7LCAuUyxBGbe5En9
:0:99999:7:2:16822:
user7:$6$6/1bEf8n$c2ryLf2tonHrpRmRr0y.vjXY.BroSVH44Q4vFKa/i9VvB/nIQIE/Q.gMVyW/OCejRrm
:0:99999:7:5::
user8:$6$SZy1hZdP$R5GolkxGVInBcVEsLDV9.T1WtEb1Gi42WEg2x6rlreruMMIu.cKg5SqReVYVTZlKxaQ
:0:99999:7:::
user9:$6$0aqHqqVJ$V4LnAoRIUB0/Sb7FhKKCZ34Ie/r2n3E2UeszTlMV/6ANXd8a8g/0EXSB2BUnbhd2J4m
:0:99999:7:::
[root@localhost ~]# fapCentOS saludos
```

3. El comando groupadd crea nuevos grupos. Crear un par de grupos uno de ellos con el GID 60002.

```
[root@localhost fapCentOSuser]# groupadd meowOS
[root@localhost fapCentOSuser]# groupadd -g 60002 gatoOS
[root@localhost fapCentOSuser]# cat /etc/group | tail
user1:x:501:
user2:x:502:
user3:x:503:
user4:x:504:
user5:x:505:
user6:x:506:
user7:x:507:
user8:x:508:
meowOS:x:509:
gatoOS:x:60002:
```

4. Para modificar una cuenta de usuario se usa el comando usermod:

1. Deshabilitar una de las cuentas creada cambiando su shell.

```
user6:x:505:506:usuario 6 xd:/home/user6:/bin/bash
user7:x:506:507:usuario 7:/home/user7:/sbin/nologin
user8:x:507:508:usuario 8:/home/meow:/sbin/nologin
user9:x:508:500:usurio 9:/home/user9:/bin/bash
[root@localhost fapCentOSuser]#
```

2. Añadir una de las cuentas creadas a uno de los nuevos grupos (notar la diferencia entre -g y -G y la opción -a).

De la misma forma se puede modificar un grupo con groupmod (consultar su página de manual).

-g : Modifica el grupo principal del usuario

-G : Añade al usuario a grupos secundarios

-a : Junto con -G permite que se adicionen grupos complementarios al usuario

```
[root@localhost fapCentOSuser]# usermod -g meow0S user8
usermod: sin cambios
[root@localhost fapCentOSuser]# usermod -G user8 user7
[root@localhost fapCentOSuser]# cat /etc/group | tail
user1:x:501:
user2:x:502:
user3:x:503:
user4:x:504:
user5:x:505:
user6:x:506:
user7:x:507:
user8:x:508:user7
meow0S:x:509:
gato0S:x:60002:
```

```
[root@localhost fapCentOSuser]# usermod -a -G user6 user8
[root@localhost fapCentOSuser]# cat /etc/group | tail
user1:x:501:
user2:x:502:
user3:x:503:
user4:x:504:
user5:x:505:
user6:x:506:user8
user7:x:507:
user8:x:508:user7
meow0S:x:509:
gato0S:x:60002:
```

Viendo el man groupmod

```
GROUPMOD(8)                                System Management Commands          GROUPMOD(8)

NAME
    groupmod - modify a group definition on the system

SYNOPSIS
    groupmod [options] GROUP

DESCRIPTION
    The groupmod command modifies the definition of the specified GROUP by
    modifying the appropriate entry in the group database.

OPTIONS
    The options which apply to the groupmod command are:

    -g, --gid GID
        The group ID of the given GROUP will be changed to GID.

        The value of GID must be a non-negative decimal integer. This value
        must be unique, unless the -o option is used. Values between 0 and
        999 are typically reserved for system groups.

        Any files that have the old group ID and must continue to belong to
        GROUP, must have their group ID changed manually.
:
```

5. Se pueden borrar las cuentas con userdel y groupdel, consultar las opciones (especialmente -r para userdel). Probar estos comandos con algunos de los nuevos usuarios y grupos.

Se eliminan los usuarios user6, user7, user8 y user9 además de sus grupos principales y el grupo gatoOS. La opción -r elimina el usuario, su directorio /home con ficheros dentro y su almacén de correos.

```
[root@localhost fapCent0Suser]# userdel -r user6
[root@localhost fapCent0Suser]# userdel -r user7
[root@localhost fapCent0Suser]# userdel -r user8
[root@localhost fapCent0Suser]# userdel -r user9
[root@localhost fapCent0Suser]# groupdel gatoOS
```

Actividad 8

1. Puede ser necesario permitir el acceso a root al sistema, aunque se puede restringir los terminales desde los que se puede hacer login. El fichero `/etc/securetty` especifica que terminales son seguros para root:

- Hacer una copia del fichero.

```
[root@localhost fapCentOSuser]# cat /etc/securetty
console
vc/1
vc/2
vc/3
vc/4
vc/5
vc/6
vc/7
vc/8
vc/9
vc/10
vc/11
tty1
tty2
tty3
tty4
tty5
tty6
tty7
tty8
tty9
tty10
tty11
[root@localhost fapCentOSuser]#
```

- Dejar solo `tty3` y probar su comportamiento.

```
[root@localhost fapCentOSuser]# cp /etc/securetty /etc/securetty.backup
[root@localhost fapCentOSuser]# nano /etc/securetty
[root@localhost fapCentOSuser]# cat /etc/securetty
tty3
[root@localhost fapCentOSuser]#
```


Vemos que no nos permite ingresar a pesar de haber escrito los datos correctos.

```
CentOS release 6.7 (Final)
Kernel 2.6.32-279.el6.i686 on an i686

localhost login: root
Password:
Login incorrect

login: fapCentOS
Password:
Login incorrect

login: fapCentOSuser
Password:
Login incorrect

login: _
```

Pero en tty1 si da correcto, y ningún otro es correcto.

```
localhost login: root
Password:
Last login: Thu Jan 21 19:41:53 on tty1
[root@localhost ~]# _
```

2. Además de /etc/securetty para root, está el fichero /etc/security/access.conf que configura que usuarios y en que terminales pueden entrar al sistema. Cada entrada determina (+/-) habilita/deshabilita el acceso de un grupo o conjunto de usuarios al sistema desde una terminal o host (-:ALL EXCEPT root:tty1). Observar el contenido del fichero y explicar algún otro aspecto importante que observe.

```
# Disallow non-root logins on tty1
#
# -:ALL EXCEPT root:tty1
#
# Disallow console logins to all but a few accounts.
#
# -:ALL EXCEPT wheel shutdown sync:LOCAL
#
# Same, but make sure that really the group wheel and not the user
# wheel is used (use nodefgroup argument, too):
#
# -:ALL EXCEPT (wheel) shutdown sync:LOCAL
#
```

La primera línea indica que ningún usuario excepto root puede iniciar sesión en tty1.

La segunda línea indica que ningún usuario excepto wheel, shutdown y sync inicia desde el host local.

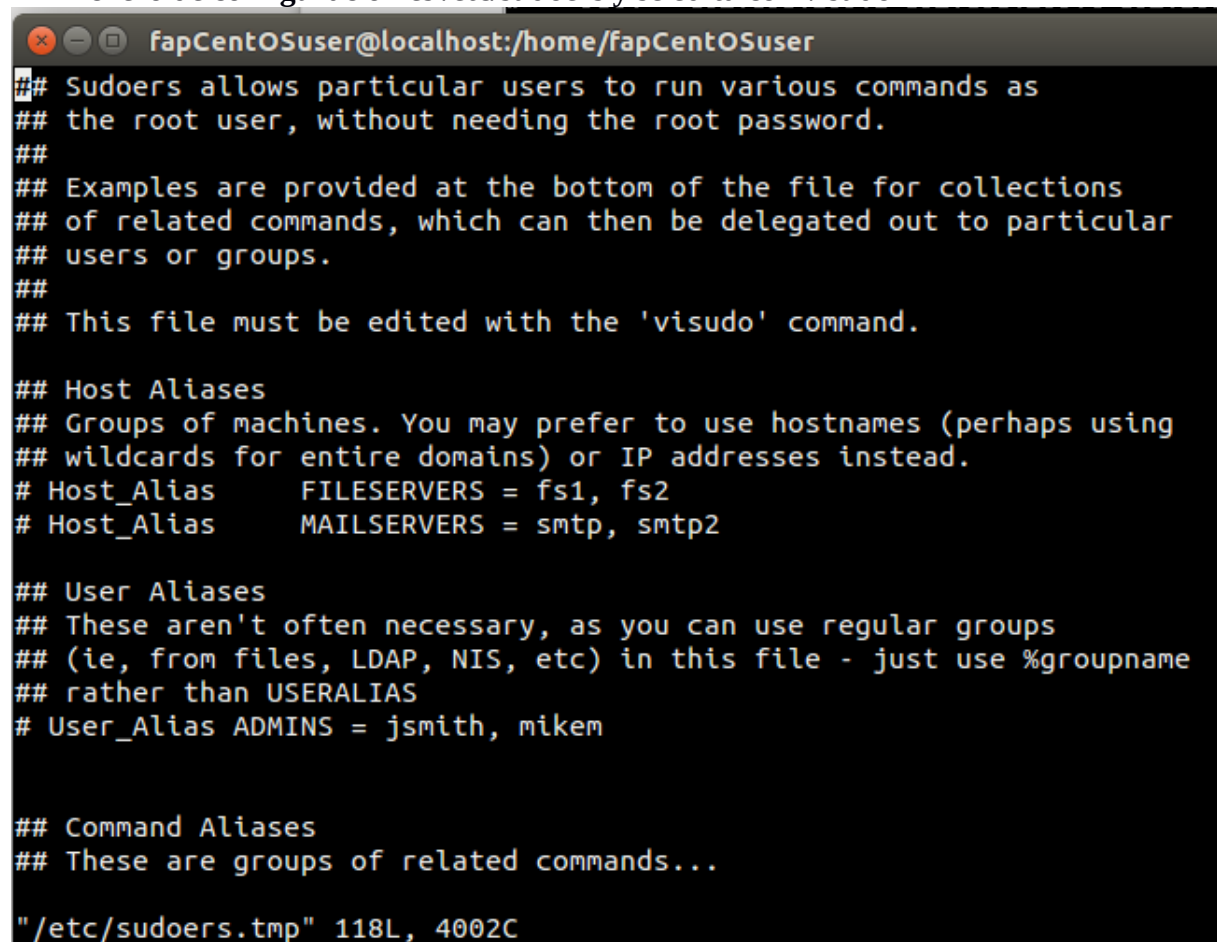
La tercera línea indica que ningún usuario excepto los usuario shutdown, sync y del grupo wheel ingresen desde el host local.

```
# User "root" should be allowed to get access from hosts with ip addresses.
##+ : root : 192.168.200.1 192.168.200.4 192.168.200.9
##+ : root : 127.0.0.1
##
```

Estas líneas indican que el usuario root debería obtener acceso al sistema desde los host descritos en las ip.

3. El comando su, permite cambiar de usuario y requiere conocer la contraseña de la cuenta destino. Normalmente se usa la orden sudo, que permite acceder a los usuarios a comandos de administración con su propia password:

◦ El fichero de configuración es /etc/sudoers y se edita con visudo



The screenshot shows a terminal window with the title bar "fapCentOSuser@localhost:/home/fapCentOSuser". The terminal displays the content of the /etc/sudoers file, which includes comments and configuration for the sudo command. The text is as follows:

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particular
## users or groups.
##
## This file must be edited with the 'visudo' command.

## Host Aliases
## Groups of machines. You may prefer to use hostnames (perhaps using
## wildcards for entire domains) or IP addresses instead.
# Host_Alias      FILESERVERS = fs1, fs2
# Host_Alias      MAILSERVERS = smtp, smtp2

## User Aliases
## These aren't often necessary, as you can use regular groups
## (ie, from files, LDAP, NIS, etc) in this file - just use %groupname
## rather than USERALIASES
# User_Alias ADMINS = jsmith, mikem

## Command Aliases
## These are groups of related commands...

"/etc/sudoers.tmp" 118L, 4002C
```

- Observar el fichero y la sintaxis empleada (usuario máquina=comandos). ¿Qué significan las entrada:

- **root ALL=(ALL) ALL**

```
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
```

Indica que root puede ejecutar con sudo desde el host que sea(eso indica el primer ALL) como todos los usuarios disponibles(eso indica el segundo ALL) todos los comandos disponibles (eso indica el tercer ALL).

- **%sys ALL = NETWORKING, SOFTWARE**

```
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
```

Permite a todos los usuarios del grupo sys ejecutar binarios de NETWORKING y SOFTWARE.

```
## Networking
# Cmnd_Alias NETWORKING = /sbin/route, /sbin/ifconfig, /bin/ping, /sbin/dhclient, /usr/bin/net, /sbin/iptables,
, /usr/bin/rfcomm, /usr/bin/wvdial, /sbin/iwconfig, /sbin/mii-tool

## Installation and management of software
# Cmnd_Alias SOFTWARE = /bin/rpm, /usr/bin/up2date, /usr/bin/yum
```

- **%wheel ALL=(ALL) NOPASSWD: ALL**

```
## Same thing without a password
# %wheel    ALL=(ALL)    NOPASSWD: ALL
```

Al ejecutar sudo, no exige la contraseña del usuario.

- **Dar permisos al usuario para ejecutar cualquier comando sin contraseña.**

```
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)    ALL
fapCentOSuser ALL=(ALL) NOPASSWD: ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more
```

- Comprobar el comportamiento, reiniciando el servicio sshd mediante sudo.

```
[fapCentOSuser@localhost ~]$ sudo service sshd restart
Parando sshd: [ OK ]
Iniciando sshd: [ OK ]
[fapCentOSuser@localhost ~]$ _
```

- Como su usuario cambiar al usuario root usando sudo y la opción -i. Una vez que podemos cambiar a root con su usuario, deshabilitar el acceso con contraseña a root.

```
[fapCentOSuser@localhost ~]$ sudo -i
[root@localhost ~]#
```

```
root:!:16821:0:99999:7:::
bin:*:15513:0:99999:7:::
daemon:*:15513:0:99999:7:::
adm:*:15513:0:99999:7:::
lp:*:15513:0:99999:7:::
sync:*:15513:0:99999:7:::
shutdown:*:15513:0:99999:7:::
halt:*:15513:0:99999:7:::
mail:*:15513:0:99999:7:::
uucp:*:15513:0:99999:7:::
operator:*:15513:0:99999:7:::
games:*:15513:0:99999:7:::
gopher:*:15513:0:99999:7:::
ftp:*:15513:0:99999:7:::
nobody:*:15513:0:99999:7:::
dbus:!:16821:0:99999:7:::
```

Ahora iniciamos sin clave:

```
CentOS release 6.7 (Final)
Kernel 2.6.32-279.el6.i686 on an i686

localhost login: root
Last login: Thu Jan 21 20:20:18 on tty1
[root@localhost ~]# _
```

Actividad 9

1. Consultar el contenido del directorio /etc/skel, que contiene los archivos que se copian cuando se crea una cuenta de usuario (.bashrc, .bash_profile, .bash_logout...) y explicar su contenido.

Cuando se crea un usuario, estos 3 ficheros se agregaran al directorio home del usuario

.bashrc : Se ejecutara este script cada vez que el usuario llame a un subshell (como una shell de usuario, ejemplo, en ubuntu con ctrl + alt + t abre una terminal, esa es una subshell)

```
[fapCentOSuser@localhost ~]$ cat .bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
    . /etc/bashrc
fi

# User specific aliases and functions
[fapCentOSuser@localhost ~]$
```

.bash_profile : Además de .bashrc , también se ejecutara .bash_profile para agregar algunas variables, alias y funciones personalizadas localmente.

```
[fapCentOSuser@localhost ~]$ cat .bash_profile | head
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
[fapCentOSuser@localhost ~]$
```

.bash_logout: Es el fichero que se ejecuta al salir de la sesión.

```
[fapCentOSuser@localhost ~]$ cat .bash_logout | head
# ~/.bash_logout

[fapCentOSuser@localhost ~]$
```

.bash_history: es la lista de comandos que has escrito en una sesión.

```
[fapCentOSuser@localhost ~]$ ls -la /etc/skel/  
total 32  
drwxr-xr-x.  3 root root  4096 ene 21 15:41 .  
drwxr-xr-x. 99 root root 12288 ene 21 20:18 ..  
-rw-r--r--.  1 root root   18 sep 22 11:36 .bash_logout  
-rw-r--r--.  1 root root  176 sep 22 11:36 .bash_profile  
-rw-r--r--.  1 root root  124 sep 22 11:36 .bashrc  
drwxr-xr-x.  4 root root  4096 ene 21 12:57 .mozilla
```

2. El fichero /etc/bashrc contiene definiciones y configuraciones globales, se carga desde la configuración de usuario (.bashrc); estudiar su comportamiento.

```
[fapCentOSuser@localhost ~]$ cat .bashrc  
# .bashrc  
  
# Source global definitions  
if [ -f /etc/bashrc ]; then  
    . /etc/bashrc  
fi  
  
# User specific aliases and functions  
[fapCentOSuser@localhost ~]$
```

En nuestro caso, pregunta al sistema si existe el fichero .bashrc (mediante el if), si existe lo ejecuta, sino, no hace nada.

3. Finalmente /etc/profile y /etc/profile.d contienen la configuración global del entorno. Observar el contenido del fichero profile(PATH, USER, HOSTNAME...) y el contenido de algunos de los ficheros en /etc/profile.d(e.g. colorls.sh).

```
[fapCentOSuser@localhost ~]$ cat /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}
```

```
if [ -x /usr/bin/id ]; then
    if [ -z "$EUID" ]; then
        # ksh workaround
        EUID=`id -u`
        UID=`id -ru`
    fi
    USER=`id -un`
    LOGNAME=$USER
    MAIL="/var/spool/mail/$USER"
fi

# Path manipulation
if [ "$EUID" = "0" ]; then
    pathmunge /sbin
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
else
    pathmunge /usr/local/sbin after
    pathmunge /usr/sbin after
    pathmunge /sbin after
fi

HOSTNAME=`/bin/hostname 2>/dev/null`
HISTSIZE=1000
if [ "$HISTCONTROL" = "ignorespace" ] ; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignoredups
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL
```

Vemos que en la parte de

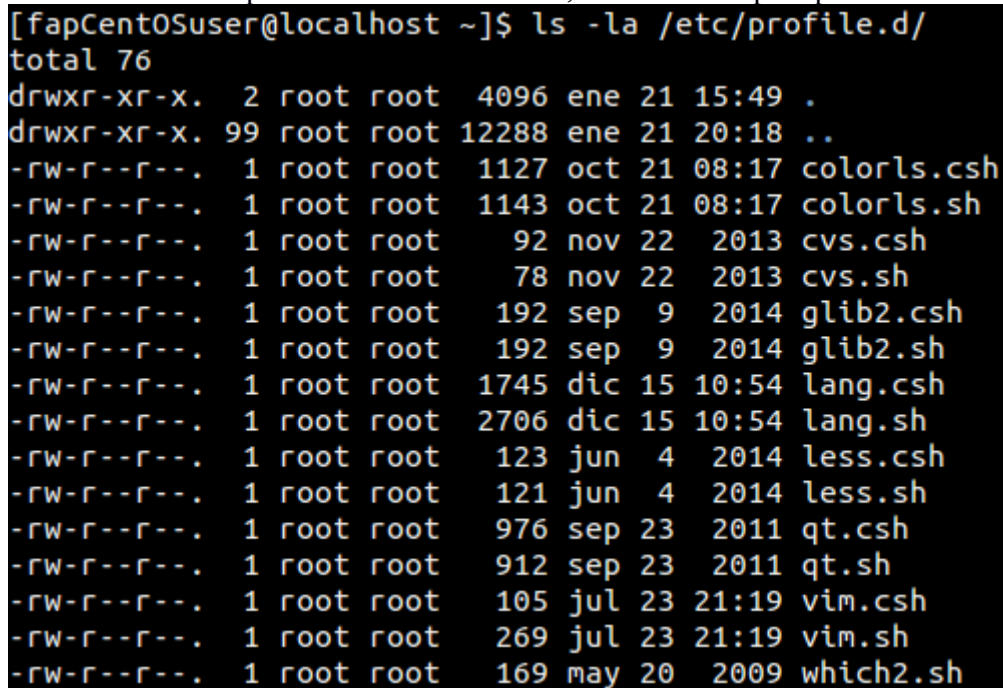
```
if [ -x /usr/bin/id ]; then
    if [ -z "$EUID" ]; then
        # ksh workaround
        EUID=`id -u`
        UID=`id -ru`
    fi
    USER=""`id -un`"
    LOGNAME=$USER
    MAIL="/var/spool/mail/$USER"
fi
```

Configura la variable USER con el nombre de usuario en la sesión.

Además , también define la variable HOSTNAME que configura la variable HOSTNAME con la salida del binario hostname.

```
# Path manipulation
if [ "$EUID" = "0" ]; then
    pathmunge /sbin
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
else
    pathmunge /usr/local/sbin after
    pathmunge /usr/sbin after
    pathmunge /sbin after
fi
```

Crea la variable PATH dependiendo si es root o no, Viendo la carpeta profile.d



```
[fapCentOSuser@localhost ~]$ ls -la /etc/profile.d/
total 76
drwxr-xr-x.  2 root root  4096 ene 21 15:49 .
drwxr-xr-x. 99 root root 12288 ene 21 20:18 ..
-rw-r--r--.  1 root root  1127 oct 21 08:17 colorls.csh
-rw-r--r--.  1 root root  1143 oct 21 08:17 colorls.sh
-rw-r--r--.  1 root root    92 nov 22  2013 cvs.csh
-rw-r--r--.  1 root root    78 nov 22  2013 cvs.sh
-rw-r--r--.  1 root root   192 sep  9  2014 glib2.csh
-rw-r--r--.  1 root root   192 sep  9  2014 glib2.sh
-rw-r--r--.  1 root root  1745 dic 15 10:54 lang.csh
-rw-r--r--.  1 root root  2706 dic 15 10:54 lang.sh
-rw-r--r--.  1 root root   123 jun  4  2014 less.csh
-rw-r--r--.  1 root root   121 jun  4  2014 less.sh
-rw-r--r--.  1 root root   976 sep 23  2011 qt.csh
-rw-r--r--.  1 root root   912 sep 23  2011 qt.sh
-rw-r--r--.  1 root root   105 jul 23 21:19 vim.csh
-rw-r--r--.  1 root root   269 jul 23 21:19 vim.sh
-rw-r--r--.  1 root root   169 may 20  2009 which2.sh
```

Viendo el archivo colorls.sh

```
# color-ls initialization

#when USER_LS_COLORS defined do not override user LS_COLORS, but use them.
if [ -z "$USER_LS_COLORS" ]; then

    alias ll='ls -l' 2>/dev/null
    alias l.='ls -d .*' 2>/dev/null

    # Skip the rest for noninteractive shells.
    [ -z "$PS1" ] && return

    COLORS=

    for colors in "$HOME/.dir_colors.$TERM" "$HOME/.dircolors.$TERM" \
        "$HOME/.dir_colors" "$HOME/.dircolors"; do
        [ -e "$colors" ] && COLORS="$colors" && break
    done

    [ -z "$COLORS" ] && [ -e "/etc/DIR_COLORS.256color" ] && \
        [ "x`tty -s && tput colors 2>/dev/null`" = "x256" ] && \
        COLORS="/etc/DIR_COLORS.256color"

    if [ -z "$COLORS" ]; then
/etc/profile.d/colorls.sh _
```

Indica los colores que usará ls para enlistar archivos, directorios, ejecutables y demás.

Viendo el fichero vim.sh

```
[fapCentOSuser@localhost ~]$ cat /etc/profile.d/vim.sh
if [ -n "$BASH_VERSION" -o -n "$KSH_VERSION" -o -n "$ZSH_VERSION" ]; then
    [ -x /usr/bin/id ] || return
    ID=`/usr/bin/id -u`
    [ -n "$ID" -a "$ID" -le 200 ] && return
    # for bash and zsh, only if no alias is already set
    alias vi >/dev/null 2>&1 || alias vi=vim
fi
```

Comprueba si el terminal cumple con los requisitos necesarios para lanzar vim, si cumplen, ejecuta, sino. No hace nada.

El fichero lang.sh, tiene para escoger el lenguaje de sistema por defecto, puede ser cambiado entre:

Japones, Koreano, Chino, Ingles, entre otros

```
if [ -n "$LANG" ]; then
  case $LANG in
    *.utf8*|*.UTF-8*)
      if [ "$TERM" = "linux" ]; then
        if [ "$consoletype" = "vt" ]; then
          case $LANG in
            ja*) LANG=en_US.UTF-8 ;;
            ko*) LANG=en_US.UTF-8 ;;
            si*) LANG=en_US.UTF-8 ;;
            zh*) LANG=en_US.UTF-8 ;;
            ar*) LANG=en_US.UTF-8 ;;
            fa*) LANG=en_US.UTF-8 ;;
            he*) LANG=en_US.UTF-8 ;;
            en_IN*) ;;
            *_IN*) LANG=en_US.UTF-8 ;;
          esac
        fi
      fi
    fi
  ;;
```

4. En algunas circunstancias la gestión basada sólo en usuario y grupo no es suficiente (por ejemplo queremos dar permiso de lectura a dos grupos a un mismo fichero). Se pueden fijar esos atributos con setfacl y getfacl. Estudiar dichos comandos y exponer ejemplos de ellos.

getfacl : Se utiliza para determinar los permisos establecidos en las listas de control de acceso de un fichero o directorio dado.

```
[fapCentOSuser@localhost ~]$ getfacl fapMeow/
# file: fapMeow/
# owner: fapCentOSuser
# group: fapCentOSuser
# flags: -s-
user::rwx
group::rwx
other::---

[fapCentOSuser@localhost ~]$
```

setfacl : Se utiliza para modificar los permisos en la listas de control de acceso de un fichero o directorio dado.

```
[fapCentOSuser@localhost ~]$ setfacl -m u:fapCentOSuser:rx fapMeow/
[fapCentOSuser@localhost ~]$ getfacl fapMeow/
# file: fapMeow/
# owner: fapCentOSuser
# group: fapCentOSuser
# flags: -s-
user::rwx
user:fapCentOSuser:r-x
group::rwx
mask::rwx
other::---
```

Referencias:

<http://shakaran.net/blog/2010/09/como-anadir-entorno-grafico-a-un-servidor-centos/>
<https://es.wikipedia.org/wiki/Ls>
<https://es.wikipedia.org/wiki/Umask>
<http://unix.stackexchange.com/questions/19333/disable-a-users-login-without-disabling-the-account>
<http://www.computerhope.com/unix/groupadd.htm>
<http://www.computerhope.com/unix/usermod.htm>
<http://www.computerhope.com/unix/groupmod.htm>
<http://www.computerhope.com/unix/userdel.htm>
<http://www.computerhope.com/unix/groupdel.htm>
<http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap5sec42.html>
http://docs.linux-es.org/FAQ/Html/FAQ_Linux_V2.0.2-130.html