6. Administración básica del sistema Linux

Autor	Xerach Casanova
Clase	Sistemas Informáticos
Fecha	@Mar 29, 2021 8:55 PM

1. Administración de usuarios y grupos

A lo largo de esta unidad, en los ejemplos, el prompt del sistema o shell se verá en negro, el comando ejecutado en negrita, las líneas devueltas por la máquina en color verde y los comentarios en rojo.

1.1. Creación de usuarios y grupos

Creación de nuevos usuarios (en un grupo nuevo)

adduser nombre_usuario

Este comando crea un usuario y también un grupo. Al ejecutarlo se realizan 3 tareas:

- Se crea un usuario.

- Se crea un grupo del mismo nombre.
- Se introduce ese usuario en ese grupo.

Al ejecutarlo se solicita un password que hay que introducir 2 veces.

```
miguel@SistemasUbuntu:~$ sudo adduser juan
[sudo] password for miguel:
Añadiendo el usuario `juan' ...
Añadiendo el nuevo grupo `juan' (1004) ...
Añadiendo el nuevo usuario `juan' (1004) con grupo `juan' ...
Creando el directorio personal `/home/juan' ...
Copiando los ficheros desde `/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX: Se introduce la contraseña
passwd: contraseña actualizada correctamente deseada 2 veces
Cambiando la información de usuario para juan
Introduzca el nuevo valor, o presione INTRO para el predeterminado
Nombre completo []: Juan Lopez Aquí se puede rellenar
Número de habitación []: o dejar todo en blanco
Teléfono del trabajo []: 987654321
Teléfono de casa []:
Otro []:
¿Es correcta la información? [S/n] s
```

En Windows hay un grupo Usuarios y los usuarios se crean por defecto en ese grupo.

Comandos para crear grupos, introducir a un usuario en ese grupo crear password

Crear grupos

addgroup nombre_grupo

Crear nuevo usuario en un grupo ya existente

adduser nombre_usuario —ingroup nombre_grupo

Cambiar contraseña a un usuario

passwd nombre_usuario

Hay que introducir 2 veces el password nuevo. El root puede cambiar la contraseña de todos los usuarios.

Ficheros muy importantes de usuarios y grupos

Cuando creamos usuarios y grupos escribimos en estos ficheros:

- /etc/passwd una línea para cada usuario
- /etc/shadow contiene las contraseñas de los usuarios, encriptadas

- /etc/group una linea para cada grupo

Crear en Ubuntu 3 nuevos usuarios:Usuario **juan** en grupo **juan**.Usuarios **pablo** y **juana** en un nuevo grupo llamado **electricista**.¿Cuáles son los comandos a ejecutar?

miguel@SistemasUbuntu:~$ **sudo adduser juan**

miguel@SistemasUbuntu:~$ **sudo addgroup electricista**

miguel@SistemasUbuntu:~$ **sudo adduser pablo --ingroup electricista**

miguel@SistemasUbuntu:~$ **sudo adduser juana --ingroup electricista**

Observación: Se podría ejecutar el primer comando **sudo su** de tal forma que se cambia la identidad a root y en el resto de comandos no habría que poner sudo. En ese caso, recordar que para volver al usuario alumno, se escribe **exit**

Significado de /etc/passwd

Este fichero tiene tantas líneas como usuarios. Aparecen bastantes más usuarios aparte de los creados por nosotros al haber muchos servicios con usuario asignado.

```
miguel@SistemasUbuntu:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash El usuario root siempre aparece en primera línea
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin usuario del servicio mail
.....
miguel:x:1000:1000:miguel,,,:/home/miguel:/bin/bash
alumno:x:1001:1001:,,,:/home/juan:/bin/bash
juan:x:1002:1002:Juan Lopez,,987654321,:/home/juan:/bin/bash
pablo:x:1003:1003:,,,:/home/pablo:/bin/bash
juana:x:1004:1003:JuanaGarcia,,765432198,:/home/juana:/bin/bash
```

Hay 7 campos separados por el carácter dos puntos ":".

juana representa el nombre de usuario.

x la x sirve para indicar que el usuario está habilitado y la contraseña encriptada.

1004 es el identificador de usuario UID, único para cada usuario. En Ubuntu empieza a partir de 1000 para usuarios normales, el root siempre es 0.

1003 representa el identificador de grupo GID. Es el grupo principal del usuario.

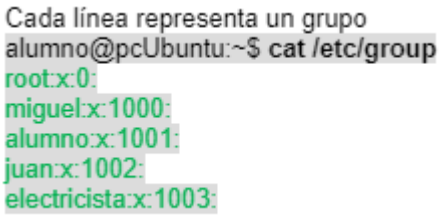
JuanaGarcia,,765432198 son los datos que se hayan introducido en la descripción del usuario.

/home/juana representa el directorio $HOME del usuario.

/bin/bash informa de la shell por defecto que utiliza el usuario, en los Linux actuales suele ser bash (bourne again shell) que derivó de sh (bourne shell).

En los servicios de los usuarios especiales la shell pone nologin y significa que no se podrá iniciar sesión en terminal o en gráfica con ese usuario.

Significado de las líneas de /etc/group



Hay 4 campos separados por el carácter dos puntos ":".

electricista representa el nombre del grupo.

x no tiene significado especial.

1003 es el GID del grupo.

En el 4º campo se especifican los usuarios que pertenecen a ese grupo de forma secundaria, separado por comas. Se utiliza solo cuando un usuario pertenece a varios grupos.

1.2. Eliminación y modificación de usuarios y grupos. propietarios de archivos

Eliminación de usuarios

userdel [-r] nombre_usuario

La opción -r elimina también el directorio $HOME correspondiente. Si no se pone, se elimina solo el usuario.

Eliminación de grupos: groupdel

groupdel nombre_grupo

Solo permite borrar un grupo si ningún usuario pertenece ya al grupo.

Realmente, userdel y groupdel borran las líneas correspondientes en /etc/passwd y /etc/group.

Cambiar a un usuario de grupo o su $HOME

Se puede realizar un cambio de grupo o la ruta de su &HOME, de manera manualmente en el archivo /etc/passwd, pero quedarían cosas pendientes. Por ejemplo, si se cambia la ruta en el fichero luego habrá que mover el directorio y cambiar permisos. Para realizar todo ese trabajo, se utiuliza el comando usermod.

usermod [-dgm] nombre_usuario

La opción -d cambia el directorio home del usuario.

La opción -g cambia al usuario de grupo principal.

la opción m mueve los archivos del directorio home antiguo al nuevo (solo funciona si se usa a la vez con -d).

Cambiar a un usuario en otros grupos suplementarios o secundarios

Todos los usuarios pueden pertenecer a varios grupos secundarios. Para añadirlos se ejecuta **adduser usuario grupo** teniendo en cuenta que el usuario y el grupo deben existir previamente.

Queremos que el usuario miguel, aparte de su grupo principal miguel, pertenezca también al grupo electricista

miguel@SistemasUbuntu:~$ **sudo adduser miguel electricista**

miguel@SistemasUbuntu:~$ **cat /etc/group**

.....electricista:x:1002:miguel #Ahora aparece en el cuarto campo que miguel tiene como grupo secundario el grupo electricista. En cambio del comando, también se podría haber añadido directamente en el archivo.

Comando grupos para ver grupos a los que pertenece un usuario

El comando groups muestra los grupos a los que pertenece el usuario.

sudo es un grupo de linux, en el que por defecto está el usuario que instala Linux, para que otro usuario pueda ejecutar sudo añadimos al usuario al grupo:

sudo adduser juan sudo

Cambiar el usuario y grupo propietario de un archivo o directorio

Todo archivo tiene un usuario y grupo propietario.

```
-rw-r--r-- 1 juana electricista 5 2012-05-22 00:00 archivo.txt
```

El usuario propietario de archivo.txt es juana (es quien creó el fichero) y el grupo propietario es electricista (el grupo principal del usuario).

Cambiar usuario propietario:

chown [-R] nuevoUsuarioPropietario fichero/directorio

La opción -R de recursivo, sirve para cambiar el propietario de un directorio con su árbol.

Cambiar grupo propietario

chgrp [-R] nuevoGrupoPropietario fichero/directorio

La opción -R sirve para lo mismo que en chown.

Cambiar en un solo comando usuario y grupo propietario

chown [-R] nuevoUsuarioPropietario:nuevoGrupoPropietario fichero/directorio

Cambiar a "archivo.txt" situado en /home/miguel al $HOME de pablo y que el propietario del archivo sea pablo y su grupo propietario electricista

```
root@SistemasUbuntu:/home/miguel# ls –l archivo.txt

-rw-r--r-- 1 miguel miguel 0 2012-05-22 00:00 archivo.txt

root@SistemasUbuntu:/home/miguel# mv archivo.txt /home/pablo/archivo.txt      #Movemos archivo.txt del $HOME de Miguel al $HOME de pablo

root@SistemasUbuntu:/home/miguel# cd /home/pablo

root@SistemasUbuntu:/home/pablo# chown pablo:electricista /home/miguel/archivo.txt      #Cambiamos con chown tanto el usuario propietario como el grupo propietario

root@SistemasUbuntu:/home/pablo# ls –l archivo.txt

-rw-r--r-- 1 pablo electricista 0 2012-05-22 00:00 archivo.txt
```

Diferencias de comandos de crear usuarios y grupos entre distintos Linux

En la mayoría de distribuciones funciona adduser o useradd. En Ubuntu adduser realiza más tareas que useradd y en Red Hat ocurre al contrario.

2. Montaje de dispositivos de almacenamiento

Para utilizar cualquier partición, pendrive o CD lo tenemos que montar.

directorio /dev

Para montar una partición tenemos que conocer donde está su archivo de dispositivo. Equivale a decir que en /dev se encuentran los drivers. Es necesario saber la nomenclatura de los medios de almacenamiento.

Nomenclatura de dispositivos

- Primer disco: /dev/sda
- Segundo disco: /dev/sdb (y sucesivamente por orden alfabético).
- Particiones: Números 1 a 4 primarias, a partir de la 5, lógicas.

Por ejemplo: /dev/sdb7 representa la tercera partición lógica del segundo disco.

- Disquetera: /dev/fd0 (en otras versiones de unix-linux es /dev/floppy.
- CD y DVD: /dev/sr0 (en otras versiones de unix-linux es /dev-floppy).

¿Qué discos duros y particiones tenemos en nuestro PC?

Como los discos duros son sda, sdb,... y las particiones van con números, se trata de ver los archivos en /dev que empiezan por sd.

```
miguel@SistemasUbuntu:~$ ls -l /dev/sd*
brw-rw---- 1 root disk 8, 0 nov 27 09:35 /dev/sda
brw-rw---- 1 root disk 8, 1 nov 27 09:35 /dev/sda1
brw-rw---- 1 root disk 8, 2 nov 27 09:35 /dev/sda2
brw-rw---- 1 root disk 8, 16 dic 17 23:06 /dev/sdb
brw-rw---- 1 root disk 8, 17 dic 17 23:06 /dev/sdb1
```

En la captura se ven 2 discos: **sda** representa el disco duro de la máquina SistemasUbuntu con sus 2 particiones, **sda1** la partición / y **sda2** la partición **swap**. El segundo disco, **sdb** es un pendrive en la máquina con una partición **sdb1**.

Punto de montaje: /media o /mnt

En Windows las particiones se montan en las letras D, E... En Línux solo hay un árbol de directorios (/), todos los dispositivos de almacenamiento como

pendrive, discos externos, cd... se montan en /media

Hay veces que se ha de hacer de forma manual, en este caso lo montamos manualmente en el directorio /mnt

Denominación de los distintos sistemas de archivos en Linux

Los posibles sistemas de ficheros en una unidad lógica son:

ext2, ext3, ext4 los de linux

msdos si es FAT16

vfat si es FAT32

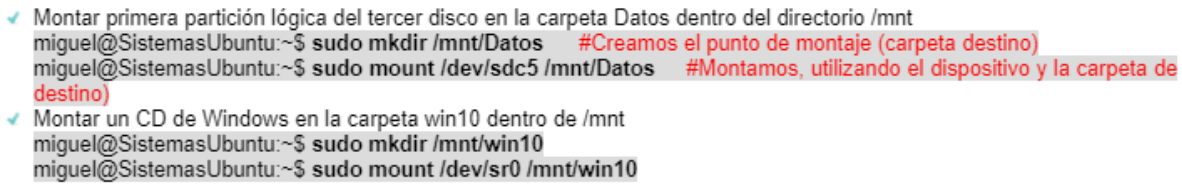
ntfs

iso9660 (en CDROM)

Comando de montaje mount

mount dispositivo punto_montaje

Antes de montar con el comando debe existir el punto de montaje o directorio de destino.



A veces mount no puede determinar automáticamente el sistema de ficheros del dispositivo, para ello se utiliza la opción -t:

sudo mount -t iso9660 /dev/sr0/mnt/win10

Comando df

Muestra los dispositivos montados, con el espacio total, ocupado y libre. Con la opción -h da la información con unidades.

```
miguel@SistemasUbuntu:~$ df -h
S.ficheros Tamaño Usados Disp Uso% Montado en
/dev/sda1 46G 5,3G 38G 13% /
Compartir 440G 379G 62G 87% /media/sf_Compartir
/dev/sdb1 59G 56G 2,7G 96% /media/miguel/LabelPendrive
/dev/sr0 4,4G 4,4G 0 100% /mnt/win10
```

La primera línea es la primera partición del primer disco (/dev/sda1), tiene 46GB y 5,3GB están ocupados.

La segunda línea es la carpeta compartida de máquina anfitrión.

La tercera línea es la primera partición del segundo disco (/dev/sdb1), corresponde a un pendrive de 59GB montado automáticamente en /media/miguel/LabelPendrive

La cuarta línea es un CD-Rom (/dev/sr0) que está montado en /mnt/win10 y el dvd tiene 4,4GB

Desmontar un dispositivo: unmount

Es obligatorio desmontar un dispositivo antes de extraerlo para evitar incoherencias. Muchas escrituras no se realizan en el momento, sino cuando el procesador dispone de tiempo. El comando unmount admite dos sintaxis.

unmount dispositivo

unmount punto_de_montaje

Windows del ejemplo anterior se puede hacer:

unmount /dev/sr0

unmount /mnt/win10

Fichero /etc/fstab

El comando mount monta el dispositivo, pero cada vez que se inicia el equipo habrá que ejecutarlo. En este fichero podemos añadir una lista de dispositivos que queremos que se monten al iniciar el equipo.

Añadir las líneas correspondientes en /etc/fstab para montar siempre:

- En /mnt/Datos1 la primera partición lógica del segundo disco, formateada fat32
- En /mnt/Datos2 la segunda partición lógica del segundo disco, formateada ntfs

```
miguel@SistemasUbuntu:~$ sudo nano /etc/fstab
<file system> <mount point> <type> <options> <dump> <pass>
..... Añadimos al final del fichero las 2 líneas siguientes:
/dev/sdb5 /mnt/Datos vfat user,rw 0 0
/dev/sdb6 /mnt/Datos2 ntfs user,rw 0 0
```

Opciones:

user/nouser: permite utilizar la partición a todos los usuarios o solo al root.

rw/ro: montar lectura- escritura o solo lectura.

defaults: aplica las opciones rw, suid, dev, exec, auto, nouser, async.

dump es una utilidad de copia de seguridad, normalmente ponemos 0.

pass tiene que ver con la comprobación de errores, normalmente ponemos 0.

man fstab muestra el manual de /etc/fstab.

3. Administración de discos y particiones

Crear particiones: programa fdisk

Las herramientas internas como fdisk no permiten estas opciones como las herramientas de partición externas (gparted) pero hay algunos motivos para trabajar con ellos:

- fdisk es la herramienta nativa de Unix/Linux y se encuentra en todas las distribuciones.
- Solo se necesita interfaz de texto y es buena opción cuando se accede en remoto. Muchos equipos y servidores se encuentran en la nube en máquinas virtuales y suelen trabajar sin entorno gráfico.

comandos fdisk:

fdisk -l Muestra en pantalla información de todas las particiones de todos los discos.

fdisk -l /dev/sda información de todas las particiones del primer disco.

fdisk /dev/sda Si no se utiliza la opción-l se abre fdisk para administrar particiones. Las opciones son:

- **m** muestra las posibles opciones
- **p** muestra en pantalla las particiones actuales (print)
- **n** añade una nueva partición (new)
- **d** borra una partición (delete
- **w** guarda cambios realizados y sale de fdisk (write)

Interpretación de la información de fdisk

```
miguel@virtual:~$ sudo fdisk -l
[sudo] password for miguel:
Disco /dev/sda: 42.9 GB, 42949672960 bytes #Primer disco de 43GB es una máquina virtual)
255 cabezas, 63 sectores/pista, 5221 cilindros, 83886080 sectores en total #83 millones de sectores
Unidades = sectores de 1 * 512 = 512 bytes #tamaño del sector, 512 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador del disco: 0x00096458        #Fijarse que en cada partición pone sector de inicio y sector final.
Dispositivo Inicio Comienzo Fin Bloques Id Sistema #También aparece sistema de archivos
/dev/sda1 * 2048 29296639 14647296 83 Linux #Primera partición primaria con sistema archivos Linux (la partición raíz) El *
en la primera partición significa que es la activa.
/dev/sda2 29296640 31297535 1000448 82 Linux swap / Solaris #Segunda partición primaria (la swap o área de intercambio)
/dev/sda3 43585536 83886079 20150272 5 Extendida #Tercera partición primaria que es la extendida
/dev/sda5 43587584 64067583 10240000 b W95 FAT32 #Dentro de extendida, primera lógica tipo fat32
/dev/sda6 64069632 83886079 9908224 7 HPFS/NTFS/exFAT #Segunda lógica, tipo ntfs
Disco /dev/sdb: 62.7 GB, 62742792192 bytes #Segundo disco, es un pendrive de 64GB
13 cabezas, 4 sectores/pista, 2356625 cilindros, 122544516 sectores en total
Unidades = sectores de 1 * 512 = 512 bytes
Tamaño de sector (lógico / físico): 512 bytes / 512 bytes
Tamaño E/S (mínimo/óptimo): 512 bytes / 512 bytes
Identificador del disco: 0x000e76a6
Dispositivo Inicio Comienzo Fin Bloques Id Sistema
/dev/sdb1 2048 122544515 61271234 7 HPFS/NTFS/exFAT #Una partición NTFS ocupando todo el pendrive
```

Los primeros 2048 sectores, (equivalen a 2048*512bytes = 1MB) están reservados para MBR, GPT o futuras estructuras.

Nos preguntamos, ¿Queda espacio libre en disco /dev/sda?

Si nos fijamos, en sectores de inicio y final, Vemos que sda2 termina en sector 31 millones y sda3 comienza en sector 43 millones (en número redondos). La extendida sda3, va del 43 millones al 83millones que coincide con el total de sectores del disco. Además, las lógica sda5 y sda6 ocupan totalmente la extendida.

Por tanto, el espacio libre es del sector 31 millones al 43 millones, es decir unos 6GB.

Formatear particiones con mkfs

Cuando se crea una partición, hay que formatearlas antes de utilizarlas, una vez creada una partición con fdisk se debe reini8ciar el equipo y después formatearlas.

mkfs -t ext4 /dev/sda6 Formatea la partición sda6 con formato ext4

mkfs /dev/sda6 formatea la partición sda6 con formato ext3, formato por defecto si no se utiliza -t

4. Permisos de ficheros y directorios

Cuando listamos las propiedades de un archivo con ls -l, aparecen en el primer campo 10 caracteres. El primer carácter indica si es directorio o fichero y los 9 restantes son los permisos.

Un fichero con todos los permisos tendría: rwxrwxrwx, donde r es permiso de lectura, w es permiso de escritura y x es permiso de ejecución (si aparece un guión significa que no tiene permiso).

Los tres grupos de letras corresponden a:

- **Primer grupo:** user, permisos del usuario propietario.
- **Segundo grupo:** group, permisos del grupo propietario del fichero, excluido el propietario que puede incluso no pertenecer al grupo.
- **Tercer grupo:** otros, permisos del resto de usuarios.

Por ejemplo, en el siguiente archivo:

-r w - r - - - - - 1 pablo electricista 40 2018-02-01 22:27 archivo.txt

- El usuario propietario es pablo y sus permisos son rw-.
- El grupo propietario es electricista y sus usuarios (salvo pablo) .solo pueden leer el archivo (r- -).
- El resto de usuarios no tiene ningún permiso.
- Nadie puede ejecutar el fichero.

En Windows un fichero es ejecutable si tiene extensión ejecutable, en Linux es si tiene la x en los permisos. Además, si en el ejemplo anterior el grupo electricista tuviese mayores permisos que pablo, pablo no tendría esos permisos aunque pertenezca al grupo. Los permisos en linux no son acumulativos.

Permisos en directorios

Los permisos de lectura, escritura y ejecución en directorios significan lo siguiente:

- **r**: Permiso para listar el contenido del directorio (sin ese permiso no se puede ejecutar ls).
- **w**: Permiso para crear o borrar entradas en el directorio (protege borrado de ficheros en la estructura del directorio)
- **x**: Permiso para acceder a las entradas. No se puede ejecutar cd sin ese permiso. r y x van relacionadas, se permiten ambas o ninguna.

Cambiar permisos: comando chmod

Los permisos solo los pueden cambiar el usuario propietario del fichero y el usuario root.

chmod permisos nombre_fichero

Hay dos notaciones para cambiar los permisos.

1ª forma (Notación octal)

Se convierte cada grupo de permisos en un número octal. Cada letra se sustituye por 1 o 0, después sumamos a los que son 1 el valor siguiente: lectura 4, escritura 2, ejecución 1.

(r w -) (r w x) (- - x)

(1 1 0) (1 1 1) (0 0 1)

(4+2)(4+2+1)(1) = 671

chmod 671 archivo.txt

2ª forma (notación simbólica)

Se utiliza un patrón de texto formado por

1. Las categorías afectadas:

   u: para el propietario.

   g: para el grupo.

   o: para el resto de usuarios.
2. Las abreviaturas de los tipos de permisos:

   r: lectura.

   w: escritura.

   x ejecución.

chmod o+w /home/usuario1/prueba dar permisos de escritura al resto de usuarios sobre el fichero prueba.

chmod go-w /home/usuariol/prueba quitar los permisos de escritura del fichero a todos excepto al propietario.

5. Gestión de procesos

Proceso de arranque

Cuando se inicia el equipo primero se inicia la BIOS, seguidamente el gestor de arranque que en Linux se llama GRUB y en el caso de iniciar un sistema GNU/Linux accede al directorio /boot, donde se carga el kernel y el proceso init con PID 1, el cual es encargado de iniciar todos los servicios para que el usuario funcione correctamente.

Tradicionalmente el arranque estaba basado en Unix System V, pero la mayoría han adoptado el sistem systemd, el cual inicia el proceso kthreadd con PID 2, el cual gestiona el resto de servicios. Todos los procesos son hijos, nietos del init o del kthreadd

Comando de procesos

Comando ps [-efl]

ps lista los procesos vivos (activos, en espera o bloqueados), los procesos ya terminados no aparecen. Las opciones habituales son -ef o -efl

```
UID PID PPID C STIME TTY TIME CMD
root 1 0 1 23:41 ? 00:00:02 /sbin/init splash
root 2 0 0 23:41 ? 00:00:00 [kthreadd]
root 3 2 0 23:41 ? 00:00:00 [kworker/0:0]
.....
miguel 1387 1 0 23:42 ? 00:00:00 /lib/systemd/systemd --user
.....
miguel 2054 1387 1 23:43 ? 00:00:00 /usr/bin/gnome-calendar --gapplication-service
miguel 2055 1387 1 23:43 ? 00:00:00 /usr/lib/gnome-terminal/gnome-terminal-server
miguel 2142 2055 0 23:43 pts/0 00:00:00 bash
miguel 2153 1418 0 23:43 tty2 00:00:00 update-notifier
miguel 2162 2153 13 23:43 tty2 00:00:01 /usr/bin/python3 /usr/lib/update-notifier/apt-check
miguel 2177 2142 0 23:43 pts/0 00:00:00 ps -ef      #El último proceso es el PID 2177, es la propia ejecución del comando ps –ef. Su padre (PPID) es el PID 2142, que es la propia terminal bash y su padre es el 2055. El padre del 2055 es el 1387, que a la vez es hijo del PID 1 que es el proceso init.
```

UID: Usuario que ejecutó el proceso.
PID: Identificador de proceso, son correlativos.
PPID: Identificador de proceso padre.
C: Porcentaje de utilización de CPU para el proceso.

STIME: Hora al que se inició el proceso.
TTY: Terminal que lo ha ejecutado.
TIME: Tiempo utilizado del procesador para la ejecucióm.
CMD: Nombre del comando ejecutado.

Terminar procesos con kill

kill -9 PID

El número -9 es una señal para matar procesos, la señal por defecto es la -15, pero -9 es más potente, se pueden ver todas las señales con el comando **man kill**

Los procesos los pueden terminar el usuario que ejecuta y el administrador.

Comando yes

manda el carácter infinitamente hasta que lo finalicemos. **yes hola** devuelve hola infinitamente.

Comando top

Muestra los procesos ordenados por consumo de recursos.

Ejemplo. Ejecutar yes en una terminal y desde otra terminal descubrir PID, consumo de recursos y matar yes

Paso 1.
En primera terminal, ejecutar yes:
miguel@SistemasUbuntu:~$ **yes**
y
y #Muestra y infinitas, hasta que paremos el proceso. Consume mucho procesador

Paso 2.
Sin cerrar la primera terminal, abrir una segunda terminal, ejecutar ps –ef o top (en ambos comandos, se puede ver PID y consumo de procesador)
miguel@SistemasUbuntu:~$ **top**
top - 00:46:15 up 13 min, 2 users, load average: 3,84, 2,90, 1,52
Tareas: 209 total, 5 ejecutar, 174 hibernar, 0 detener, 0 zombie
%Cpu(s): 29,1 usuario, 16,2 sist, 0,2 adecuado, 53,9 inact, 0,3 en espera, 0,0 hardw int, 0,2 sof
KiB Mem : 2041304 total, 241500 libre, 1124408 usado, 675396 búfer/caché
KiB Intercambio: 3999740 total, 3999740 libre, 0 usado. 744020 dispon Mem
PID USUARIO PR NI VIRT RES SHR S %CPU %MEM HORA+ ORDEN
1076 miguel 20 0 447168 100404 47388 R 20,0 4,9 0:31.35 Xorg
1634 miguel 20 0 802324 39420 28316 R 20,0 1,9 2:14.02 gnome-terminal-
31 root 20 0 0 0 0 R 15,0 0,0 0:29.58 kworker/u2:1
1964 miguel 20 0 14576 732 668 S 15,0 0,0 1:14.99 yes
2214 miguel 20 0 49020 3860 3236 R 15,0 0,2 0:00.03 top #En este ejemplo, el comando yes tiene el PID 2214 y está consumiendo el 15% de CPU (la velocidad a las que manda y es muy grande)

Paso 3. Terminar el proceso yes definitivamente:
miguel@SistemasUbuntu:~ $ **kill -9 2214**

Comando nice: ejecutar proceso con prioridad concreta.

La prioridad de los procesos en Linux a desde el -20 que es la máxima, hasta 19 que es la mínima, siendo 0 la prioridad por defecto.

Para ejecutar un proceso con prioridad se utiliza **nice [-n prioridad] comando**

Todos los usuarios pueden utilizar nice, pero solo el root puede usar números negativos.

nice yes (ejecuta yes con prioridad 0)
nice -n -10 yes (ejecuta yes con prioridad -10)

Comando renice: cambiar prioridad a un proceso ya en ejecución.

renice [prioridad] -p PID

renice 10 -p 3183 cambia a prioridad 10 el proceso con PID 3183, Si no se designa prioridad la cambia a 0. Los usuarios solo pueden bajar prioridad, mientras que el root puede subirla y bajarla.

6. Información del sistema y registro

Información del sistema y kernel. Comando uname [-ra]

uname devuelve información del sistema, con la opción -r se devuelve la versión de kernel instalada y con la opción -a devuelve información del Linux instalado, con su kernel, nombre de equipo y si es 32 o 64 bits.

```
miguel@SistemasUbuntu:~$ uname -r
4.15.0-43-generic kernel 4.15
miguel@SistemasUbuntu:~$ uname -a
Linux SistemasUbuntu 4.15.0-43-generic #46-Ubuntu SMP Thu Dec 6 14:45:28 UTC 2018 x86_64 x86_64 x86_64
GNU/Linux     #Linux instalado en máquina con nombre SistemasUbuntu. Es Ubuntu del 2018 con kernel 4.15 de 64
bits, instalado en procesador de 64 bits
```

El kernel o núcleo de GNU-Linux es independiente de la distribución. Cuando se adquiere cualquier tipo de hardware, si es compatible con Linux te informan del kernel mínimo requerido.

Información de la memoria principal y swap: comando free

Muestra información sobre la memoria principal (RAM) y memoria de intercambio total, utilizada y libre.

```
miguel@SistemasUbuntu:~$ free -h
total usado libre compartido búfer/caché disponible
Memoria: 1,9G 1,1G 169M 19M 665M 666M
Swap: 3,8G 0B 3,8G
El equipo tiene 1,9GB, con 1,1GB ocupados y 169MB libres.
La memoria swap tiene 3.8GB, toda libre.
```

Características del procesador: comando lscpu

Devuelve información del procesador: núcleos, velocidad...

```
miguel@SistemasUbuntu:~$ lscpu
Arquitectura: x86_64 #procesador de 64 bits
modo(s) de operación de las CPUs: 32-bit, 64-bit
Orden de los bytes: Little Endian
CPU(s): 8 #¿8 CPU? Son 8 hilos de ejecución en total, que se llama también 8 CPU lógicas o 8 CPU virtuales.
Lista de la(s) CPU(s) en línea: 0
Hilo(s) de procesamiento por núcleo: 2 #2 hilos de ejecución por núcleo. Hay muchos procesadores que solo tienen
1 hilo de ejecución por núcleo.
Núcleo(s) por «socket»: 4 #4 núcleos.
«Socket(s)» 1
Modo(s) NUMA: 1
ID de fabricante: GenuineIntel
Familia de CPU: 6
Modelo: 60
Nombre del modelo: Intel(R) Core(TM) i7-4702MQ CPU @ 2.20GHz#Procesador Intel i7 de cuarta generación (4702)
Revisión: 3
CPU MHz: 2194.914 #Velocidad procesador 2194MHz
BogoMIPS: 4389.82
Fabricante del hipervisor: KVM
Tipo de virtualización: VT-x #Procesador con instrucciones específicas para virtualización
Caché L1d: 32K
Caché L1i: 32K #L1: 32KB de datos + 32KB de instrucciones
Caché L2: 256K #L2: 256KB
Caché L3: 6144K #L3: 6144KB
CPU(s) del nodo NUMA 0: 0-7
.....
```

Particiones montadas: Comando df [-h]

df devuelve el espacio total, libre y ulizidado de los dispositivos y particiones montadas. La opción -h aparece en MB o GB, sin la opción -h aparece en KB.

```
miguel@SistemasUbuntu:~$ df -h
S.ficheros Tamaño Usados Disp Uso% Montado en
/dev/sda1 46G 5,3G 38G 13% / #sda1 tiene montada la partición / con 46GB de los que hay ocupados 5.3GB.
.....
Compartir 440G 380G 61G 87% /media/sf_Compartir #Montada la carpeta compartir con la máquina anfitrión en
/media/sf_Compartir
/dev/sr0 56M 56M 0 100% /media/miguel/VBox_GAs_5.2.16 #Está montado sr0 que es el CD de Guest adittions
/dev/sdb1 15G 14G 1,5G 91% /media/miguel/NUEVO VOL #Está montado un pendrive, como segundo disco sdb de
15GB y 1,5GB libres
```

Información de los directorios: Comando du [-sh]

Informa del espacio utilizado por el directorio especificado, incluyendo los que ocupan los subdirectorios.

-s devuelve solo la línea del directorio y no la de subdirectorios.

-h devuelve la información en MB o GB.

Directorio /proc

Es un directorio cargado en memoria RAM. Su contenido no se guarda en el disco.

cat /proc/cpuinfo información del procesador (parecida a lscpu).

cat /proc/meminfo información de memoria.

En él se guarda información de cada proceso, para cada PID se crea una carpeta /proc/PID.

Registro del sistema: syslog

Tanto Windows como Linux tiene archivos log de texto plano que registran lo que ocurre en el sistema. Cuando se inicia el PC se escribe en esos archivos.

En Linux el directorio es /var/log en la que hay una carpeta para cada servicio, donde se guardan los archivos .log sobre cada servicio.

Archivo de registro principal /var/log/syslog

Recoge todo lo que ocurre en sistema, se guardan los eventos desde que se instaló GNU-Linux. Normalmente se mira el contenido de las últimas líneas con el comando tail.

```
miguel@portatil:~$ tail -20 /var/log/syslog #se muestran últimas 20 líneas del archivo de registro
Feb 9 10:05:57 portatil colord: Profile added: DCP9020CDW-Gray.. #mensajes relacionados con impresoras
Feb 9 10:05:57 portatil colord: Profile added: DCP9020CDW-RGB.. #brother DCP y HP instaladas en el equipo
Feb 9 10:05:57 portatil colord: Device added: cups-DCP9020CDW #cups es el servicio de impresoras de GNU-Linux
Feb 9 10:05:57 portatil colord: Profile added: hp-LaserJet-1000-Gray..
Feb 9 10:05:57 portatil colord: Profile added: hp-LaserJet-1000-RGB..
Feb 9 10:05:57 portatil colord: Device added: cups-hp-LaserJet-1000
Feb 9 10:09:43 portatil kernel: [ 260.532054] capability: warning: `VirtualBox' uses 32-bit capabilities (legacy support in use) #Esta línea y
siguientes relacionadas con inicio VirtualBox
Feb 9 10:09:51 portatil kernel: [ 268.747435] vboxdrv: ffffffffc09e5020 VMMR0.r0
Feb 9 10:09:51 portatil kernel: [ 268.860732] vboxdrv: ffffffffc0adf020 VBoxDDR0.r0
Feb 9 10:09:51 portatil kernel: [ 268.863287] vboxdrv: ffffffffc0408020 VBoxDD2R0.r0
Feb 9 10:09:51 portatil kernel: [ 268.894066] vboxdrv: ffffffffc0135020 VBoxEhciR0.r0
#próximas líneas relacionadas con cron: trabajos programados. Ejecución automática diaria (cron, se ve en próximo libro)
Feb 9 10:10:27 portatil anacron[1107]: Job `cron.daily' started
Feb 9 10:10:27 portatil anacron[2765]: Updated timestamp for job `cron.daily' to 2018-02-09
Feb 9 10:11:04 portatil anacron[1107]: Job `cron.daily' terminated (exit status: 1) (mailing output)
Feb 9 10:11:04 portatil anacron[1107]: Can't find sendmail at /usr/sbin/sendmail, not mailing output
Feb 9 10:11:04 portatil anacron[1107]: Normal exit (1 job run)
Feb 9 10:17:01 portatil CRON[3150]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
Feb 9 10:20:02 portatil udisksd[2245]: Cleaning up mount point /media/miguel/MiguelAngelGarcia (device 8:17 no longer exist)
#desmontando pendrive
Feb 9 10:20:02 portatil ntfs-3g[2292]: Unmounting /dev/sdb1 (MiguelAngelGarcia)
```

7. Tareas programadas

Demonio cron y archivo /etc/crontab

En Linux se programan tareas con cron, como por ejemplo realizar una copia de seguridad todos los días a la misma hora.

Se compone de dos elementos: cron, el demonio (un ejecutable que está corriendo todo el tiempo) y el archivo de configuración **/etc/crontab**

```
miguel@SistemasUbuntu:~$ ps -ef | grep cron #devolvemos solo líneas que aparezca cron (en Windows, utilizábamos "| find" en vez de "| grep"
root 580 1 0 06:51 ? 00:00:00 /usr/sbin/cron –f #efectivamente se está ejecutando el demonio cron
miguel 4517 2179 0 09:05 pts/1 00:00:00 grep --color=auto cron #esta línea no tienen ningún valor, solo está relacionada con la ejecución del ps
```

Archivo de configuración de las tareas programadas

Para añadir tareas programadas se puede hacer directamente en el archivo /etc/crontab con cualquier editor, normalmente se utiliza nano.

La diferencia entre editar con nano/etc/crontab y utilizar el comando crontab -e es que a este último no le hacen falta privilegios de root. Cuando se ejecuta crontab -e se solicita el editor a utilizar.

Sintaxis de cada línea:

```
# m h dom mon dow command              #encabezamiento del archivo
10   8    *       *       3    /home/miguel/script.sh    #Se ejecuta script.sh todos los miércoles a las 8:10. Ver significado de cada campo:
```

m: minuto
h: hora (entre 0 y 23).
dom: día del mes
mon: mes
dow: día de la semana, válidos entre 0 (domingo) y 6 (sábado).

Comodines: - , * /

Comodín *: todos los valores
3-6 Valores 3, 4, 5, 6
3,6 Valores 3 y 6
*/10 Cada 10

Ejemplos:

1. Apagar el ordenador todos los días a las 21:50: 50 21 * * * poweroff
2. Ejecutar script.sh cada 20 minutos desde las 9 hasta las 10 de lunes a viernes. 2 opciones:

   00,20,40 9 * * 1-5 /home/miguel/script.sh
   /20 9 * * 1-5 /home/miguel/script.sh
3. Crear una copia de seguridad cada 2 días a las 23:50 del home y todos los usuarios.

   50 23 /2 * * tar –cvzf /root/home.tar.gz /home/*

Otras opciones de crontab:

- Ver tareas programadas de un usuario: **crontab -l -u nombre_usuario**
- Eliminar tareas de un usuario **crontab -r -u nombre_usuario**

Carpetas predefinidas para la ejecución periódica de tareas programadas en el directorio

Con versiones actuales de Linux, en /etc se encuentran los siguientes directorios:

cron.hourly
cron.daily
cron.weekly
cron.monthly

Los scripts que se pongan en cada subdirectorio se ejecutarán según el subdirectorio utilizado.

Los scripts en Linux son equivalentes a los archivos por lotes de Windows. En Linux lo importante es que tengan permiso de ejecución y para diferenciarlos, aunque no es obligatorio se le suele poner la extensión.sh

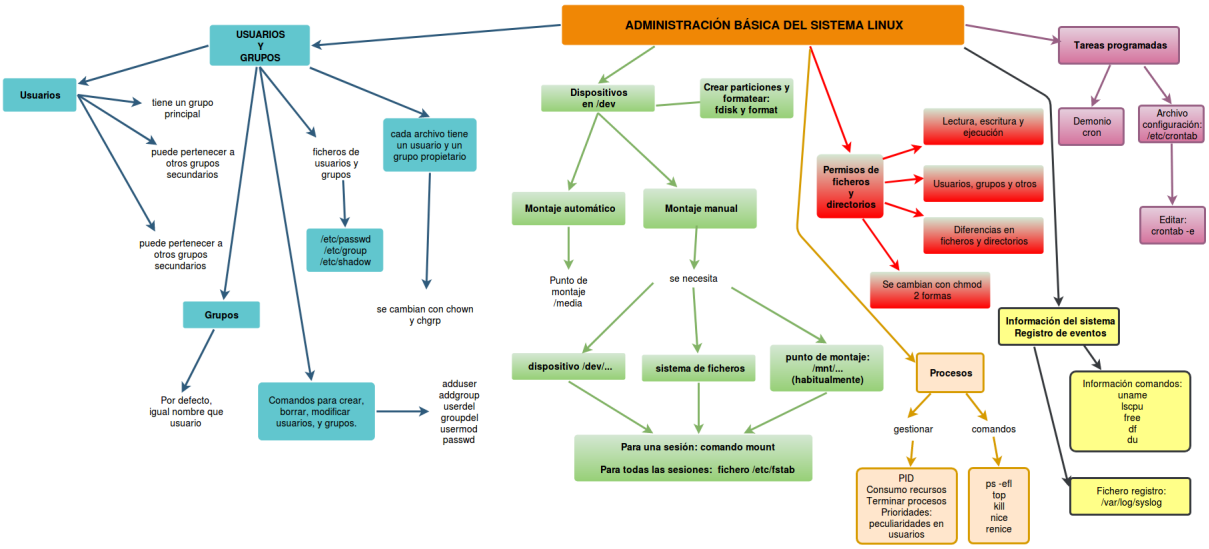
La primera línea de un script.sh es: **#!/bin/bash** y sirve para decir que el script lo va a interpretar la shell bash

Ejemplo de ejecución de un script:

Realizar un script que cree un directorio, direccione contenido a un fichero del directorio, limpie la pantalla y muestre el listado del directorio y el contenido del fichero.

```
miguel@SistemasUbuntu:~$ nano ejemplo.sh
#Con nano se escribe el contenido del script. Se muestra su contenido con cat.
miguel@SistemasUbuntu:~$ cat ejemplo.sh
#!/bin/bash
mkdir /home/miguel/carpeta
echo línea1 del fichero.txt > /home/miguel/carpeta/fichero.txt
clear
ls -l /home/miguel/carpeta
echo Se muestra contenido del fichero.txt:
cat /home/miguel/carpeta/fichero.txt
miguel@SistemasUbuntu:~$ ls -l ejemplo.sh
-rw-r--r-- 1 miguel miguel 218 ene 25 02:33 ejemplo.sh
#El archivo no es ejecutable, por lo que si se ejecuta, devuelve "Permiso denegado"
miguel@SistemasUbuntu:~$ ./ejemplo.sh
bash: ./ejemplo.sh: Permiso denegado
#Se pone permiso de ejecución al usuario propietario (miguel)
miguel@SistemasUbuntu:~$ chmod u+x ejemplo.sh
#Se ejecuta sin problemas. Limpia la pantalla.
miguel@SistemasUbuntu:~$ ./ejemplo.sh
#Limpia la pantalla, y devuelve listado de la carpeta y el contenido de fichero.txt
total 4
-rw-r--r-- 1 miguel miguel 24 ene 25 02:34 fichero.txt
Se muestra contenido del fichero.txt:
línea1 del fichero.txt
miguel@SistemasUbuntu:~$
```

Mapa conceptual


ADMINISTRACIÓN BÁSICA DEL SISTEMA LINUX
USUARIOS Y GRUPOS
Usuarios
tiene un grupo principal
puede pertenecer a otros grupos secundarios
puede pertenecer a otros grupos secundarios
Grupos
Por defecto, igual nombre que usuario
Comandos para crear, borrar, modificar usuarios, y grupos.
adduser
addgroup
userdel
groupdel
usermod
passwd
ficheros de usuarios y grupos
/etc/passwd
/etc/group
/etc/shadow
cada archivo tiene un usuario y un grupo propietario
se cambian con chown y chgrp
Dispositivos en /dev
Crear particiones y formatear: fdisk y format
Montaje automático
Montaje manual
Punto de montaje /media
se necesita
dispositivo /dev/...
sistema de ficheros
punto de montaje: /mnt/... (habitualmente)
Para una sesión: comando mount
Para todas las sesiones: fichero /etc/fstab
Permisos de ficheros y directorios
Lectura, escritura y ejecución
Usuarios, grupos y otros
Diferencias en ficheros y directorios
Se cambian con chmod 2 formas
Procesos
gestionar
comandos
PID
Consumo recursos
Terminar procesos
Prioridades:
peculiaridades en usuarios
ps -efl
top
kill
nice
renice
Tareas programadas
Demonio cron
Archivo configuración: /etc/crontab
Editar: crontab -e
Información del sistema Registro de eventos
Información comandos:
uname
lscpu
free
df
du
Fichero registro: /var/log/syslog