**TEMA 5: ADMINISTRACIÓN DE GNU/LINUX**

**1- Administración de usuarios y grupos locales.**

GNU/Linux es un sistema multiusuario donde varios usuarios pueden iniciar sesión simultáneamente

Las tareas que realiza el administrador de usuarios:

* Añadir, modificar y eliminar usuarios en el sistema.
* Añadir, modificar y eliminar grupos locales o globales.
* Fijar el plan de cuentas y contraseñas en el equipo junto con una política de derechos de usuario.
* Establecer el sistema de auditorías.

Cada usuario pertenece a un conjunto de usuarios denominado **grupo de usuario**

El comando para salir del sistema es logout, para terminales en modo texto, y exit para modo gráfico

Los usuarios se identifican por un número único llamado UID y un número de grupo llamado GID

**En el sistema existen tres tipos de usuarios que son:**

**Usuario root:** tiene como UID el valor cero y se encarga de todas las tareas de administración del sistema

**Usuarios del sistema**, como por ejemplo: daemon,nobody,backup,etc

**Usuarios normales:** se usan para usuarios individuales,ubicados en /home

**Grupos:**

El sistema dispone de una serie de grupos predeterminados

Cuando se da de alta un usuario se crea un grupo con el mismo nombre que el usuario y con los mismos derechos y privilegios que se conceden al usuario, por lo tanto cada vez que se instala un servicio en el ordenador se crea un grupo perteneciente al servicio

La información de los grupos se guarda en /etc/group

**Recuerda que un grupo no puede contener a otro grupo.**

–La información de las cuentas de usuario en Linux se almacena en dos archivos:

**/etc/passwd: contiene las cuentas de usuario existentes**

<nombre usuario>:<contraseña encriptada>:<uid>:<gid>:<descripción de la cuenta>:<el directorio local [home]>:<shell>

**/etc/shadow: contiene las contraseñas encriptadas de los usuarios e información relacionada con la gestión de las contraseñas**

<nombre usuario><contraseña encriptada>**<1><2><3><4><5><6><7>**

**Donde:**

**1-**Días transcurridos desde 1-1-1970 donde el password fue cambiado por última vez.  
**2-**El mínimo número de días entre cambios de contraseña.  
**3-**Días máximos de validez de la cuenta.  
**4-**Días que avisa antes de caducar la contraseña.  
**5-** Días después de que un password caduque para deshabiltar la cuenta  
**6-** Fecha de caducidad. días desde 1-1-1970, donde la cuenta es deshabilatada y el usuario no podrá iniciar sesión  
**7-** Campo reservado para usos futuros.

Para confirmar cambios usamos change (change -l <usuario>)

–La información de los grupos de usuario en Linux se almacena en los archivos:

**/etc/group: cada línea representa a un grupo con la siguiente información:**

<nombre grupo>: <contraseña encriptada o "x" si no tiene contraseña>: <gid o identif. del grupo>:<lista de los miembros del grupo>

**/etc/gshadow: contraseñas encriptadas por grupos**

**Otros:**

El directorio **/etc/skel** proporciona una forma de estar seguro de que todos los nuevos usuarios de tu sistema tienen la misma configuración inicial

| **Comandos para gestionar usuarios:** |
| --- |
| **adduser** <nombreusuario> → crea un usuario.  **useradd** <nombreusuario> → crea un usuario. No se recomienda su uso en Debian.  **deluser** <nombreusuario> → elimina un usuario.  **userdel** <nombreusuario> → elimina un usuario. No se recomienda su uso en Debian.  **usermod** <nombreusuario> → modifica el usuario.  **passwd** <nombreusuario> → crea o cambia la contraseña a un usuario.  **chfn** <nombreusuario> → cambia la información de un usuario.  **chsh** <nombreusuario> → cambia la shell a utilizar por un usuario.  **chage** <opciones ><nombreusuario> → cambia la información de caducidad de contraseñas. |

| **Comandos para gestionar grupos:** |
| --- |
| **groups** <nombrusuario> → ver los grupos a los que pertenece el usuario pasado como argumento.  **addgroup** <nombregrupo> → crea un grupo.  **delgroup** <nombregrupo> → elimina un grupo. No elimina un grupo primario de un usuario existente.  **groupdel** <nombregrupo> → elimina un grupo.  **groupmod** –n <nombrenuevo> <nombreactual> → cambia el nombre a un grupo.  **gpasswd** –a <usuario> <grupo> → añade un usuario a un grupo.  **gpasswd** –d <usuario> <grupo> → elimina un usuario de un grupo.  **newgrp** <grupo> → cambia el grupo activo al usuario actual. |

**Para administrar grupos hay que instalar gnome-system-tools**Existe el fichero */etc/profile* que contiene el perfil igual para todos los usuarios

**2-Gestión de permisos en GNU/Linux.**

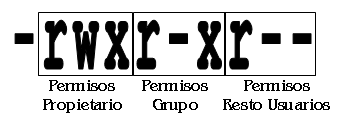
Los permisos básicos son:

-Lectura ® –ver contenido

-Escritura(w) --modificar contenido

-Ejecución(x) –ejecutar contenido

El comando ls -l nos permite ver los permisos



El primer guión puede ser:

* **-** fichero normal.
* **d** directorio o carpeta.
* **l** enlace simbólico.
* **c** dispositivo de caracteres.
* **b** dispositivo de bloques.
* **p** tubería.
* **s** socket.

**Para cambiar los permisos a un fichero debemos usar el comando “chmod”. Este comando podemos usarlo de dos maneras.**

Método 1: usando notación octal:

**chmod <permisos en octal> <fichero>**

| **Octal** | **Binario** | **Permisos** |
| --- | --- | --- |
| 0 | 000 | --- |
| 1 | 001 | --x |
| 2 | 010 | -w- |
| 3 | 011 | -wx |
| 4 | 100 | r-- |
| 5 | 101 | r-x |
| 6 | 110 | rw- |
| 7 | 111 | rwx |

Método 2: usando notación simbólica:

**chmod <usuario><operador><permiso> <fichero>**

Donde **<usuario>** puede ser:

**u** → propietario (user)

**g** → grupo (group)

**o** → el resto (other)

**a** → todos (all)

Donde **<operador>** puede ser:

**+** → añade el permiso.

**-** → quita el permiso.

**=** → para asignar exactamente un permiso concreto (eliminando cualquier otro que ya pudiese tener el fichero.

Donde **<permisos>** puede ser:

**r** → lectura

**w** → escritura

**x** → ejecución

**El comando umask sirve para establecer los permisos por defecto que tendrán los nuevos ficheros y directorios que creemos**

**La máscara que tiene un usuario se define en el fichero “/etc/profile”.**

Los permisos iniciales por defecto para un archivo son 666 y para una carpeta 777. Estos permisos son modificados por la máscara que tiene el usuario de la siguiente manera:

**<Permiso inicial> – <máscara del usuario> = <permiso del fichero>**

**-Para cambiar el propietario o el grupo de un fichero debemos ser el propietario o el usuario root. Los comandos que nos permiten hacer esto son:**

**CHOWN:**

El comando chown se utiliza para cambiar el propietario al que pertenece un fichero o directorio

chown [opciones] usuario[:grupo] fichero(s) o directorio(s)

**CHGRP**

El comando **chgrp** se utiliza para cambiar el grupo al que pertenece un fichero o directorio

chgrp [opciones] fichero(s) o directorio(s)

\*opciones:

-R : cambia recursivamente

–version : muestra la versión

-v : salida más descriptiva

–reference: cambia el grupo/usuario de un fichero, tomando como referencia el propietario de otro

**3- Gestión de procesos**

Cuando el usuario está interactuando con un proceso, se dice que este proceso está en primer plano.

Cuando el proceso se ejecuta "en silencio", sin necesidad de que el usuario interactúe con él se dice que el proceso está en segundo plano.

Un ejemplo típico de procesos en segundo plano son los demonios.

**Sólo un proceso puede estar en primer plano en un momento dado. Por el contrario, en segundo plano pueden existir varios procesos a la vez.**

**Por otro lado, un proceso puede pasarse de primer plano a segundo plano, y viceversa, cuantas veces se desee.**

Dentro del monitor del sistema podemos ver los diferentes procesos, a dichos procesos se le pueden aplicar una serie de operaciones, que son:

* **Detener el proceso**: pasa el proceso a estado suspendido.
* **Continuar el proceso**: reanuda un proceso que estaba detenido.
* **Finalizar el proceso**: pasa el proceso a estado terminado.
* **Matar el proceso**: pasa el proceso a estado terminado de forma inmediata.
* **Cambiar la prioridad**: permite variar la prioridad del proceso siendo 0 la prioridad normal y pudiendo oscilar entre los valores -20 (máxima prioridad) y 20 (mínima prioridad).

INFORMACIÓN DE PROCESOS

COMANDO **ps**: Informa del estado de los procesos

-COMANDO **ps -A, ps -e**: Muestra todos los procesos

-COMANDO **ps -p PID**: Selecciona procesos con un determinado PID (donde el PID es el identificador del proceso)

-COMANDO **ps -u usuario**: Selecciona procesos de un determinado usuario

-COMANDO **ps -f**: Muestra procesos con información detallada

-COMANDO **ps -F**: Muestra procesos con información detallada y extendida

-COMANDO **ps -j**: Muestra procesos con forma de árbol

-COMANDO **ps -o (orden1,orden2)**: Muestra procesos en formato definido por el usuario. Ej: ps -o user,pid

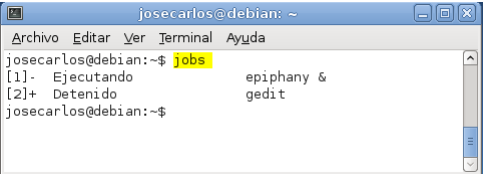
COMANDO **pstree**: Informa del estado de los procesos en forma de árbol

-COMANDO **pstree -G**: Salida estándar de pstree

-COMANDO **pstree -u**: Muestra el árbol con los usuarios

-COMANDO **pstree usuario**: Muestra las ramas del árbol correspondientes a un usuario

COMANDO **jobs**: Informa sobre los procesos que se están ejecutando en primer y segundo plano (Únicamente se pone jobs por comando)



Donde:

[x]: Número de proceso

+, -: + es proceso en primer plano y – es proceso en segundo plano

Estado: Ejecutando, Detenido

Proceso: Nombre del proceso

ADMINISTRACIÓN DE PROCESOS

COMANDO **kill**: Envía una señal al proceso

Las principales señales son:

1: SIGHUP Hang up.

2: SIGINT Interrumpir el proceso.

3: SIGQUIT Salir del proceso.

9: SIGKILL Terminar el proceso (forzado).

15: SIGTERM Terminar el proceso (solicitado)

18: SIGCONT Continuar la ejecución.

19: SIGSTOP Detener el proceso.

20: SIGSTP Como SIGSTOP.

Dichos comandos se pueden introducir de dos formas:

-Por su nombre: kill -nombreseñal PID (pid se ve con el comando ps)

EJ: kill -SIGKILL 43242

-Por su número de señal: kill -numeroseñal PID

EJ: kill -9 43242

CAMBIAR PRIORIDAD DE PROCESOS

COMANDO (**nice -n prioridad proceso)**: Asigna una prioridad a un proceso antes de iniciarse

EJ: nice -n 5 gedit

COMANDO (**renice prioridad PID)**: Cambia la prioridad de un proceso ya iniciado sin reiniciarlo

EJ: renice 8 2778

CAMBIAR DE PLANO LOS PROCESOS

COMANDO (**proceso &)**: Inicia un proceso en segundo plano

EJ: gnomines & (ejecutará el buscaminas en segundo plano)

COMANDO **(bg %numeroproceso)**: Ejecuta un proceso en segundo plano

EJ: bg %1 (el 1 lo obtenemos haciendo un jobs y viendo el número de proceso)



COMANDO **(fg %numeroproceso)**: Ejecuta un proceso en primer plano

EJ: fg %1 (el 1 lo obtenemos haciendo un jobs y viendo el número de proceso)



COMANDO **(nohup nombreproceso)**: Ejecuta un proceso haciendole inmune a los hang ups (Cuelgues)

EJ: nohup gnomines

**4- Gestión de servicios**

Los servicios en Linux al igual que en Windows son programas que se ejecutan en segundo plano y que no requieren de la intervención del usuario

Normalmente, a los servicios en Linux se les llama demonios (daemons)

-Tradicionalmente init ha sido el programa encargado de controlar la secuencia de arranque de las distribuciones linux (System -V).

-Sin embargo, desde Ubuntu 6.10 se añadió al init System -V un sistema llamado Upstart, que estaba basado en eventos y no en niveles (como System -V)

-Finalmente en el 2011 un nuevo sistema llamado systemd. A partir de Ubuntu 15.04 se dejó de usar Upstart por systemd, su principal diferencia es que **ya no requiere satisfacer ciertas dependencias al momento de cargar los componentes**

**Los métodos son:**

#### **Método 1: el clásico de System-V**

* $sudo /etc/init.d/<nombre\_servicio> start (inicia el servicio)
* $sudo /etc/init.d/<nombre\_ servicio> stop (detiene el servicio)
* $sudo /etc/init.d/<nombre\_ servicio> restart (detiene e inicia el servicio)
* $sudo /etc/init.d/<nombre\_ servicio> reload (carga la configuración nuevamente sin detener el servicio)
* $sudo /etc/init.d/<nombre\_ servicio> status (muestra el estado del servicio)

#### **Método 2: usando el comando service**

* $sudo service <nombre\_ servicio> start (inicia el servicio)
* $sudo service <nombre\_ servicio> stop (detiene el servicio)
* $sudo service <nombre\_ servicio> restart (detiene e inicia el servicio)
* $sudo service <nombre\_ servicio> reload (carga la configuración nuevamente sin detener el servicio)
* $sudo service <nombre\_ servicio> status (muestra el estado del servicio)

#### **Método 3: usando los comandos de upstart (obsoleto) \***

* $sudo start <nombre\_ servicio> (inicia el servicio)
* $sudo stop <nombre\_ servicio> (detiene el servicio)
* $sudo restart <nombre\_ servicio> (detiene e inicia el servicio)
* $sudo reload <nombre\_ servicio> (carga la configuración nueva-mente sin detener el servicio)
* $sudo status <nombre\_ servicio> (muestra el estado del servicio)

#### **Método 4: usando los comandos de systemd**

* **$sudo systemctl start <nombre\_ servicio>.service (inicia el servicio)**
* **$sudo systemctl stop <nombre\_ servicio>.service (detiene el servicio)**
* **$sudo systemctl restart <nombre\_ servicio>.service (detiene e inicia el servicio)**
* **$sudo systemctl reload <nombre\_ servicio>.service (carga la configuración nueva-mente sin detener el servicio)**
* **$sudo systemctl status <nombre\_ servicio>.service (muestra el estado del servicio)**

**SERVICIOS BASADOS EN RUNLEVELS**

El sistema System-V sólo es capaz de ejecutar o parar procesos ante la entrada en un runlevel

El runlevel es cada uno de los estados de ejecución en que se puede encontrar el sistema Linux, que son:

* Runlevel 0: Apagar la máquina.
* Runlevel 1: Arrancar en Modo Monousuario
* Runlevel 2 al 5: Arrancar en Modo Multiusuario con entorno gráfico si está instalado.
* Runlevel 6: Reiniciar la máquina.

Para indicar los procesos que se han de ejecutar o parar en cada runlevel, existen los directorios “/etc/rcX.d/”, donde X es el número de runlevel.

#### 

#### 

#### 

#### **-Seleccionar un runlevel en el arranque de Ubuntu**

* En el fichero “/etc/init/rc-sysinit.conf” modificamos la siguiente línea:

------------------------ Fichero /etc/init/rc-sysinit.conf -------------------------

[…]

env DEFAULT\_RUNLEVEL=2

[…]

----------------------- Fin Fichero /etc/init/rc-sysinit.conf ----------------------

Con independencia del runlevel seleccionado se ejecutarán todos los servicios de /etc/rcS.d/.

#### **-Seleccionar un runlevel una vez iniciado el sistema**

* Ver el runlevel en el que nos encontramos:

$runlevel

* Cambiar el sistema de runlevel:

$init <número que indique el runlevel al que queremos ir>

**5- Monitorización del sistema**

Las herramientas que se utilizan para la monitorización en Linux podemos agruparlas de la siguiente manera:

**-Herramientas integradas:** son herramientas propias del sistema, que existen en la gran mayoría de las distribuciones Linux de forma predeterminada.

**-Monitor del sistema:** es una herramienta que permite monitorizar los procesos que se encuentran en ejecución en el sistema

**-Herramientas Sysstat**: es una colección de herramientas de monitorización que, además de proporcionar datos de rendimiento en tiempo real permite almacenarlos

**HERRAMIENTAS INTEGRADAS**

**uptime**: monitoriza la carga del sistema

**time**: monitoriza el tiempo de ejecución de un programa

**top**: monitoriza la actividad de los procesos (TODOS)

**ps**: monitoriza la actividad de los procesos (Solo los ejecutados por el usuario)

**vmstat**: monitoriza la actividad de la memoria (física, virtual,transferencias,interrupciones,uso del procesador,etc)

**free**: monitoriza la actividad de la memoria (memoria física y swap)

**w**: monitoriza la actividad de los usuarios del sistema

**Directorio /proc:**  contiene un sistema de archivos imaginario o virtual. Este no existe físicamente en disco, sino que el núcleo lo crea en memoria. Se utiliza para ofrecer información relacionada con el sistema

Dentro del directorio proc encontramos dos tipos de contenido: directorios numerados y ficheros de información del sistema

**MONITOR DEL SISTEMA**

El Monitor del sistema en Linux es el interfaz gráfico de la orden **top** vista anteriormente

Se encuentra en la ruta: **Sistema→Administración→Monitor del sistema**

Si no se encuentra, puede instalarse mediante **sudo apt install gnome-system-monitor**

**Las opciones que muestra dicho monitor son:**

* **Sistema:** proporciona información básica como puede ser la versión del sistema operativo, el procesador integrado o la memoria RAM disponible.
* **Procesos**: contiene un listado con los procesos del sistema, su estado y la carga media de los últimos minutos
* **Recursos**: informa a través de gráficos dinámicos el uso de los principales recursos del sistema (el procesador, la memoria y la red)(Esta pestaña puede ser modificada)
* **Sistemas de archivos**: muestra un listado con los sistemas de archivos montados y sus principales características (espacio libre, punto de montaje, etc…).

**MONITORIZACIÓN CON SYSSTAT**

Sysstat es un conjunto de herramientas orientadas a monitorizar el rendimiento de equipos con sistema operativo Linux.

Algunas herramientas que ofrece sysstat son:

* **mpstat**: genera informes del rendimiento de cada procesador del sistema.
* **iostat**: genera informes de la actividad de la CPU y de los dispositivos de E/S.
* **pidstat**: genera informes de las tareas gestionadas por el kernel.
* **sar**: recoge la información de rendimiento de todas las áreas del sistema (CPU, memoria, red,…).
* **sadf**: permite exportar los datos recopilados por sar en diferentes formatos (XML, CVS,…).

**LOGS O BITÁCORAS**

Son archivos de registro que ayudan al administrador en la monitorización y en el mantenimiento de una máquina

Estos registros ayudan a la localización de errores, ataques maliciosos y hasta podemos saber que hacen nuestros usuarios en el sistema

-Los archivos de registro o logs se encuentran en la carpeta **/var/log**

**Podemos ver el contenido de los logs de múltiples formas, como por ejemplo:**

**-Comando cat: $cat /var/log/syslog** EJ: **$cat /var/log/syslog | grep ntp**

(para buscar cadenas que tengan ntp)

**-Comando less: $less /var/log/syslog**

**-Comando tail: $tail /var/log/syslog** EJ:**$tail –n 20 /var/log/syslog**

(para mostrar 20 líneas del log)

**6- Programación de tareas**

Antes de programar las tareas hay que comprobar que el servicio cron se encuentra en ejecución mediante el comando:

**service cron status**

Para modificar el fichero de configuración de cron, usamos:

**crontab -e**

Además, crontab también permite al administrador crear tareas modificando el fichero /etc/crontab ,siguiendo el siguiente formato:



**Para asegurar el sistema sólo el usuario root puede modificar los scripts que ejecuta crontab.**

Otra forma de poder programar tareas es guardar el script o programa que quiere ejecutar en las siguientes carpetas de configuración de cron:

**/etc/cron.hourly # Ejecuta el script cada hora**

**/etc/cron.daily # Ejecuta el script diariamente**

**/etc/cron.weekly # Ejecuta el script semanalmente**

**/etc/cron.monthly # Ejecuta el script mensualmente**

Ejemplos:

**55 23 \* \* 0 /home/usuario/copia\_semanal.sh**

* Ejecutará el script “copia\_semanal” todos los domingos a las 23:55h.

**0 9,17 \* \* 1-5 /home/usuario/copia.sh**

* Ejecuta el script “copia” de lunes a viernes a las 9:00h y a las 17:00h.

**30 17 \* \* 1,2,3,4,5 /home/usuario/copia.sh**

* Ejecuta el script a las 17:30h todos los días de lunes a viernes.

**00 12 1,15,28 \* \* /home/usuario/copia.sh**

* Ejecuta el script a las 12 a.m. todos los días 1, 15 y 28 de cada mes.

**COMANDO AT**

-El comando o programa at, permite planificar la ejecución de ciertas tareas en un momento dado. Este comando pueden ejecutarlo los usuarios y el root.

Su sintaxis es: **at [opciones] [hora] [fecha]**

**Ejemplo:** at now + 5 minutes Pueden ser minutes (minutos), hours (horas) o days (días). Esto significa que la tarea se ejecutará en 5 minutos

**Otra forma es con el formato HH:MM**:at 21:00 102411 at> cp /etc/network/interfaces /backup

-Comando para ver tareas pendientes: **atq**

-Comando para ver el contenido de una tarea: **at -c numerotarea**

-Comando para eliminar tareas: atrm **numerotarea**

**7- Mantenimiento y optimización del sistema**

**-BleachBit, herramienta para limpiar el sistema:**

BleachBit es una herramienta para Linux con la cual podrás hacer mantenimiento del sistema analizando cachés, historiales y borrando archivos basura. Añade mejoras de privacidad, es compatible con varios programas y es realmente eficaz

**-FSlint:**

FSlint es un software libre que nos permite encontrar ficheros duplicados, carpetas vacías y te permitirá eliminarlos o copiarlos hacia otra carpeta para controlarlos

**8- Configuración de Grub**

Cuando se ejecuta Grub al iniciar el arranque lee su configuración del fichero **/boot/grub/grub.cfg** , pero si queremos modificar algo no podemos hacerlo en el propio archivo

**update grub**

**Ficheros de configuración de grub**

**/boot/grub/grub.cfg:** este fichero no se modifica manualmente, sino que toma su configuración al leer el comando update-grub.Algunas de las modificaciones más importantes son:

**-GRUB\_DEFAULT= 0**: Permite seleccionar la opción por defecto del menú. Con la opción **0** hacemos que se seleccione por defecto el primer "menuentry" en el fichero **grub.cfg**, con la opción **1** la segunda y así sucesivamente

**GRUB\_DEFAULT="xxxx":** Permite seleccionar una entrada del menú exacta, incluyendo símbolos de entrecomillado

EJ: GRUB\_DEFAULT="Ubuntu, Linux 2.6.31-9-generic”

**-GRUB\_TIMEOUT=10:** Le ponemos el **tiempo de espera** hasta ejecutar la entrada que tenemos puesta como default si no hemos pulsado ninguna tecla. Establecer el valor a -1 hará que se muestre el menú hasta que el usuario haga una selección.

**GRUB\_HIDDEN\_TIMEOUT=x:** Le ponemos **tiempo de espera** pero no se mostrar ningún menú de selección

**Contenido del directorio /etc/grub.d**

Dentro de este directorio se encuentran los ficheros encargados de detectar los sistemas operativos instalados en nuestro equipo y de generar las entradas correspondientes en grub.cfg cuando se ejecuta **update-grub**