

## BLOQUE 4: Gestión de Usuarios y Procesos

### **ÍNDICE DE CONTENIDO**

|  |                  |
|--|------------------|
| <b><i>Gestión de usuarios por Línea de Comandos en Linux.....</i></b>                | <b><i>2</i></b>  |
| Configuración de Usuarios y Grupos .....   | 2                |
| Comandos de Gestión de Usuarios .....  | 3                |
| Comandos de Gestión de Grupos.....   | 5                |
| Seguridad de cuentas de usuarios y contraseñas: .....                                | 6                |
| Acceso a recursos y permisos locales.....  | 7                |
| Modificación de permisos: .....  | 7                |
| Configuración de perfiles: .....   | 9                |
| Variables:.....  | 9                |
| Alias: .....   | 10               |
| <b><i>Gestión de usuarios por Interfaz Gráfica de Windows .....</i></b>              | <b><i>10</i></b> |
| <b><i>Gestión de procesos por Línea de Comandos en Linux .....</i></b>               | <b><i>10</i></b> |
| Identificador de procesos:.....  | 11               |
| Ejecución en primer y segundo plano .....  | 12               |
| Prioridad de órdenes.....  | 13               |
| Envío de señales: .....  | 13               |
| <b><i>Gestión de procesos por Interfaz gráfica en Windows .....</i></b>              | <b><i>14</i></b> |
| <b><i>Automatización de tareas de Linux.....</i></b>                                 | <b><i>14</i></b> |
| <b><i>Monitorización y Gestión del Sistema. Evaluación de Prestaciones .....</i></b> | <b><i>15</i></b> |
| Linux .....  | 15               |
| Windows: .....   | 15               |
| <b><i>Aplicaciones para el mantenimiento .....</i></b>                               | <b><i>15</i></b> |

A la fuerza, los sistemas operativos tienen que ofrecer un soporte seguro, amigable y que permita actuar junto a las aplicaciones que esta tiene. Se tiene que poder manejar de forma amigable la gestión de usuarios.

También tienen que gestionar la ejecución de los procesos, haciendo que los procesos (bajo la supervisión de un administrador) actúen respetando la eficiencia y la seguridad del sistema

Un administrador se va a encargar de la gestión de los usuarios y los procesos para asegurar que se cumplan las normas de seguridad del sistema

## Gestión de usuarios por Línea de Comandos en Linux

### Configuración de Usuarios y Grupos

Linux gestiona los usuarios y grupos por medio de los archivos `/etc/passwd` y `/etc/group` principalmente. Cualquier edición solo se va a poder hacer por medio del root.

Otros a tener en cuenta son `/etc/sudoers` y `/etc/shadow`.

- `/etc/passwd` va a almacenar la información de los usuarios en cada línea. Cada campo que hay da una información diferente, delimitado por ":"
  - o Campo 1: Nombre de usuario
  - o Campo 2: Contraseña (normalmente encriptada (se verá una x))
  - o Campo 3: ID de usuario (0: sudo, del 1 al 99: predeterminadas, del 100 al 999: cuentas administrativas, a partir del 1000: cuentas nuevas)
  - o Campo 4: GID: ID del grupo (Principal siempre, y puede pertenecer a más de un grupo)
  - o Campo 5: Información personal (Datos de localización, teléfono...)
  - o Campo 6: Carpeta personal (\$HOME desde el propio usuario, `/home/usuario`)
  - o Campo 7: Shell (Por defecto `/bin/bash`)

Los usuarios administradores se entienden que son los que disponen de privilegios sin necesidad de ser root. Se puede usar comandos con privilegios por medio del comando `sudo` o también se puede hacer si se pertenece a un grupo con el privilegio otorgado

Los grupos se utilizan para facilitar la gestión de privilegios. Se crea un grupo con privilegios y todos los usuarios que se guarden en ese grupo heredarán dichos privilegios.

- `/etc/group`: Guarda la información de los grupos como `/etc/passwd` guarda la información de los usuarios.
  - o Campo 1: Nombre del grupo, identificado al creador del grupo
  - o Campo 2: Contraseña (encriptada con x (se puede buscar en `/etc/gshadow`))
  - o Campo 3: GID: ID de grupo
  - o Campo 4: Usuarios pertenecientes al grupo de forma secundaria (separados por ",")

El grupo secundario se va a utilizar para ir otorgando más permisos sobre archivos.

Cada usuario ha de pertenecer a un grupo principal, aunque pueden pertenecer a más grupos de forma secundaria, lo cual se verá reflejado en el campo 4 de `/etc/group`

Hay un usuario especial con todos los permisos, llamado super usuario o `root`, el cual está deshabilitado en varias ocasiones para que no todo el mundo pueda cambiar cosas y fastidiar el sistema. El prompt del sistema de los usuarios es "\$", mientras que el prompt del super usuario es "#".

## Pasos para habilitar al usuario root:

1. **Añadir una contraseña al usuario root** por medio del comando *"sudo passwd root"*

```
erikat@Ubuntu:~/Escritorio$ sudo passwd root
[sudo] contraseña para erikat:
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña no supera la verificación de diccionario -
Está basada en una palabra del diccionario.
Vuelva a escribir la nueva contraseña:
passwd: contraseña actualizada correctamente
```

2. **Iniciar sesión en root** por medio de *"su root"* y meter la contraseña de antes

```
erikat@Ubuntu:~/Escritorio$ su root
Contraseña:
root@Ubuntu:/home/erikat/Escritorio#
```

3. Una vez se acabe de usar **cerrar la sesión por medio de "exit"**

```
root@Ubuntu:/home/erikat/Escritorio# exit
exit
erikat@Ubuntu:~/Escritorio$
```

Es posible ejecutar los comandos del super usuario con el comando *"sudo"*. Pedirá una contraseña para usar dichos permisos. También, guardará por seguridad los datos de acceso de los usuarios que han usado dicho comando. También, como medida de seguridad, se saldrá pasado unos minutos para devolver los permisos originales al usuario.

Se puede ejecutar los comandos de otros usuarios si se escribe *"sudo -u (usuario)"* o *"su [-] [usuario]"*

Root (superusuario o sudo) tiene un nivel de acceso limitado a comandos, y se pedirá siempre la contraseña del usuario que quiere usar el sudo. Puede editar */etc/passwd*, añadiendo, modificando y eliminando campos del mismo

Administrador va a tener todos los permisos del usuario al que se conecta

En */etc/sudoers*, se indican:

- Campo 1: Usuarios
- Campo 2: Máquinas a las que va a interferir
- Campo 3: Usuarios
- Campo 4:
- Campo 5: Comandos que afecta

## Comandos de Gestión de Usuarios

### Añadir nuevo usuario: -useradd

Sintaxis: *[sudo] -useradd [-g grupo] [-G grupo, [grupo, grupo...]] [-d directorio\_trabajo [-m]] [-p contraseña] [-s Shell] login*

La diferencia entre g y G es que g es el grupo principal y G son los secundarios  
-d establece un directorio existente para ese usuario. Si no se especifica, se toma el */home/(login)*. -m crea el directorio de trabajos y crea los archivos de configuración (y otros ficheros que se añadan) del */etc/skel*.

-p es para poner una contraseña

-s establece un Shell, por defecto es */bin/bash*

### **Modificar usuario: -usermod**

Sintaxis: [sudo] usermod [-c comentario] [-g grupo] [-G grupo] [-d directorio trabajo [-m]] [-p contraseña] [-e fecha] [-f días] [-l nuevo login] [-L] [-U] [-s shell] login

- c añade un comentario en el hueco 5 del login
- p combinado con -e (fecha) o -f (días) pone una caducidad a la contraseña
- l cambia el nombre de usuario
- L bloquea la cuenta
- U desbloquea la cuenta
- s cambia el Shell

### **Ver información y cambiar otras opciones: -finger y -chfn**

Sintaxis de finger: [sudo] finger

Se puede ver con él los diferentes usuarios y sus últimos logins

Sintaxis de chfn: [sudo] chfn [-w para escribir] algo Login

Añade información de los logins, como un numero de teléfono o así

### **Eliminar usuario: -userdel**

Sintaxis: [sudo] userdel [-r] login

- r permite borrar la carpeta del usuario

### Comprobar usuarios conectados: -who

Sintaxis: who [am i] [-u] [-H] [-q]

-u: Muestra informacion de usuarios conectados

-am i: Muestra el usuario actual

-H imprime cabeceras

-q muestra solamente los logins y los usuarios conectados

### Comandos de Gestión de Grupos

#### Añadir grupos: -groupadd

Sintaxis: [sudo] groupadd [-g GID] grupo

-g sirve para darle un GID personalizado, si se omite, se asigna uno por defecto

#### Eliminar grupos: -groupdel

Sintaxis: [sudo] groupdel grupo

Necesitará eliminar antes todas las cuentas de usuario que tengan ese grupo de principal

#### Modificar grupos: -groupmod

Sintaxis: groupmod [-g GID] [-n nuevonombre] grupo

-g servirá para cambiar el GID

-n servirá para cambiar el nombre del grupo a uno nuevo

***Los archivos ligados al grupo tendrán que cambiarse manualmente***

#### Comprobación de usuarios a grupos: -groups o -id

Sintaxis: groups [usuario], id [usuario]

id muestra más información, identifica el UID del usuario y el GID de cada grupo

#### Añadir usuarios a grupos: -adduser

Sintaxis: adduser usuario grupo

Añade un usuario a un grupo específico

#### Eliminar usuarios de grupos: -deluser

Sintaxis: deluser usuario grupo

Elimina un usuario de un grupo específico

### Usuarios y Grupos predeterminados:

Además del root, existen grupos por defecto para otorgar permisos y facilitar la administración del sistema.

| Grupo  | Descripción  |
|--------|--|
| adm    | Grupo de administración que permite accesos a archivos de registro y comandos como sudo y su |
| users  | Grupo de usuarios estándar   |
| nobody | Sin privilegios  |

|                      |  |
|----------------------|--|
| root (Super usuario) | Administración sin restricciones sobre todo el sistema |
| tty                  | Aporta privilegios sobre algunos dispositivos          |
| lpadmin              | Da privilegios sobre dispositivos de puerto paralelo   |

### Modificación de propietarios y grupos sobre archivos: **-chown** y **-chgrp**

Sintaxis: [sudo] chown [-R] [-h] nuevopropietario[.nuevogrupo] fichero

-R: Recursivo

-h: Afecta a enlaces simbólicos

Sintaxis: [sudo] chgrp [-R] grupo ficheros

-R: Recursivo

### Seguridad de cuentas de usuarios y contraseñas:

Linux emplea un sistema de autenticación llamado Linux-PAM.

1. Login
2. Contraseña encriptada. Dependiendo del algoritmo, el cifrado será más o menos largo, y comienza con \$character\$.
3. Días transcurridos desde el 01/01/1970 (inicio de la época GNU) hasta el día que la contraseña fue cambiada por última vez
4. Número de días que tienen que pasar hasta que el usuario tenga que cambiar la contraseña
5. Número de días que será una contraseña válida
6. Número de días hasta que el sistema avisará para cambiar la contraseña
7. Número de días después de que la contraseña caduque para deshabilitar la cuenta
8. Días transcurridos desde el 01/01/1970 hasta la fecha donde será deshabilitada

Para asignar o cambiar contraseñas: **-passwd**

Sintaxis: [sudo] [protocolo] passwd [opciones] [usuario] contraseña

-el protocolo será el que queramos meter, por ejemplo, "openssl"

Cada contraseña debería ser única y no intuitiva

### Gestión de contraseñas de usuarios: **-chage** / **-passwd** / **-usermod**

Sintaxis: chage [opciones] login

-d: Establece el ultimo día del ultimo cambio de la contraseña. Si pone un 0, necesita actualizar

-E: Establece una fecha de expiración. Si tiene un -1 no hay limitación, si tiene un 1 se elimina de inmediato

-h: Ayuda

-i: usa YY-MM-DD para imprimir fechas

-l: Establece el numero de días de inactividad después de que una contraseña se haya bloqueado hasta que se borre. Si hay -1 no hay limitación, si hay un 0 se deshabilita nada mas caducar la contraseña

-l: Muestra la información de la cuenta

-m indica el mínimo de días para poder cambiar la contraseña, 0 = no hay limitación

- M indica el máximo de días desde que se cambio la ultima vez, -1= sin limitación
- w: Indica el numero de días de preaviso antes de que caduque

#### Acceso a recursos y permisos locales

El i-nodo de los archivos almacena el UID del propietario del archivo, el grupo propietario y la máscara de permisos, representada de la siguiente manera:



r: Permiso de lectura  
w: Permiso de escritura  
x: Permiso de ejecución

Usuario: El usuario propietario del archivo  
Grupo: El grupo propietario  
Otros: Los usuarios y grupos invitados

#### Bits raros:

- **Set-uid: (-rws r-x r-x)** Toma el UID del propietario al ser ejecutado, es decir, usa los permisos del usuario propietario en vez de los permisos del usuario que se está usando actualmente (Ejemplo: Si estamos en el usuario "Paco" abriendo el archivo del usuario "María", lo abriremos con los permisos del usuario "María")
- **Get-uid: (- rwx r-s r-x)** Toma el GID del grupo propietario (archivo) o fuerza que el usuario pertenezca al grupo del directorio
- **Sticky-bit: (d rwx rwx rwt)** Solo el propietario de un archivo o del directorio con sticky bit activo lo puede borrar, es decir, aunque el resto de usuarios tengan el permiso de escribir, solo el usuario del archivo lo puede borrar

#### Ejercicios:

- Máscara de permisos de un archivo regular en el que el propietario puede leer y escribir, los usuarios del grupo pueden leer y el resto de usuarios no disponen de privilegios: **- rw- r-- ---**
- Máscara de permisos de un archivo regular en el que el propietario puede leer y escribir, los usuarios pueden leer y el resto pueden leer y escribir: **- rw- r-- rw-**
- Máscara de permisos de un archivo regular en el que el propietario no dispone de privilegios, los usuarios del grupo pueden leer y ejecutar y el resto pueden leer, escribir y ejecutar: **- --- rw- rwx**
- Máscara de permisos de un directorio en el que el propietario puede leer, escribir y acceder a su contenido, los usuarios pueden leer su contenido y el resto no tienen privilegios: **d rwx r-- ---**

#### Modificación de permisos:

Sintaxis: **chmod [-R] permisos(OCTAL) archivos**

- Para sticky bit, sumas 1000 en octal
- Para set-gid sumas 2000 en octal
- Para set-uid sumas 4000 en octal

**MODO SIMBÓLICO:** Combinación de letras y símbolos [u][g][o]{+|-|=}[r][x][w][s][t]

- **Destinatario:**
  - Usuario (u)
  - Grupo (g)
  - Otros (o)
  - Todos (a)
- **Modificación:**
  - Añade (+)
  - Quitar (-)
  - Añade los permisos y quita el resto (=)
- **Permisos:**
  - Lectura “r”
  - Escritura “w”
  - Ejecución “x”
  - Set-uid/gid “s”
  - Sticky-Bit “t”

**PERMISOS ORIGINALES:** El permiso original (o por defecto) de los archivos es 0666, y de los directorios 0777. Por defecto, la máscara de permisos va a ser 0022 o 0002. Esto último se puede ver con el comando **umask**.

**Sintaxis: umask [máscara]**

Si no se mete máscara, te dice la máscara de permisos por defecto de tu equipo

- **Pasos para aplicar la máscara de permisos:**
  1. Convertir la máscara de permisos de octal a binario
  2. Aplicar el NOT sobre la máscara
  3. Realizar AND entre los permisos originales y la máscara



### Configuración de perfiles:

Es la configuración del entorno de trabajo de un usuario en concreto. Va a durar desde que el usuario inicia sesión hasta que se desconecta. Se rigen por unos archivos de configuración globales (afectan a todos y es necesario tener permisos de administrador para editarlos) y los locales (afectan al propio usuario y lo puede modificar)

#### Localización de archivos globales:

- /etc/skel: Plantilla de creación de perfiles de usuario
- /etc/profile: Configuración genérica de perfiles cuando se inicia sesión como login Shell (login por usuario)
- /etc/bash.bashrc: Configuración genérica cuando se inicia sesión con el Shell bash interactivo (interfaz de línea de comandos)

#### Archivos locales:

- ~/.bashrc: Configuración local del usuario cuando inicia sesión con bash Shell
- ~/.bash\_logout: Configuración local del usuario cuando termina su sesión
- ~/.profile: Configuración local del usuario cuando inicia sesión en el sistema con un Shell de inicio de sesión.

### Variables:

Las variables son identificadores que almacenan cadenas de caracteres. Pueden tener varios valores asignados.

#### Tipos de variables:

- **Variables de entorno:** Reconocidas y utilizadas por procesos de dentro de Shell  
Para ver las listas de variables globales se puede ver con el comando env o printenv
  - o SHELL: Shell por defecto `SHELL=/bin/bash`
  - o USER: Usuario actual `USER=erikat`
  - o PWD: Directorio de trabajo `PWD=/home/erikat/Escritorio`
  - o OLDPWD: Directorio de trabajo previo `OLDPWD=/`
  - o PATH: Rutas de directorios con ejecutables `PATH=/usr/local/bin:/usr/bin:/usr/sbin:/usr/games:/usr/local/games:/snap/bin:/usr/lib`
  - o HOME: Directorio home del usuario `HOME=/home/erikat`
- **Variables locales (de Shell):** Solo pueden ser interpretadas por el Shell:
  - o **HOSTNAME:** Nombre del equipo `HOSTNAME=Ubuntu`
  - o **IFS:** Valores de separación en la línea de comandos `IFS=$' \t\n'`
  - o **PS1:** Valor del prompt  
`PS1='\[\e]0;\u@h:\w\[\e]S{debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@h\[\033[00m\]:\[\033[02;34m\]\w\[\033[00m\]}$ '`
  - o **UID:** UID del usuario actual `UID=1000`

#### Establecimiento de variables: export

Sintaxis: export [-n] VARIABLE

La n pasa una variable global a local

Sin la n, establece una variable como global

#### Eliminación de una variable: unset

Sintaxis: unset VARIABLE

#### Alias:

Los alias sirven para ejecutar comandos de manera personalizada, será una variable local a menos que se configure en bashrc

**Creación de alias: alias nombre='comando'**

**Eliminación de alias: unalias nombre**

#### Gestión de usuarios por Interfaz Gráfica de Windows

Se realiza a través del apartado "Cuentas" en la aplicación Configuración. Se hará seleccionando "Agregar otra persona a este equipo" y se nos preguntará cómo iniciará sesión esa persona. Si no se desea usar los datos de inicio de sesión, se puede saltar ese paso. Al iniciar, pide la contraseña y sus oportunas pistas por si se olvida la contraseña.

Otro método es accediendo al panel de control -> Cuentas de usuario -> Cuentas de usuario -> Administrar otra cuenta -> Agregar un nuevo usuario en Configuración

Por defecto, Windows crea varias cuentas administrativas que no se encuentran habilitadas por seguridad, aunque a través de esta ventana se pueden activarse.

- Administrador (root de Linux)
- Invitado: Pocos privilegios (acceden de forma esporádica)

#### Gestión de procesos por Línea de Comandos en Linux

Los sistemas operativos gestionan todos los procesos (o tareas) mediante operaciones de **creación, comunicación, compartición y finalización** de los procesos. Van a ser administrados por el sistema como un recurso más. El planificador de procesos se encarga de realizar las tareas de forma concreta a la hora de marcar los ritmos, atendiendo siempre al algoritmo de planificación:

Se asigna una pequeña ventana de tiempo (cuantum) a aquellos procesos que están ocupando la CPU. Pasado ese cuantum, se genera una interrupción en el tiempo para que salga el proceso de la CPU y entre otro. Con esto se evita que los procesos que necesitan ocupar mucho tiempo controlen toda la CPU.

Cada proceso tiene una prioridad,

La estructura mas representativa va a ser e PCB (Bloque de Control de Procesos), y se va a almacenar toda la información de procesos.

Procesos:

- Nuevo: Proceso de creación de la estructura de datos del proceso, se le asigna también un espacio.
- Listo: Los procesos que se acaban de crear o que han vuelto de la lista de bloqueados y están listos para pasar a ejecución
- Ejecutándose: El proceso esta ocupando tiempo de CPU. Puede pasar a bloqueados una vez pase el tiempo.
- Finalizado: El proceso ha terminado de ejecutarse y deja libre todos los recursos y estructuras de datos (PCD) que estuviera usando

- Bloqueados: Estando en proceso de ejecución, se pone en espera porque hay otro proceso de mayor prioridad tomando los recursos que ocupa el proceso. Se pasa a la cola de Listo en cuanto se pueda volver a comenzar el proceso

El procesador puede ejecutar las instrucciones de dos modos:

- Modo Usuario:
- Modo Kernel o privilegiado:

La ejecución de los procesos puede ser de varios modos:

- Por lotes o batch: Se lanza un conjunto de tareas a realizar y se ejecutan todas ellas unas detrás de otras (secuencialmente), **sin intervención del usuario**. Ejemplos: Copias de seguridad
- Interactivos: Interviene el usuario, le solicitan continuamente su interacción para su continuidad
- Servicios (demonios): Comienzan automáticamente cuando se inicia el sistema. Ejemplo: DNS

Identificador de procesos:

Los procesos disponen de:

- Identificador del proceso: PID
- Identificación del proceso Padre: PPID
- Usuario propietario
- Valores del estado del procesador
- Estado
- Ficheros abiertos

Toda esa información se guarda en el PCB, y se puede obtener su información con el comando **ps [modificador]**

Modificadores:

- ps -e: Muestra todos los procesos en ejecución del sistema
- ps -u [usuario]: Muestra los procesos que pertenecen a un usuario
- ps -f: Salida de formato completo, muestra todo lo que sale en el PCB
- ps -l: Formato largo de salida
- ps -p [PID]: Muestra información del proceso con dicho PID
- ps -t [tty]: Muestra los procesos que se ejecutan en un terminal en particular
- ps -aux: Combina varias opciones (a, u, x) para mostrar la salida de todos los procesos (a) de todos los usuarios(u), incluidos los procesos que no tienen asignados Shell (x)

El comando PS puede mostrar la siguiente información establecida en la cabecera:

- Hueco 1: UID Del propietario del proceso
- Hueco 2: PID del proceso
- Hueco 3: PPID del proceso (PID del proceso padre)
- Hueco 4: Índice de utilización reciente
- Hueco 5: Tiempo de inicio del proceso
- Hueco 6: Terminal de Lanzamiento (TTY)
- Hueco 7: Tiempo de CPU Consumido (TIME)
- Hueco 8: Orden de ejecución (COMMAND)

- Hueco 9: Tamaño del proceso en la memoria virtual (VSZ)
- Hueco 10: Tamaño de memoria física residente del proceso (RSS)
- Hueco 11: Porcentaje de uso de CPU (%CPU)
- Hueco 12: Porcentaje de uso de memoria (%MEM)
- Hueco 13: Estado del procesador (STAT / S)

#### Estados del procesador:

- **R**: Ejecutándose o listo para ser ejecutado (Runnable)
- **S**: Bloqueado o durmiendo (Sleeping)
- **T**: Parado
- **Z**: Zombi (Proceso muerto pero el proceso padre no ha reconocido su muerte)
- **I (idle)**: Inactivo
- **N**: Prioridad menor de lo normal (NICE)
- **<**: Prioridad mayor de lo normal
- **+**: Proceso en primer plano
- **s**: Proceso líder
- **l**: Proceso multi hilo (un mismo proceso con diferentes tareas ejecutadas en paralelo)

El comando ***ps tree*** muestra el árbol de procesos

Para ver el estado del sistema en tiempo real está el comando ***top***

Con el comando ***NICE***, muestra el grado de “amabilidad” para ceder la prioridad a otros procesos

#### ***TOP***:

En la primera línea se ve la hora actual, el tiempo de uso y cuántos usuarios están usando el sistema. También muestra el tiempo promedio de carga (Load average: (1, 5 y 15 minutos)).

En la segunda línea aparece el total de tareas, los procesos ejecutándose o listo (R), los bloqueados (S), los parados (T) y zombis (Z)

En la tercera línea aparece el porcentaje del CPU, los tiempos de uso del usuario (us), sistema (is), nice (ni), idle (id), procesos de hardware (hi), procesos de software (si), tiempos de espera (wa) interrupciones de software (st)

En la cuarta línea aparece la Memoria física total, libre, usada y en caché (MiB)

En la quinta línea, el MiB de intercambio: total, libre, usada y memoria disponible

TOP se puede modificar para sacar x valores. Por ejemplo, con top -d cambia el tiempo de actualización, top -u muestra los procesos de x usuario y top -p muestra un proceso que tenga ese PID

#### Ejecución en primer y segundo plano

Cuando un usuario ejecuta un comando, tiene que esperar el control del intérprete de comandos. Eso se denomina ejecución en primer plano

Cuando un usuario puede ejecutar una tarea sin esperar a su terminación para que el Shell le devuelva el prompt, ese proceso se encuentra en segundo plano. Para ejecutar comandos en segundo plano, se hace añadiendo un ampersand (&) al final de las órdenes

IDENTIFICACIÓN DE TAREAS: ***jobs***

Paso de procesos de primer a segundo plano: (ctrl + z para pararlo) y bg [%][tarea]

Pasar procesos de segundo a primer plano: fg [%] [tarea]

```
erikat@Ubuntu:~/Escritorio$ yes > /dev/null
^Z
[1]+  Detenido          yes > /dev/null
erikat@Ubuntu:~/Escritorio$ jobs
[1]+  Detenido          yes > /dev/null
erikat@Ubuntu:~/Escritorio$ bg %1
[1]+ yes > /dev/null &
erikat@Ubuntu:~/Escritorio$ jobs
[1]+  Ejecutando        yes > /dev/null &
erikat@Ubuntu:~/Escritorio$ fg %1
yes > /dev/null
^Z
[1]+  Detenido          yes > /dev/null
```

### Prioridad de órdenes

La prioridad de los procesos se puede alterar mediante el índice nice:

- Oscila entre -20 (máxima prioridad) y 19 (menor prioridad)
- Puede ser modificada por el propietario del proceso o el superusuario
- Sigue el modelo de Robbins junto a FIFO (First-In-First-Out)
- Podemos lanzar procesos modificando su prioridad con nice
- Podemos modificar la prioridad de un proceso con renice
  - o El superusuario es el único que puede dar un valor negativo al nice de un proceso
- La prioridad nativa de un proceso es NICE
- En **ps -l** aparecen las columnas **PRI** y **NICE** (por defecto en 80 y 0 respectivamente)

Iniciar procesos modificados: **nice**

Sintaxis: nice [-[n]] {[+|-} num\_nice orden]

[-[n]] es en un principio de obligado cumplimiento que esté (salvo la n)  
+ o - servirá para marcar la prioridad, si es positiva o negativa

Modificar procesos ya iniciados: **renice**

Sintaxis: renice prioridad [[-p] PID's] [[-u] usuarios]

-p sirve para elegir el PID del proceso

-u sirve para elegir el UID del usuario para modificar sus NICE

### Envío de señales:

Los procesos pueden recibir señales mediante la orden **kill**

Sintaxis: kill -señal PID

Señales:

- o 2 (SIGINT): Interrumpir proceso (CTRL + C)
- o 9 (SIGKILL): Matar proceso definitivamente de golpe, sin copias de seguridad ni nada
- o 15 (SIGTERM): Matar proceso "suavemente"
- o 18 (SIGCONT): Continuar ejecución de un proceso (CTRL+Z pausado)
- o 19 (SIGSTOP): Pausar la ejecución (CTRL + Z en ejecución)

## Gestión de procesos por Interfaz gráfica en Windows

Se usa el administrador de tareas para administrar los procesos.

En la pestaña “procesos” se pueden ver todos los procesos en ejecución agrupados por aplicaciones (en primer plano), procesos en segundo plano, y procesos de Windows (Servicios y otras aplicaciones de Windows trabajando en segundo plano)

Si se hace clic derecho sobre las aplicaciones, se pueden ver más detalles de esta, además de tener la opción de finalizar la tarea

En la pestaña “detalles”, se puede ver de mejor forma los procesos corriendo, viendo los usuarios que lo están ejecutando, y deja cambiar opciones como el orden de prioridad.

## Automatización de tareas de Linux

Las tareas se pueden planificar para un momento puntual, siendo un lanzamiento recurrente o un momento puntual. Las tareas planificadas para un momento puntual no implican la repetición de ésta, mientras que el lanzamiento recurrente sí que implica que se repita.

### Ejecución de tareas para un momento puntual: at

Sintaxis: at hora [fecha] [-f fichero]

Necesitamos instalarlos con el **apt install**.

Las ejecuciones de tareas en at se realizan en el prompt PS2

### Ejecución de tareas para un momento puntual más adecuado: batch

Sintaxis: batch < tareas

### Ejecución de tareas periódicas: cron

Sintaxis: cron [0..59 (minutos)] [0..23 (hora)] [1..31 (día)] [1..12(mes)] [0..6 (día de semana)] orden

Las ordenes periodicas se guardan en /etc/crontab, un archivo que no puede editar nadie más que el superusuario. Si un campo se ignora, se especifica con un asterisco. Se pueden especificar listas de números mediante comas, y los

Resumen:

(\*): Cualquier valor (ignora el valor)

(, **sin espacios en blanco**): Listas

(-): Rango de valores mínimo y máximo

(v1/v2), valor de inicio/valor incremental.

- El primer campo son los minutos (de 0 a 59)
- El segundo campo son las horas (de 0 a 23)
- El tercer campo son los días (de 1 a 31)
- El cuarto campo son los meses (de 1 a 12)
- El quinto campo son los días de la semana (de 0 (domingo) a 6 (sábado))

### EJEMPLOS:

- Ejecuta orden en el minuto 3 de cada hora y cada día todos los meses para todos los días de la semana:
  - 03 \* \* \* \* \* orden
- Ejecuta orden cada hora en punto, de todos los días del mes que sean domingo:
  - 00 \* \* \* \* 0 orden
- Ejecuta orden cada 3 minutos empezando por el minuto 3 de las horas 2,5,8, etc del día 4 de abril y que sea jueves
  - 3/3 2/3 4 4 4 orden
- Ejecuta orden a las 18:30 horas del día 20 de cada mes de enero a Junio:
  - 30 18 20 1-6 \* orden
- Ejecuta orden a las 06:45 horas de los días impares de cada mes, siempre que sean lunes miércoles o viernes
  - 45 06 1/2 \* 1,3,5 orden
- Ejecuta orden cada 15 minutos de 10 a 14 horas todos los sábados
  - \*/15 10-14 \* \* 6 orden
- Ejecuta orden a las 08:10, 08:30 y 08:45 cada dos días y cada mes
  - 10,30,45 08 \*/2 \* \* orden

## Monitorización y Gestión del Sistema. Evaluación de Prestaciones

### Linux

Un administrador de sistema tiene entre sus labores el tener herramientas para afrontar problemas futuros, y mejorar el sistema en el que se está trabajando. Tanto el sistema como programas de terceros va a dar información muy útil.

Algunos comandos son ps, top, uptime (numero de usuarios, tiempo de uso, carga... (la primera línea del top)), free (obtiene información del espacio de la memoria física y virtual), vmstat (muestra información de la memoria virtual, RAM, procesador, intercambios entre RAM y disco...), df(muestra el uso de las unidades de almacenamiento del sistema), w(who con mas cosas, muestra quién está conectado y qué está haciendo)

### Windows:

A través del administrador de tareas, en la sección de Rendimiento, nos muestra el uso de la CPU, la cantidad de procesos y subprocesos, el uso de discos, el estado de la RAM, el tráfico de los adaptadores de red...

En la pestaña de Usuarios se podrá ver cuánto está actualmente utilizando cada usuario que se conecta a la red

## Aplicaciones para el mantenimiento

Se pueden clasificar en varios grupos:

- Aplicaciones de actualización y control de driver:
- Aplicaciones de Sinronización, copias de seguridad e imágenes del sistema
- Antivirus: Programas contra malware como Aira, AVG, AVAST
- Optimización del sistema: CCleaner (Windows) o Stacer (Linux) para poder optimizar las funciones como el almacenamiento o la memoria