

# UD3 – LINUX – ADMINISTRACIÓ I CONFIGURACIÓ-III

1º DAW - CFGS

Prof. Manuel Enguidanos  
*[menguidanos@fpmislata.com](mailto:menguidanos@fpmislata.com)*

## 3.4. PERMISOS



## 3.4. Permisos

Todos los archivos pertenecen a un usuario y a un grupo. Por defecto, cuando un usuario crea un fichero se le añade el usuario y el grupo primario del usuario. Para ver los permisos de un archivo se puede recurrir al comando `ls` con la opción `-l` o también al comando `stat` (Figura 3.26).



```
usuario@Equipo: ~  
usuario@Equipo:~$ stat fichero.txt  
  Fichero: fichero.txt  
  Tamaño: 33885229      Bloques: 66184      Bloque E/S: 4096  fichero regular  
Dispositivo: 803h/2051d Nodo-l: 394038      Enlaces: 1  
Acceso: (0664/-rw-rw-r--)  Uid: ( 1000/ usuario)  Gid: ( 1000/ usuario)  
Acceso: 2022-01-18 13:03:47.512947678 +0100  
Modificación: 2022-01-19 01:44:39.928045529 +0100  
  Cambio: 2022-01-19 01:44:39.928045529 +0100  
  Creación: 2022-01-18 13:03:47.512947678 +0100  
usuario@Equipo:~$ ls -l fichero.txt  
-rw-rw-r-- 1 usuario usuario 33885229 ene 19 01:44 fichero.txt  
usuario@Equipo:~$
```

Figura 3.26. Permisos e información adicional del fichero fichero.txt.

## 3.4. Permisos

En Linux, para todos los ficheros, existen tres grupos de permisos. Cada fichero debe pertenecer a un grupo y a un usuario. Los grupos de permisos son los pertenecientes al propietario del grupo (**u**), los pertenecientes al grupo (**g**) y los pertenecientes a los que no son ni el propietario ni el grupo (**o**), es decir, el resto de los usuarios.

Los permisos que se pueden añadir son los siguientes: lectura (**w**), escritura (**r**) y ejecución (**x**).

<pre>usuario@Equipo:~\$ ls -l fichero.txt -rwxr-xr-- 1 usuario usuario 574 jul 31 20:58 fichero.txt</pre>			
-	<b><u>rwx</u></b>	<b>r-x</b>	<b>r--</b>
• Tipo de archivo	• Permisos del propietario (u)	• Permisos del grupo (g)	• Permisos de otros (o)

**Figura 3.27.** El primer campo indica el tipo de archivo (- es fichero regular, d es directorio), los permisos del propietario (rwx es lectura, escritura y ejecución), los del grupo (r-x es lectura y ejecución) y los de los demás usuarios (r-- es solo lectura).

## 3.4. Permisos

Tabla 3.6. Permisos que pueden tener los ficheros de Linux y su significado

Permiso	Significado
r	Permiso de lectura.
w	Permiso de escritura.
x	Permiso de ejecución en ficheros; buscar o poder entrar en directorios.
X	Permiso X mayúscula: añade permiso x a los directorios y en el caso de que algún fichero contenga algún permiso de ejecución se le amplía ese permiso de ejecución para todos los usuarios.
s	Permiso <i>setuid bit</i> o <i>setgid bit</i> , que indica que al ejecutar un fichero el proceso pertenecerá al propietario del fichero o al grupo, respectivamente, no al usuario que ejecutó el fichero.
t	Permiso <i>sticky bit</i> (bit pegajoso), que se aplica sobre directorios y permite que un usuario si tiene permisos de escritura pueda modificar los archivos, pero le impide eliminarlos del directorio.

Para los tres primeros números de la derecha, cada número significa:

- 0: ningún permiso.
- 1: permiso de ejecución.
- 2: permiso de escritura.
- 4: permiso de lectura.



## 3.4. Permisos

**Tabla 3.7.** Equivalencia entre los permisos de Linux, su valor binario y su valor en octal

Permisos	Valor binario	Valor numérico
---	000	0
--x	001	1
-w-	010	2
-wx	011	3
r--	100	4
r-x	101	5
rw-	110	6
rwX	111	7

- $X = 1$
- $W = 2$
- $R = 4$

Para los permisos especiales (el cuarto dígito de la derecha, si existe), los números indican lo siguiente:

- 1: *sticky bit*.
- 2: *setgid bit*.
- 4: *setuid bit*.

## 3.4. Permisos

### chmod

Establece los permisos de un archivo o directorio.

Sintaxis:

```
chmod [opciones] permisos[,permisos]... archivos...
```

```
chmod [opciones] número_octal archivos...
```

Los permisos se pueden establecer mediante un modo simbólico que indique qué permisos y a quién se le asignan, o bien de forma numérica, como un número octal (de 0 a 7) de hasta 4 dígitos.

En el modo simbólico cada permiso se establecerá de la siguiente forma (se pueden establecer varios permisos separados por comas):

```
chmod [ugoa] [[-+=] [rwxXst]] archivos...
```

En este caso, la letra indica si el permiso se le va a modificar al propietario (**u**), a los del grupo (**g**), a los que no sean propietarios ni del grupo (los otros, **o**) y a todos (*all*, **a**). El permiso se podrá añadir (+), quitar (-) o establecer (=). Los permisos pueden ser los indicados en la Tabla 3.6.



## 3.4. Permisos

### chmod

Establece los permisos de un archivo o directorio.

Sintaxis:

```
chmod [opciones] permisos[,permisos]... archivos...
```

```
chmod [opciones] número_octal archivos...
```



Las opciones podemos indicarla o no, según queramos. Opciones típicas son:

- -R para que mire también en los subdirectorios de la ruta.
- -v para que muestre cada fichero procesado
- -c es como -v, pero sólo avisa de los ficheros que modifica sus permisos



## 3.4. Permisos

### Otro modo de trabajo con chmod

Luego también se pueden asignar permisos de otra manera, utilizando otra posible sintaxis de chmod, que tal vez resulte más útil si no queremos tratar con los valores en octal.

***chmod [opciones] modo[,modo] fichero***

Para ello tenemos que tener claros los distintos grupos de usuarios:

- u: usuario dueño del fichero
- g: grupo de usuarios del dueño del fichero
- o: todos los otros usuarios
- a: todos los tipos de usuario (dueño, grupo y otros)

También hay que saber la letra que abrevia cada tipo de permiso:

- r: se refiere a los permisos de lectura
- w: se refiere a los permisos de escritura
- x: se refiere a los permisos de ejecución

## 3.4. Permisos

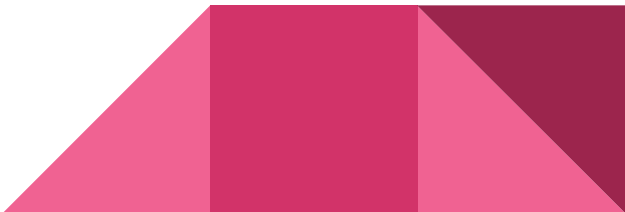
```
chmod 700 nombre_archivo
```

```
chmod u=rwx nombre_archivo
```

```
chmod -R 755 directorio
```

```
chmod 777 nombre_archivo
```

```
chmod +x /direccion/del/archivo
```



## 3.4. Permisos

**umask** (*user mask*)

Muestra o modifica los permisos predeterminados que tendrán los archivos y directorios que se creen.

Sintaxis:

```
umask [máscara_de_permisos]
```

La máscara de permisos funcionará de forma diferente si se crea un directorio o un fichero después de establecerla. Los permisos base para los ficheros son 666, y para los directorios son 777; por tanto, una vez que se ha establecido **umask**, para ver qué permisos tendrá el fichero o el directorio creado se puede calcular de la siguiente forma:

- Para los directorios puede restarse **umask** a los permisos base.
- Para los archivos, habría que realizar la operación **and** al valor negado (**not**) de **umask**.

El motivo del valor de estos permisos base es evitar que en los ficheros se active el permiso de ejecución por defecto. El valor predeterminado de **umask** está definido en los últimos sistemas como 0002 para los usuarios no *root* y como 0022 para el usuario *root*, en el fichero `/etc/login.defs`.

El comando **umask** también se puede utilizar en modo simbólico, al igual que el comando **chmod**.



# **Realitzar Activitats Resoltes**





## Actividad resuelta 3.17

Comprueba los permisos de los siguientes ficheros y carpetas: `/etc/shadow`, `/root`, `/`, `.profile`, `.bashrc`, `.bash_logout`. Indica cómo se indicarían esos permisos de forma numérica.

### Solución

Recuerda que para ver los permisos del directorio debes utilizar la opción `-d`; si no, te mostraría el contenido del mismo, es decir, los archivos y las carpetas que contiene y sus permisos.

<code>ls -l /etc/shadow</code>	Permisos: <code>rw-r-----</code>	Forma numérica: 640
<code>ls -ld /root</code>	Permisos: <code>rw-x-----</code>	Forma numérica: 700
<code>ls -ld /</code>	Permisos: <code>rw-xr-xr-x</code>	Forma numérica: 755

Los siguientes archivos son los *scripts* de inicio de sesión (los dos primeros) y de fin de sesión (el último) de cada usuario, es decir, los *scripts* que se ejecutan cada vez que el usuario inicia o cierra la sesión en el sistema. Cada usuario tiene los archivos en su directorio personal, así que para asegurarte de estar en él, puedes escribir:

<code>cd</code>	Sin opciones te lleva a tu directorio personal.	
<code>ls -l .profile</code>	Permisos: <code>rw-r-----</code>	Forma numérica: 640
<code>ls -l .bashrc</code>	Permisos: <code>rw-r-----</code>	Forma numérica: 640
<code>ls -l .bash_logout</code>	Permisos: <code>rw-r-----</code>	Forma numérica: 640



### Actividad resuelta 3.18

Establece la máscara predeterminada para que cuando crees un fichero tú tengas todos los permisos activos salvo los de ejecución, y los de tu grupo y los otros ninguno. Cámbialo para que cuando crees una carpeta tú tengas todos los permisos, los de tu grupo tengan el de lectura y ejecución, y los otros ninguno.

#### Solución

`umask 177` (también valdría `umask 077` ya que en ficheros el de ejecución no se activa).  
`umask 027`



### Actividad resuelta 3.19

Crea una carpeta en tu sistema operativo anfitrión y compártela desde VirtualBox.

#### Solución

Crea una carpeta y llámala compartida en tu sistema operativo anfitrión. Si es Windows, con el botón secundario del ratón sobre un sitio libre en el escritorio selecciona **Nuevo → Carpeta**; cuando tengas creada la carpeta renómbrala como **compartida**.

En VirtualBox, ve a **Dispositivos → Carpetas compartidas → Preferencias de las carpetas compartidas**. En **Carpetas de la máquina**, pulsa sobre el icono **Añadir nueva carpeta compartida** (símbolo + verde). En la ventana que se abre, en **Ruta carpeta**, pincha la flecha hacia abajo, selecciona **Otro...** y busca la ruta de la carpeta que acabas de crear. En **Nombre de la carpeta** pon el nombre que quieras que tenga la carpeta en Linux, que no necesariamente tiene que coincidir. Por último, marca **Automontar** y **Hacer permanente** para tenerla siempre. Para el punto de montaje deja que sea el sistema el que lo elija (Figura 3.28).



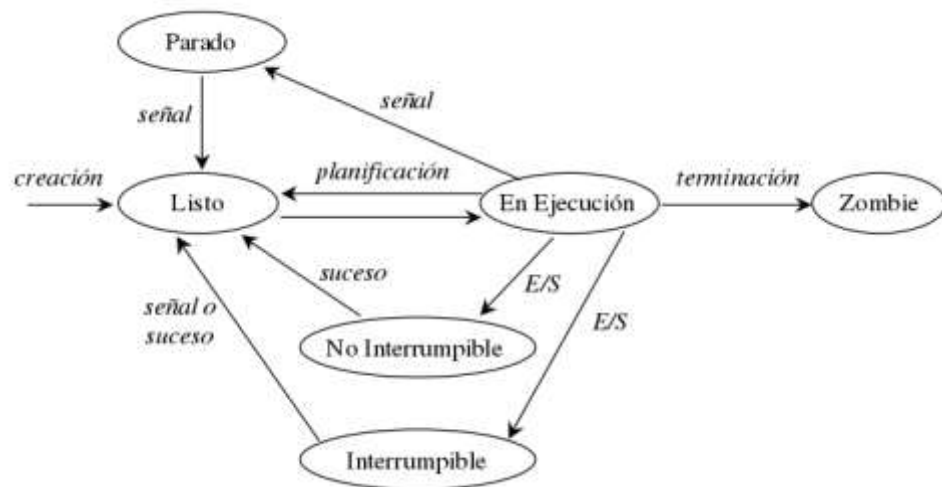
## **3.5. GESTIÓN DE PROCESOS Y SERVICIOS**



## 3.5. Gestión de procesos y servicios

Un proceso, o tarea, es un programa que está en ejecución. Cada vez que se ejecuta un programa se crea uno o varios procesos. Puede haber más de un proceso o instancias de un mismo programa, cada uno con sus propiedades y su contexto que lo diferencian de otro, como el usuario que lo ha lanzado, que es el dueño del proceso, los permisos del proceso, si ha sido llamado por otro proceso, al que se le llama proceso padre, etcétera.

Además, cada proceso se identifica en el sistema por un número entero llamado **PID** (**P**rocess **I**D, identificador del proceso). No puede haber dos procesos en un sistema con el mismo PID.



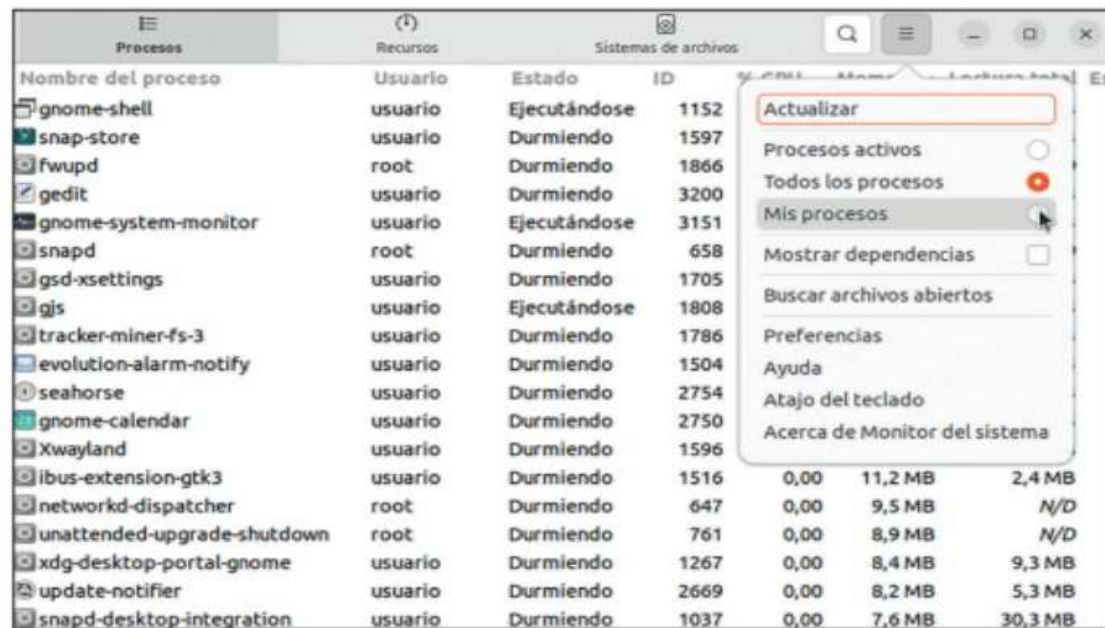
## 3.5.1. GESTIÓN DE PROCESOS



## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Monitor del sistema



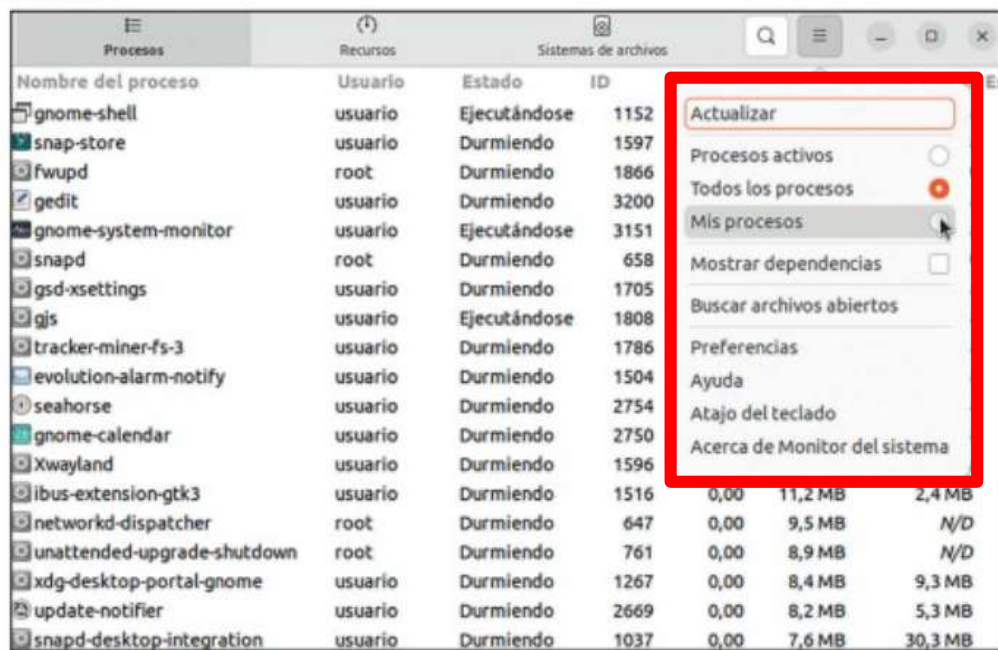
Nombre del proceso	Usuario	Estado	ID			
gnome-shell	usuario	Ejecutándose	1152			
snap-store	usuario	Durmiendo	1597			
fwupd	root	Durmiendo	1866			
gedit	usuario	Durmiendo	3200			
gnome-system-monitor	usuario	Ejecutándose	3151			
snapt	root	Durmiendo	658			
gsd-xsettings	usuario	Durmiendo	1705			
qjs	usuario	Ejecutándose	1808			
tracker-miner-fs-3	usuario	Durmiendo	1786			
evolution-alarm-notify	usuario	Durmiendo	1504			
seahorse	usuario	Durmiendo	2754			
gnome-calendar	usuario	Durmiendo	2750			
Xwayland	usuario	Durmiendo	1596			
ibus-extension-gtk3	usuario	Durmiendo	1516	0,00	11,2 MB	2,4 MB
networkd-dispatcher	root	Durmiendo	647	0,00	9,5 MB	N/D
unattended-upgrade-shutdown	root	Durmiendo	761	0,00	8,9 MB	N/D
xdg-desktop-portal-gnome	usuario	Durmiendo	1267	0,00	8,4 MB	9,3 MB
update-notifier	usuario	Durmiendo	2669	0,00	8,2 MB	5,3 MB
snapt-desktop-integration	usuario	Durmiendo	1037	0,00	7,6 MB	30,3 MB

Figura 3.30. Estado de los procesos con el programa que los generó, el usuario, su PID, etcétera.

## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Monitor del sistema



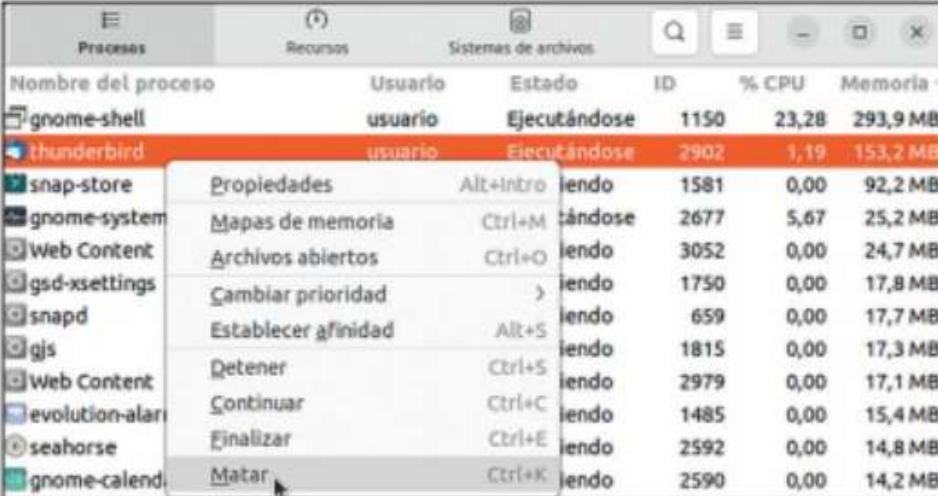
Nombre del proceso	Usuario	Estado	ID	Memoria	Memoria compartida	Memoria caché
gnome-shell	usuario	Ejecutándose	1152			
snap-store	usuario	Durmiendo	1597			
fwupd	root	Durmiendo	1866			
gedit	usuario	Durmiendo	3200			
gnome-system-monitor	usuario	Ejecutándose	3151			
snapd	root	Durmiendo	658			
gsd-xsettings	usuario	Durmiendo	1705			
gjs	usuario	Ejecutándose	1808			
tracker-miner-fs-3	usuario	Durmiendo	1786			
evolution-alarm-notify	usuario	Durmiendo	1504			
seahorse	usuario	Durmiendo	2754			
gnome-calendar	usuario	Durmiendo	2750			
Xwayland	usuario	Durmiendo	1596			
ibus-extension-gtk3	usuario	Durmiendo	1516	0,00	11,2 MB	2,4 MB
networkd-dispatcher	root	Durmiendo	647	0,00	9,5 MB	N/D
unattended-upgrade-shutdown	root	Durmiendo	761	0,00	8,9 MB	N/D
xdg-desktop-portal-gnome	usuario	Durmiendo	1267	0,00	8,4 MB	9,3 MB
update-notifier	usuario	Durmiendo	2669	0,00	8,2 MB	5,3 MB
snapd-desktop-integration	usuario	Durmiendo	1037	0,00	7,6 MB	30,3 MB

Figura 3.30. Estado de los procesos con el programa que los generó, el usuario, su PID, etcétera.

## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Monitor del sistema



Nombre del proceso	Usuario	Estado	ID	% CPU	Memoria
gnome-shell	usuario	Ejecutándose	1150	23,28	293,9 MB
thunderbird	usuario	Ejecutándose	2902	1,19	153,2 MB
gnome-system					
Web Content					
gsd-xsettings					
snapsd					
gjs					
Web Content					
evolution-alarm					
seahorse					
gnome-calend					

Propiedades

Alt+Intro

Mapas de memoria

Ctrl+M

Archivos abiertos

Ctrl+O

Cambiar prioridad

>

Establecer afinidad

Alt+S

Detener

Ctrl+S

Continuar

Ctrl+C

Finalizar

Ctrl+E

Matar

Ctrl+K

Figura 3.31. Visualización de las propiedades de un proceso, cambio de su prioridad, modificación de su estado, etcétera.

## ■ 3.5. Gestión de procesos y servicios

### ■■ 3.5.1. Gestión de procesos

#### ■■■ Comandos para gestionar procesos

`ps`

Informa del estado de los procesos que se están ejecutando en el sistema.

Sintaxis:

`ps [opciones]...`

Opciones:

<code>-A, -e</code>	Muestra todos los procesos.
<code>-l</code>	Formato largo.
<code>-Y</code>	Muestra la cantidad de memoria física.
<code>a</code>	Muestra todos los procesos con su terminal.
<code>u</code>	Utiliza un formato adecuado para ser visualizado.
<code>x</code>	Muestra todos los procesos aunque no tengan asignada una terminal.



# ■ 3.5. Gestión de procesos y servicios

## ■ 3.5.1. Gestión de procesos

### ■ Comandos para gestionar procesos

Ejemplos:

```
$ ps
```

PID	TTY	TIME	CMD
10966	pts/1	00:00:00	bash
11017	pts/1	00:00:00	ps

Por defecto muestra:

- **PID** : Id del Proceso.
- **TTY** : Terminal.
- **TIME** : Tiempo de ejecución.
- **CMD** : Comando.

## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

```
$ ps u
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
edu	4288	0.0	0.0	201292	5548	tty2	Ss1+	08:21	0:00	/usr/lib/gdm3/gdm-x-se....
edu	4290	2.3	1.6	650560	129248	tty2	Sl+	08:21	6:04	/usr/lib/xorg/Xorg vt2 -display..
edu	4315	0.0	0.3	469756	30124	tty2	Sl+	08:21	0:00	mate-session
edu	4412	0.0	0.4	1075928	33944	tty2	Sl+	08:21	0:04	/usr/bin/mate-settings-daemon
edu	4416	0.0	0.4	492204	33180	tty2	Sl+	08:21	0:05	marco
edu	4420	0.0	0.9	699116	76368	tty2	Sl+	08:21	0:07	mate-panel

#### • **u** : Orientado al usuario:

- **USER**
- **% CPU** : uso de procesador.
- **% MEM** : uso de memoria.
- **VSZ** : Memoria virtual.
- **RSS** : Memoria física.
- **STAT** : Estado.
- **START** : Iniciado.

#### Significados de los estados **STAT** :

- **S** : Esperando (Sleep).
- **R** : Ejecutando (Running)
- **D** : Esperando entrada/salida.
- **T** : Pausa.
- **Z** : No responde (Zombie)

#### Información adicional de los **STAT**:

- **s** : Proceso padre
- **I** : Proceso con hilos.
- **+** : En primer plano.

## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

```
top 12:52:28 up 2 min, 2 users, load average: 1.60, 0.98, 0.39
Tasks: 143 total, 1 running, 141 sleeping, 0 stopped, 1 zombie
Cpu(s): 1.3%us, 5.6%sy, 0.0%ni, 89.1%id, 4.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 766412k total, 405660k used, 360752k free, 21480k buffers
Swap: 784380k total, 0k used, 784380k free, 172356k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
902	root	20	0	52160	29m	7008	S	2.7	3.9	0:05.50	Xorg
1349	carlos	20	0	83364	12m	10m	S	2.7	1.7	0:01.11	wnck-applet
1145	carlos	20	0	99.2m	11m	8656	S	0.7	1.5	0:00.81	gnome-settings-
21	root	20	0	0	0	0	S	0.3	0.0	0:00.85	kworker/0:1
973	root	20	0	6784	580	280	S	0.3	0.1	0:00.26	VBoxService
1163	carlos	20	0	74132	12m	9516	S	0.3	1.6	0:00.69	metacity
1179	carlos	20	0	113m	16m	12m	S	0.3	2.2	0:01.01	gnome-panel
1453	carlos	20	0	93296	14m	10m	S	0.3	1.9	0:00.93	gnome-terminal
1511	carlos	20	0	2760	1328	948	R	0.3	0.2	0:00.24	top
1	root	20	0	3052	1828	1252	S	0.0	0.2	0:01.33	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.02	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.16	ksoftirqd/0
4	root	20	0	0	0	0	S	0.0	0.0	0:00.10	kworker/0:0
5	root	20	0	0	0	0	S	0.0	0.0	0:00.46	kworker/u:0
6	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
7	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	cpuset
8	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	khelper

## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

manu@manu:~\$ **htop** sudo apt install htop



The screenshot shows the htop interface with system statistics at the top and a table of running processes below. The 'htop' process is highlighted in green.

System Statistics:

- 0% CPU, 0.7% load, Tasks: 119, 311 thr: 1 running
- 1% CPU, 0.0% load, Load average: 0.12 0.13 0.24
- Mem: 1.20G/3.82G, Uptime: 00:29:58
- Swp: 0K/3.24G

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
835	mysql	20	0	1741M	385M	35712	S	1.4	9.9	0:13.58	/usr/sbin/mysql
1084	mysql	20	0	1741M	385M	35712	S	0.7	9.9	0:07.13	/usr/sbin/mysql
1925	manu	20	0	3965M	374M	136M	S	0.7	9.6	0:27.27	/usr/bin/gnome-
1936	manu	20	0	3965M	374M	136M	S	0.7	9.6	0:06.09	/usr/bin/gnome-
4133	manu	20	0	13712	4608	3584	R	0.7	0.1	0:00.04	htop
1	root	20	0	162M	11844	8388	S	0.0	0.3	0:01.58	/sbin/init spla
185	root	19	-1	64768	17424	15888	S	0.0	0.4	0:00.57	/lib/systemd/sy
213	root	20	0	26784	6912	4608	S	0.0	0.2	0:00.20	/lib/systemd/sy
415	systemd-o	20	0	14824	6784	6016	S	0.0	0.2	0:01.30	/lib/systemd/sy
422	systemd-r	20	0	25924	14208	9600	S	0.0	0.4	0:00.26	/lib/systemd/sy
424	systemd-t	20	0	89376	7552	6656	S	0.0	0.2	0:00.09	/lib/systemd/sy
464	systemd-t	20	0	89376	7552	6656	S	0.0	0.2	0:00.00	/lib/systemd/sy
577	root	20	0	237M	8024	7128	S	0.0	0.2	0:00.16	/usr/libexec/ac
578	root	20	0	2812	1664	1664	S	0.0	0.0	0:00.00	/usr/sbin/acpid
581	avahi	20	0	7628	4096	3712	S	0.0	0.1	0:00.12	avahi-daemon: r
582	root	20	0	12108	2816	2688	S	0.0	0.1	0:00.01	/usr/sbin/cron

Footer: F1 Help F2 Setup F3 Search F4 Filter F5 Free F6 SortBy F7 Nice F8 Nice + F9 Kill F10 Quit

## ■ 3.5. Gestión de procesos y servicios

### ■ 3.5.1. Gestión de procesos

#### ■ Comandos para gestionar procesos

```
pstree
```

Muestra un árbol con los procesos del sistema.

```
nohup
```

```
&
```

Estos comandos se utilizan para poder ejecutar procesos en segundo plano y así dejar libre la terminal. Se utilizan para procesos que no necesitan interacción con el usuario. Se pueden utilizar de alguna de las siguientes formas:

Sintaxis:

```
nohup comando
```

```
comando &
```



## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

```
himanshu@ansh:~$ pstree
systemd--ModemManager--({gdbus}
                        {gmain})
--NetworkManager--dhclient
                  dnsmasq
                  ({gdbus}
                   {gmain})
--accounts-daemon--({gdbus}
                   {gmain})
--acpid
--agetty
--avahi-daemon--avahi-daemon
--colord--({gdbus}
          {gmain})
--cron
--cups-browsed--({gdbus}
                {gmain})
--cupsd--dbus
--dbus-daemon
--fwupd--3*[{GUsbEventThread}]
        ({fwupd}
         {gdbus}
         {gmain})
--geoclue--({gdbus}
           {gmain})
--gnome-keyring-d--({gdbus}
                   {gmain}
                   {timer})
```

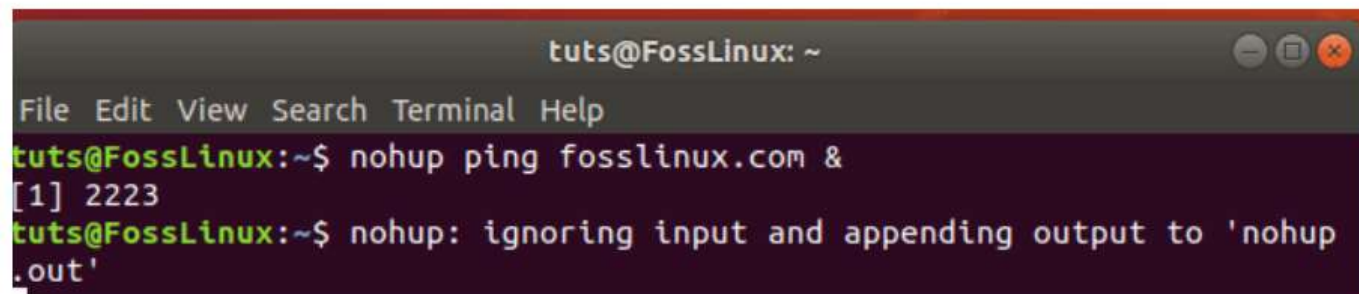
## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

```
$ nohup ping fosslinux.com
```

Ejemplo:

A terminal window titled 'tuts@FossLinux: ~' with standard window controls. The menu bar includes 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal shows the command 'tuts@FossLinux:~\$ nohup ping fosslinux.com &' being entered. The first line of output is '[1] 2223'. The second line of output is 'tuts@FossLinux:~\$ nohup: ignoring input and appending output to 'nohup.out''.

```
tuts@FossLinux: ~  
File Edit View Search Terminal Help  
tuts@FossLinux:~$ nohup ping fosslinux.com &  
[1] 2223  
tuts@FossLinux:~$ nohup: ignoring input and appending output to 'nohup.out'
```

ejecutando nohup en segundo plano



## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

##### nice

Ejecuta procesos con una prioridad más baja o muestra la prioridad predeterminada.

Sintaxis:

```
nice [-n <prioridad>] [comando]
```

**Prioridad** es un valor comprendido entre 20 (menos prioridad) y -19 (mayor prioridad); con la opción **-n** se puede establecer el valor.

##### renice

Modifica la prioridad de un proceso en ejecución.

Sintaxis:

```
renice [opciones] <identificador>
```

Opciones:

```
-p, --p  
-n, --priority  
-u, --user
```

Interpreta qué identificador es el PID del proceso.

*Identificador* es la prioridad nueva del proceso.

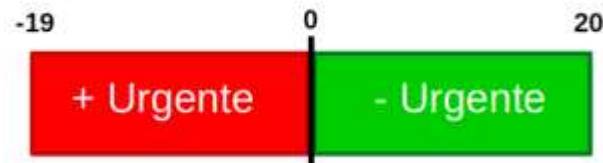
*Identificador* son los nombres o UID de los usuarios.

## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

#### Prioridad de los procesos



sudo nice -n -5 find El valor de la prioridad del proceso find decrece en 5, aumentando su prioridad.

nice -n +5 find El valor de la prioridad del proceso find aumenta en 5, disminuye su prioridad.

Supongamos que tiene que corriendo un procesos con PID 785, y el mismo realiza una operación científica compleja, y mientras el proceso está trabajando Usted desea jugar un juego. Entonces, teclea:

```
$ renice +15 785
```

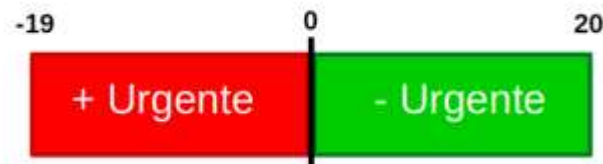
En este caso, su proceso probablemente trabajará un poquito más. Sin embargo, no evitará que otros procesos utilicen más tiempo de CPU.

## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

Prioridad de los procesos



Si Usted es el administrador del sistema y nota que algún usuario está corriendo muchos procesos que utilizan muchos recursos del sistema, puede cambiar la prioridad de los procesos de dicho usuario con un único comando:

```
# renice +20 -u peter
```

Luego de esto, todos los procesos de peter tendrán la prioridad menor y no obstruirán procesos de otros usuarios.

## ■ 3.5. Gestión de procesos y servicios

### ■■ 3.5.1. Gestión de procesos

#### ■■■ Comandos para gestionar procesos

`jobs`

Muestra los trabajos activos asociados a una terminal.

Sintaxis:

`jobs [opciones]`

Con la opción `-l` muestra además el PID de cada proceso.

`fg`

Envía a ejecutarse a primer plano un trabajo que estuviese en segundo plano.

Sintaxis:

`fg [id_trabajo]`

Sin operando selecciona el primer trabajo mostrado con el comando `jobs`. Si se quiere seleccionar otro, se escribirá: `%2` para indicar el segundo, `%3` para el tercero, y así sucesivamente.

## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

<b>-l</b>	Muestra el PID además de la información anterior.
<b>-p</b>	Solo muestra el PID de los trabajos
<b>-r</b>	Solo muestra los trabajos que están en estado de ejecución.
<b>-s</b>	Solo muestra los trabajos que están en estado de detenido.

```
$ jobs -l
[1] 4229 Ejecutando      gedit &
[2]- 4239 Parado         sleep 50
[3]+ 4241 Parado         find / -name Descargas 2> /dev/null

$ jobs -p
4229
4239
4241

$ jobs -r
[1] Ejecutando      gedit &

$ jobs -s
[2]- Detenido       sleep 50
[3]+ Detenido       find / -name Descargas 2> /dev/null
```

## ■ 3.5. Gestión de procesos y servicios

### ■■ 3.5.1. Gestión de procesos

#### ■■■ Comandos para gestionar procesos

Reanuda trabajos suspendidos poniéndolos en *foreground* (primer plano) o trabajos en *background* los pasa a primer plano. Por ejemplo ejecute `du -ah /` luego presione Control-Z (para suspender el proceso), luego al ejecutar el comando jobs vera algo como:

[1]+ Detenido du -ah /

Indicando el número de tarea, el estado y la tarea (recordar que una tarea puede contener más de un comando), con lo cual luego se podrá ejecutar:

fg %1 para poner en primer plano la tarea cuyo número de tarea es 1 (`du -ah /`).

fg %du para poner en primer plano la tarea cuyo nombre empiece por `du`

fg %?ah para poner en primer plano la tarea que contenga la cadena `ah`

fg o fg %% para poner en primer plano la última tarea (la cual es `du -ah /`).

## 3.5. Gestión de procesos y servicios

### 3.5.1. Gestión de procesos

#### Comandos para gestionar procesos

##### bg

Envía a ejecutarse en segundo plano un trabajo que estuviese parado.

Sintaxis:

```
bg [id_trabajo]
```

Sin operando selecciona el primer trabajo mostrado con el comando `jobs`. Si se quiere seleccionar otro, se escribirá: `%2` para indicar el segundo, `%3` para el tercero, y así sucesivamente.

##### kill

Envía una señal a un proceso. Por defecto la señal es matar (*kill*) el proceso. Si se van a matar varios procesos se escribirá el PID de todos ellos separados por un espacio.

Sintaxis:

```
kill [opciones] pid|nombre  
killall
```

Mata todos los procesos utilizando el nombre del programa en lugar de usar el PID.

Sintaxis:

```
killall [opciones...] [nombre_procesos...]  
killall -l, --list
```

Opciones:

<code>-u, --user &lt;usuario&gt;</code>	Procesos asociados al nombre de usuario.
<code>-i, --interactive</code>	Pide confirmación antes de terminar cada proceso.



## ■ 3.5. Gestión de procesos y servicios

### ■ 3.5.1. Gestión de procesos

#### ■ Comandos para gestionar procesos

Si lo que queremos es seguir ejecutandolo pero en background/segundo plano en lugar de «fg» utilizamos «bg»:

```
$ bg %1
```

```
$ ps
```

PID	TTY	TIME	COMMAND
1291	co	0:12	-bin/csh (csh)
3250	p0	0:00	ps
1286	p1	0:05	-bin/csh (csh)
3248	p1	0:05	vi commands

```
$ kill 1291
```

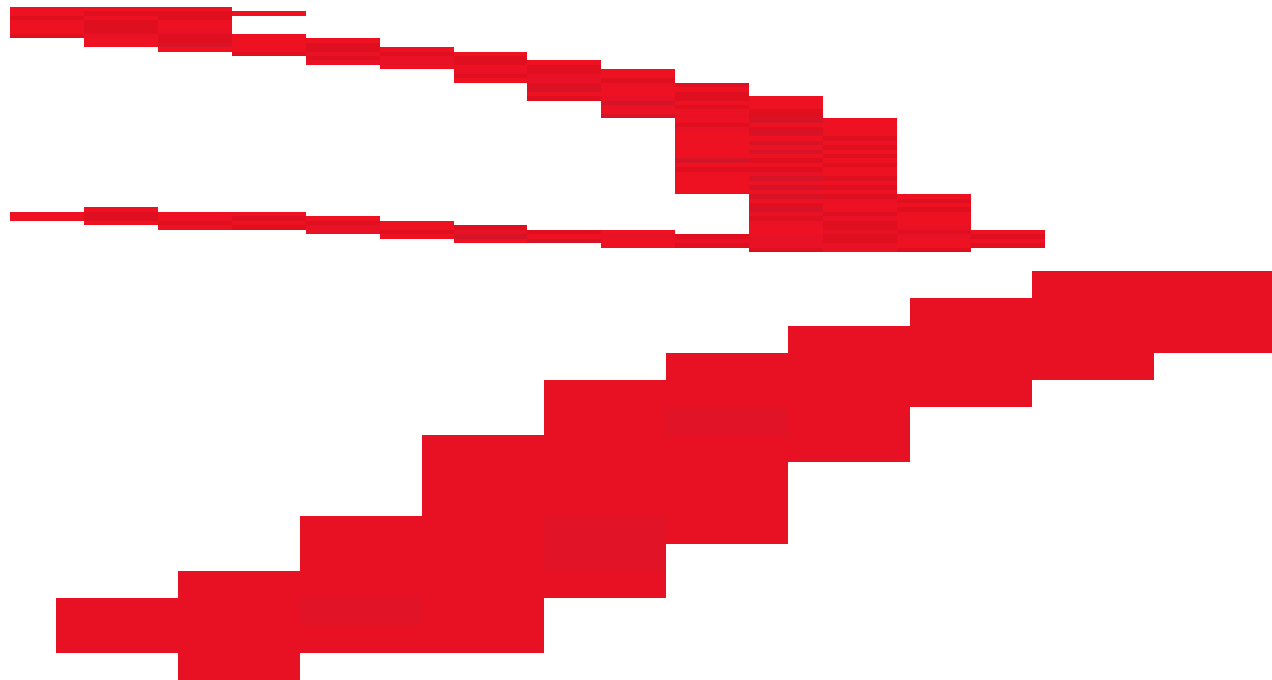
```
[1] Terminated -bin/csh/ (csh)
```

```
$
```

■ 3.5. Gestión de procesos y servicios

■ ■ 3.5.1. Gestión de procesos

■ ■ ■ Comandos para gestionar procesos





# **Realitzar Activitats Resoltes**





`time`

Ejecuta un programa y muestra el tiempo de uso de los recursos del sistema.

Ejemplos:

```
time sleep 3  
time tree /
```

## 3.5.2. GESTIÓN DE SERVICIOS



## 3.5. Gestión de procesos y servicios

Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Se permite la explotación económica y la transformación de esta obra si se menciona la fuente y se otorga crédito al autor y al editor.

© 2019, por el autor.  
Diseño y desarrollo: [www.atacama.cl](http://www.atacama.cl)  
Distribución: [www.atacama.cl](http://www.atacama.cl)  
ISBN: 978-95-0-00-0000-0

### Actividad resuelta 3.20

Ejecuta desde una terminal el programa **sleep**, que genera una pausa por el tiempo indicado, durante 120 segundos en segundo plano. Comprueba los trabajos asociados a esa terminal. Mata el proceso.

#### Solución

```
sleep 120 &  
jobs  
kill %1
```

## 3.5. Gestión de procesos y servicios

Repositorio de datos de la Universidad de Valencia

Repositorio de datos de la Universidad de Valencia - Queda a disposición de los usuarios de la Universidad de Valencia. No se garantiza la exactitud de los datos.

Repositorio de datos de la Universidad de Valencia  
Repositorio de datos de la Universidad de Valencia  
Repositorio de datos de la Universidad de Valencia  
Repositorio de datos de la Universidad de Valencia

### 3.5.2. Gestión de servicios



## 3.5. Gestión de procesos y servicios

Un servicio es un proceso que se está ejecutando en segundo plano a la espera de ser llamado por el usuario o por otro proceso para realizar la tarea para la que ha sido programado. En Linux a los servicios se les denomina *daemons*, que es un acrónimo del inglés **D**isk **a**nd **E**xecution **M**onitor aunque se suele traducir por demonios. El sistema para gestionar y administrar los procesos y los servicios en Linux es **systemd**, que sustituye a **init** en el anterior SysV, aunque ofrece compatibilidad con él. El sistema **systemd** se puede gestionar a través del comando **systemctl**.

## 3.5. Gestión de procesos y servicios

### systemctl

Principal utilidad para gestionar lo que se denomina unidades en **systemd**. Las unidades pueden ser servicios (service), dispositivos (device), objetivos de inicio (target), temporizadores (timer), etcétera.

Sintaxis:

```
systemctl [opciones...] comandos...
```

Opciones:

<b>-t, --type=</b>	Lista separada por comas de las unidades que listar.
<b>-a, --all</b>	Muestra todas las unidades.

Comandos:

<b>systemctl list-units</b>	<b>[patrón...]</b>	Muestra las unidades que están en memoria cuyo nombre coincida con el patrón. Para ver todas las unidades instaladas se utiliza <b>list-units-files</b> .
-----------------------------	--------------------	---

## 3.5. Gestión de procesos y servicios

**Ejemplos de salida del comando systemctl**

Desde la raíz del sistema el comando `systemd` que está instalado a menudo muestra los 3 primeros valores más importantes:

```
# systemctl status --no-pager\nsystemd.service\nsystemd.service\nsystemd.service\nsystemd.service\nsystemd.service
```

**`sudo systemctl start|stop|restart|reload <nombre_servicio>`**

Iniciar, parar, reiniciar o recargar un servicio. Recargar consiste en volver a cargar la configuración del servicio sin necesidad de pararlo.

**`sudo systemctl enable|disable <nombre_servicio>`**

Habilitar o deshabilitar un servicio al arrancar el sistema.

**`systemctl status <nombre_servicio>|<PID>`**

Muestra el estado del servicio. Hay veces que para acceder a cierta información habrá que ejecutarlo con **sudo**. Para salir de la información habrá que pulsar la letra **q**. El estado del servicio puede estar inactivo o desactivado (se muestra en color blanco), en un estado de fallo o error (se muestra en color rojo) o activo, activándose o recargando (se muestra en color verde).

**`systemctl is-active <nombre_servicio>`**

Indica si un servicio está activo o no.

**`<nombre_servicio>`** puede escribirse indicando únicamente el **nombre del servicio** sin necesidad de utilizar **nombre.service** ya que el sistema busca los servicios con ese nombre.

Los servicios pueden estar habilitados (*enabled*), deshabilitados (*disabled*), bloqueados o deshabilitados, y no se pueden habilitar hasta quitarles ese estado (*masked*), etcétera.

Habilitar un servicio con la opción **enable** significa que se iniciará al arrancar el sistema, pero no implica que se inicie a no ser que se utilice el comando con la opción **--now**.

## 3.5. Gestión de procesos y servicios

Requisitos del sistema para el funcionamiento normal

Se requiere un sistema de archivos de tipo ext2, ext3, ext4, o xfs. Se requiere un sistema de archivos de tipo ext2, ext3, ext4, o xfs.

Se requiere un sistema de archivos de tipo ext2, ext3, ext4, o xfs. Se requiere un sistema de archivos de tipo ext2, ext3, ext4, o xfs.

```
$ sudo systemctl start application.service
```

```
$ sudo systemctl stop application.service
```

```
$ sudo systemctl restart application.service
```

```
$ sudo systemctl enable application.service
```

```
$ sudo systemctl disable application.service
```

```
$ systemctl status application.service
```

Output:

```
• nginx.service - A high performance web server and a reverse proxy server
   loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2015-01-27 19:41:23 EST; 22h ago
   Main PID: 495 (nginx)
   CGroup: /system.slice/nginx.service
           └─495 nginx: master process /usr/bin/nginx -g pid /run/nginx.pid; error_log stderr;
             └─496 nginx: worker process

Jan 27 19:41:23 desktop systemd[1]: Starting A high performance web server and a reverse proxy:
Jan 27 19:41:23 desktop systemd[1]: Started A high performance web server and a reverse proxy:
```



# **Realitzar Activitats Resoltes**





Para ver todos los servicios se ejecuta lo siguiente:

```
systemctl -t service list-units --all  
systemctl -t service -a  
systemctl list-unit-files --type=service
```

Para ver solo los que están habilitados se ejecuta este comando:

```
systemctl list-unit-files --state=enabled
```

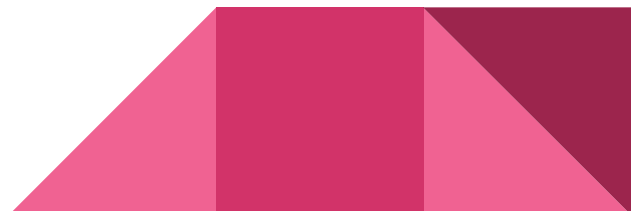
El comando **systemctl** sirve además para configurar algunos aspectos del sistema, como, por ejemplo, indicar que por defecto arranque en modo gráfico:

```
sudo systemctl set-default graphical.target
```

Para arrancar el sistema en modo rescate, apagarlo, reiniciarlo o reiniciar en modo gráfico habrá que escribir, respectivamente:

```
sudo systemctl isolate rescue.target  
sudo systemctl isolate poweroff.target  
sudo systemctl isolate reboot.target  
sudo systemctl isolate graphical.target
```

## 3.6. COPIES DE SEGURETAT









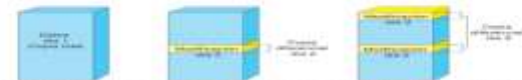
El sistema Backup utiliza estos sistemas de archivos para las copias incrementales, pero estas tienen un inconveniente: si un archivo ha sido eliminado antes que sea copiado, al crear la copia se eliminará el sistema de archivos y quedará vacío.

- **Incremental 0**: **Backup 0** = **Full Backup 0** - **Backup 0** no corresponde con el nivel 0 de Backup, porque el sistema no tiene asociado.
- **Incremental 1**: **Backup 1** = **Full Backup 1** - **Backup 1** no corresponde con el nivel 1 de Backup, porque el sistema no tiene asociado.
- **Incremental 2**: **Backup 2** = **Full Backup 2** - **Backup 2** no corresponde con el nivel 2 de Backup, porque el sistema no tiene asociado.
- **Incremental 3**: **Backup 3** = **Full Backup 3** - **Backup 3** no corresponde con el nivel 3 de Backup, porque el sistema no tiene asociado.
- **Incremental 4**: **Backup 4** = **Full Backup 4** - **Backup 4** no corresponde con el nivel 4 de Backup, porque el sistema no tiene asociado.

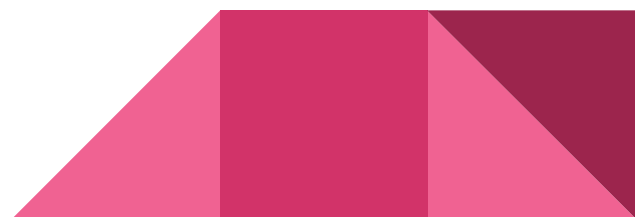
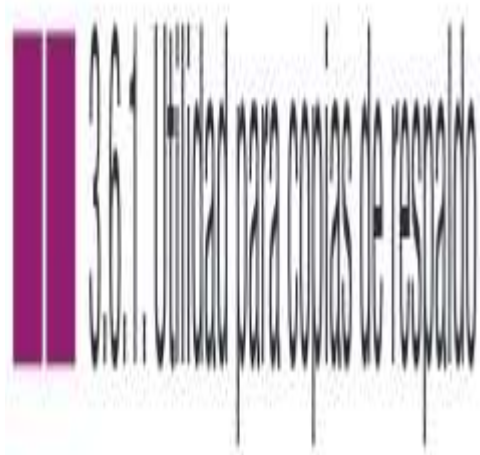
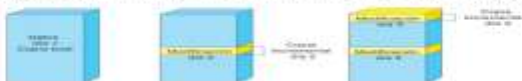
Para poder estar identificados los niveles de seguridad y organizados por nivel, se han creado los siguientes términos: **Full** y **Incremental**.

- **Total o Full**: es una copia que copia todos los archivos del sistema, independientemente de si ya se han copiado antes.
- **Incremental**: es una copia que copia solo los archivos que se han añadido o modificado desde la última copia realizada.
- **Diferencial**: es una copia que copia todos los archivos que se han añadido o modificado desde la última copia realizada.

- En la copia **total**, se realiza una copia completa y exacta de la información original, independientemente de las copias realizadas anteriormente.
- En los **backups diferenciales** cada vez que se realiza una copia de seguridad, se copian todos los archivos que ha-yan sido modificados desde la última copia completa.



- En el caso de los sistemas de copia **incremental**, únicamente se copian los archivos que se hayan añadido o modificado desde la última copia realizada, sea total o incremental.

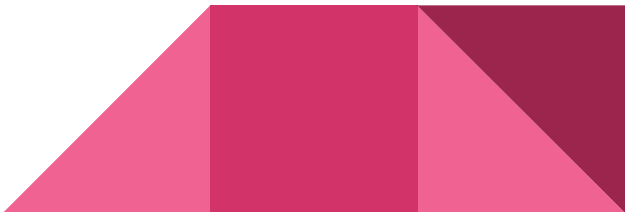
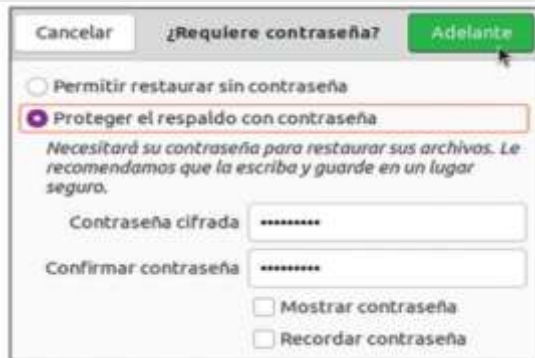





El sistema de archivos de Ubuntu viene instalado por defecto la utilidad Copias de respaldo o Déjà Dup (Figura 3.32).

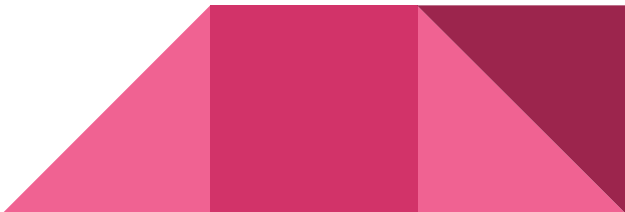
Si la utilidad no viene instalada, se puede instalar con el siguiente comando:

```
sudo apt install deja-dup
```



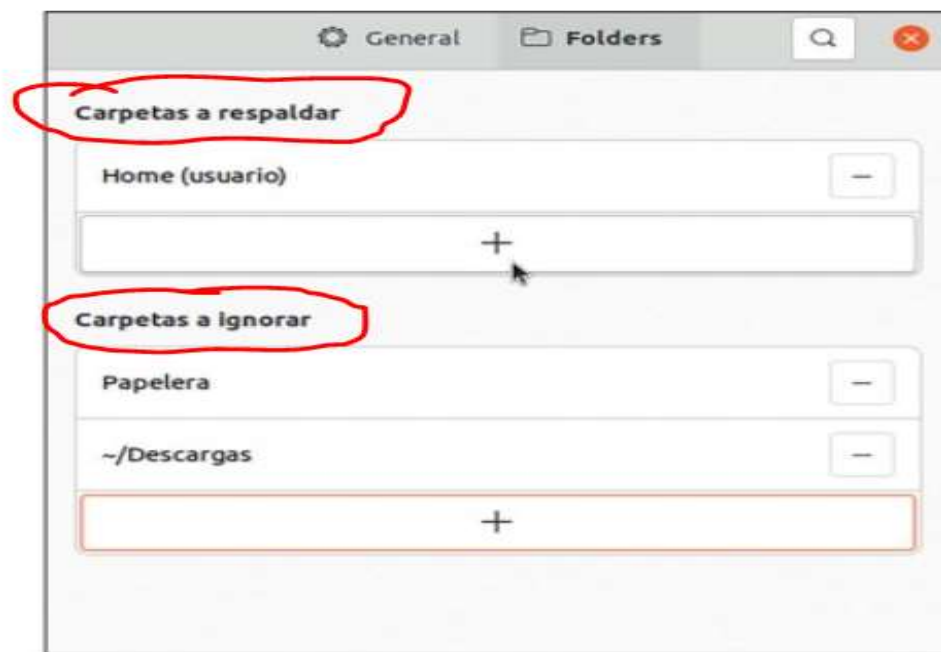


Ubicación del respaldo	<div><div> Google Drive</div><div> Servidor de red</div><div> Carpeta local</div></div>	
Carpeta	<input type="text" value="/tmp"/>	<input type="button" value="Escoger carpeta..."/>









**Figura 3.39.** Elección de las carpetas que se desean respaldar y las que se quieren ignorar.

Die Installationen werden als kleine Dateien (z.B. `install.exe`) erstellt, die auf dem System installiert werden können. Die Installationen sind in der Regel in der `bin`-Ordner des Projekts zu finden. Die Installationen sind in der Regel in der `bin`-Ordner des Projekts zu finden.



## 36.2 Comandos para copias de seguridad





Si deseas una mejor compresión, también puedes usar **.tar.gz**.

Un ejemplo de esto es:

```
tar -cvzf sampleArchive.tar.gz /home/sampleArchive
```

La opción adicional **z** representa la **compresión gzip**.

Alternativamente, puedes crear un archivo **.tgz** que sea similar a **.tar.gz**. Te mostramos un ejemplo de esto último a continuación:

```
tar -cvzf sampleArchive.tgz /home/sampleArchive
```



El archivo **.bz2** proporciona más compresión en comparación con gzip. Sin embargo, esta alternativa tomará mas tiempo para comprimir y descomprimir. Para usarla, debes usar la opción **-j**. Un ejemplo de cómo se vería la operación es el siguiente:

```
tar -cvjf sampleArchive.tar.bz2 /home/sampleArchive
```

Dicha operación es similar a **.tar.tbz** o **.tar.tb2**. Te mostramos un ejemplo a continuación:

```
tar -cvjf sampleArchive.tar.tbz /home/sampleArchive
```

```
tar -cvjf sampleArchive.tar.tb2 /home/sampleArchive
```



Una vez que hayas creado el archivo, puedes listar el contenido mediante un comando similar al siguiente:

```
tar -tvf sampleArchive.tar
```

Esto mostrará la lista completa de archivos junto con las marcas de tiempo y los permisos. Del mismo modo, para **.tar.gz**, puedes usar un comando como:

```
tar -tvf sampleArchive.tar.gz
```

Esto también funcionaría para archivos **.tar.bz2** como se muestra a continuación:

```
tar -tvf sampleArchive.tar.bz2
```





El sistema de archivos de Linux utiliza el sistema de archivos de tipo ext2, ext3 o ext4. Este sistema de archivos permite almacenar los datos de los archivos en el disco duro de forma organizada y eficiente. Los archivos se almacenan en bloques y se gestionan mediante un árbol de directorios. El sistema de archivos también permite crear y eliminar archivos y directorios, así como modificar los permisos de acceso a los archivos.



Los comandos de compresión también se pueden utilizar directamente después de haber creado el paquete. A continuación se detallan algunos de estos comandos.

gzip  
gunzip

Comprime o descomprime ficheros. Reemplaza los ficheros a comprimir por otro del mismo nombre y extensión **.gz** o **.tgz** (si es un archivo de paquete).

Sintaxis:

gzip [opciones] ficheros...  
gunzip [opciones] ficheros...

Opciones:

-l, --list  
-r, --recursive  
-f, --force

Muestra información de cada fichero que comprime

Comprime los directorios y su contenido de forma recursiva.

Fuerza la compresión o descompresión, incluso si el archivo ya existe.

El sistema puede tener otros sistemas de archivos que se corresponden por estos ficheros de seguridad. Se los describe en los ficheros `etc/passwd` y `etc/group` en el caso de usuarios de sistema o `etc/passwd` y `etc/group` en el caso de usuarios de red.

- `passwd`, `shadow`, o `shadow5`, `shadow`, se corresponden con el fichero `/etc/passwd` de usuarios de sistema.
- `passwd`, `shadow`, o `shadow5`, `shadow`, se corresponden con el fichero `/etc/passwd` de usuarios de red.
- `passwd`, `shadow`, o `shadow5`, `shadow`, se corresponden con el fichero `/etc/passwd` de usuarios de red.



Figura 3.10. Ejemplo de la información con la que se desea realizar la copia de seguridad.

## Ejemplos:

```
gzip fichero.txt
gunzip fichero.txt.gz
gzip -x trabajos
gunzip -x trabajos
```

Comprime el archivo `fichero.txt` y lo reemplaza por `fichero.txt.gz`.  
Descomprime el fichero.  
Comprime todos los archivos del directorio `trabajos`.  
Descomprime los ficheros del directorio `trabajos`.



Otros comandos utilizados en las copias de seguridad son `dd`, `dump` o `restore`



`sudo dd if=Descargas/mini.iso of=/dev/sdb`

`sudo dd if=origen of=destino`



# **Realitzar Activitats Resoltes**





- Formato:

```
restore acción [opciones] [ficheros_a_recuperar]
```

- Acciones principales:

- r restaura la copia completa
- t muestra los contenidos de la copia
- x extrae sólo los ficheros indicados
- i modo interactivo
  - permite ver los ficheros de la copia
  - con add indicamos los ficheros a extraer y con extract los extraemos
  - usar ? para ayuda

- Algunas opciones:

- -f especifica el dispositivo o fichero de la copia
- -a no pregunta de que volumen extraer los ficheros (lee todos los volúmenes empezando en 1)

- Ejemplo: restaurar el backup de /dev/st0

```
# restore -rf /dev/st0
```

- Ejemplo: restaurar el backup desde un sistema remoto

```
# export RSH=ssh  
# restore -rf sistema_remoto:/dev/st0
```

- Ejemplo: restaurar sólo un fichero

```
# restore -xaf /dev/st0 fichero
```

## 3.7. PROGRAMACIÓN DE TAREAS





## 3.7. Programación de tareas

Programar una tarea consiste en programar la tarea para que se ejecute en un momento determinado.



## Programar una tarea

El siguiente ejemplo programará el comando «cat copia-seguridad.sh» para que se ejecute una vez a las 10:00 PM.

```
1. cat /dev/null > /etc/crontab
2. crontab -e
3. 0 0 10 * * * cat /dev/null > /etc/crontab
4. cat /dev/null > /etc/crontab
```

at

Este comando ejecuta la tarea en el instante indicado. Si no está instalado el comando, habrá que ejecutar lo siguiente:

```
sudo apt install at
```

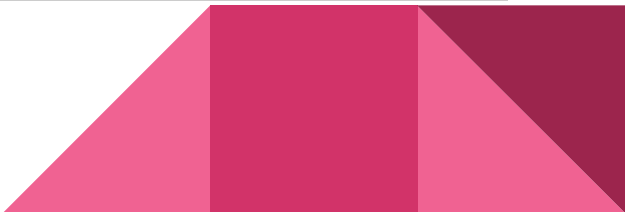


Programar una tarea a las 11 de la mañana el mismo día del próximo mes

1. at 11:00 AM next month

# Programar una tarea a las siete de la mañana del próximo domingo

1. at 07:00 AM Sun







87.500

Elimina un trabajo por su número indicado en **trabajo**.

```
2. root@servubuntu01:~# apt
```

3, 1 Mon Jul 1 22:00:00 2019 a davidochobits



## Ejecutar comandos programados

0000-0001-9340-4000

© 1997 by Blackwell Science Ltd

Downloaded from <http://ajphaphapublications.org/>

1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26

100

44

100%

Inicio	Fin	Inicio	Fin
14-03-2018	1-04-2018	1-04-2018	30-04-2018

Indikator	Skala	Unit	Salah	Skor 0-100	Angka 0-100	Angka 0-100
Indikator	Skala	Unit	Salah	Skor 0-100	Angka 0-100	Angka 0-100

Figura 3.48: I costuridels de cacha lliga del fibrous cristallals creix les que signifia cacha sfericidels plus cristallins





# **Realitzar Activitats Resoltes**





55	23	*	*	0	root	/usr/local/sbin/copiasemanal.sh
Rango	Rango	Rango	Rango	Rango		Comando
0 - 59	0 - 23	1 - 31	1 - 12	0 - 6		Usuario
						Día de la semana
						Lunes = 1, Martes = 2, Miércoles = 3 Jueves = 4, Viernes = 5, Sábado = 6, Domingo = 0
						Mes
						Enero = 1, Febrero = 2, Marzo = 3, Abril = 4, Mayo = 5, Junio = 6, Julio = 7 Agosto = 8, Septiembre = 9, Octubre = 10, Noviembre = 10, Diciembre = 12
						Día del mes
						Hora
						Minuto

# 3.8. MONITORIZACIÓN DEL SISTEMA





### 3.8.1. Rendimiento

# 3.8. Monitorización del sistema





### 3.8.1. Rendimiento

# Monitor del sistema

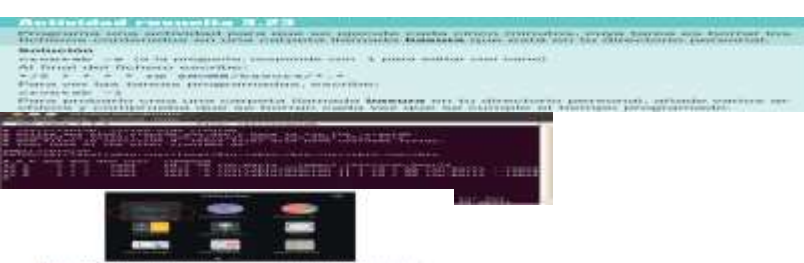


Figura 3.42. Monitorización de la CPU, de la memoria del sistema y de la red.



## Ficheros de *log*



## Carga promedio

### 3.8.2. Eventos del sistema

Cualquier actividad que sucede en el sistema se puede consultar en los ficheros **.log** que existen en el sistema operativo. Se pueden consultar a través de la aplicación **Registros** (Figura 3.43).

El comando `journalctl` permite acceder a la información contenida en los archivos de eventos del sistema o ficheros de *log*.

```
journalctl
```

Sintaxis:

```
journalctl [opciones]...[coincidencias]...
```



# **Realitzar Activitats Resoltes**





Ejemplos:

```
journalctl -r
```

Muestra todos los eventos en orden inverso.

```
journalctl -f
```

Es similar a usar `tail -f /var/log/syslog`.

```
journalctl -n10
```

Muestra las 10 últimas líneas.

```
journalctl _UID=1000
```

Muestra los mensajes del usuario cuyo UID se indique.

Para controlar los eventos de un determinado servicio:

```
journalctl -u <servicio>
```

Ejemplos (utiliza los cursores para moverte por la información y la tecla **q** para salir):

```
journalctl -r -u snapd
```

```
journalctl -r -u cron
```



# UD3 – LINUX – ADMINISTRACIÓ I CONFIGURACIÓ-II

1º DAW - CFGS

Prof. Manuel Enguidanos  
*[menguidanos@fpmislata.com](mailto:menguidanos@fpmislata.com)*