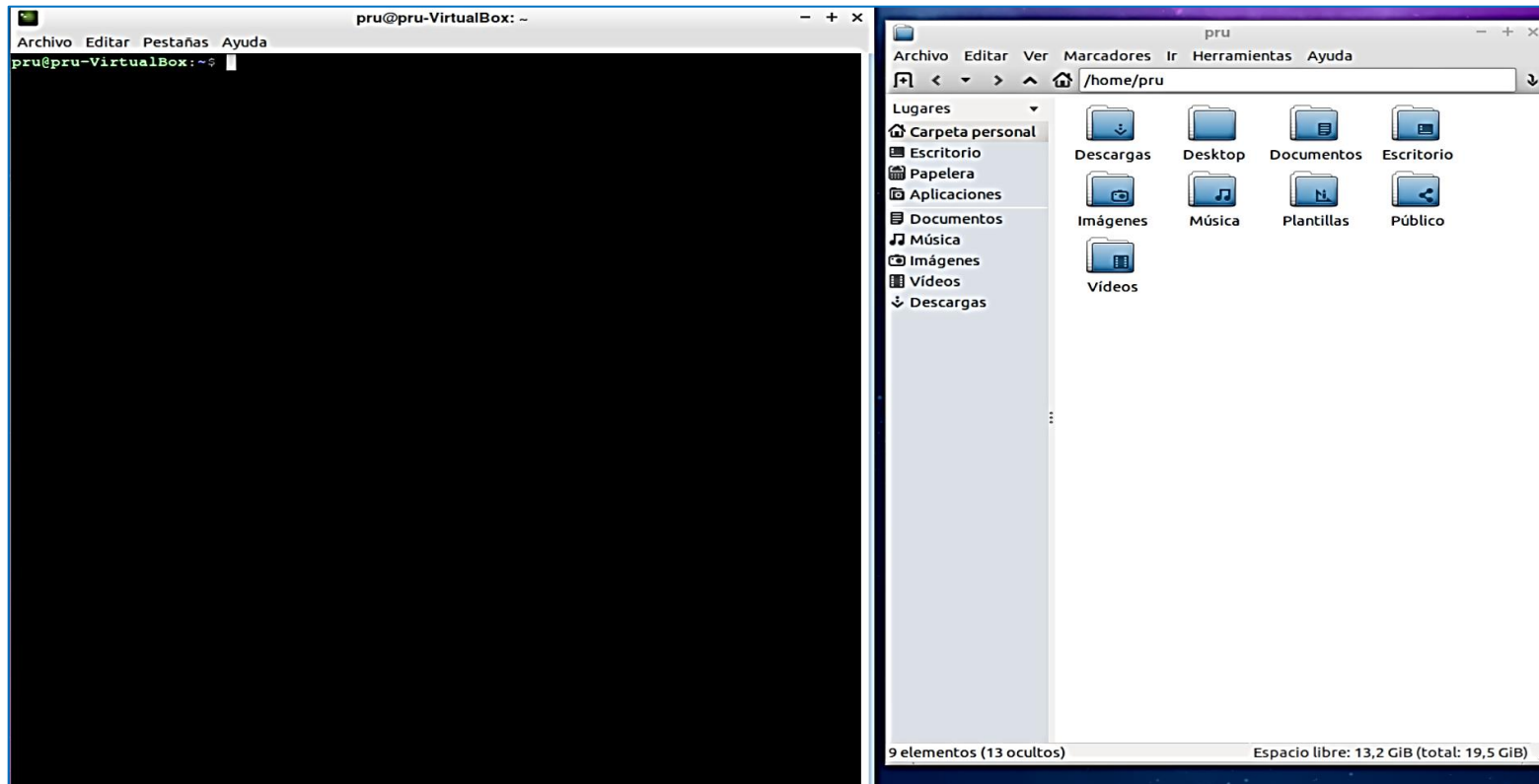


Actividad 03. Uso de la terminal y la línea de comandos en Linux

1. Abrir la terminal y gestor de archivos



Actividad 03. Uso de la terminal y la línea de comandos en Linux

2. Shell

Que es una Shell

El shell **es el entorno que hace de intermediario entre el usuario y los recursos del ordenador**, como si fuera un entorno de programación en tiempo real para ejecutar tareas. La shell aparece cuando nos logueamos en el sistema en modo consola, o bien cuando en el entorno gráfico abrimos una terminal.

La shell se compone de **un prompt**, que es un texto inicial que normalmente nos ofrece información útil como el usuario que está utilizando la shell, **el hostname** de la máquina o incluso **el directorio sobre el que estamos posicionados** en cada momento, **y el cursor** que recibe las ordenes de teclado.

Diferentes shells en Linux

Existen multitud de shells diferentes para poder interactuar con nuestro sistema operativo, aunque la más conocida y habitual en la mayoría de distribuciones es la shell bash. Cada shell cuenta con sus propias características de uso, contando con atajos de teclado, visualización en vivo de ficheros, y atajos durante la navegación entre directorios. A continuación vamos a citar las shells mas famosas y una pequeña descripción de su procedencia:

- **bash (Bourne Again Shell):** Se basa en los principios de shell Bourne de Unix pero se ha extendido en varios aspectos. Es la shell por defecto para la mayoría de las cuentas de usuario y es la que se tratará con más detalle en este curso.
- **bsh:** El shell Bourne es la shell sobre la que está basada bash. Se conoce con el nombre de BSH. Su uso no es frecuente en Linux aunque el comando bsh suele ser un enlace simbólico a bash.

Actividad 03. Uso de la terminal y la línea de comandos en Linux

- **tcsh:** Este shell tcsh se basa en el anterior shell C (csh). Es una shell bastante popular en algunos círculos pero no hay distribuciones de Linux que lo traigan por defecto. Aunque es similar a bash en muchos aspectos difiere en algunos aspectos de operación. Por ejemplo, no se asignan variables de entorno de la misma manera que en bash.
- **csh:** El original csh shell C no es muy utilizado en Linux pero si un usuario está familiarizado con csh, tcsh es un buen sustituto.
- **ksh:** El shell Korn (ksh) fue diseñado cogiendo las mejores opciones de la shell Bourne y el C shell. Tiene un pequeño pero dedicado numero de seguidores.
- **zsh:** El shell zsh Z (zsh) es la evolución de la shell Korn e incorpora características de esta última además de agregar otras.

Actividad 03. Uso de la terminal y la línea de comandos en Linux

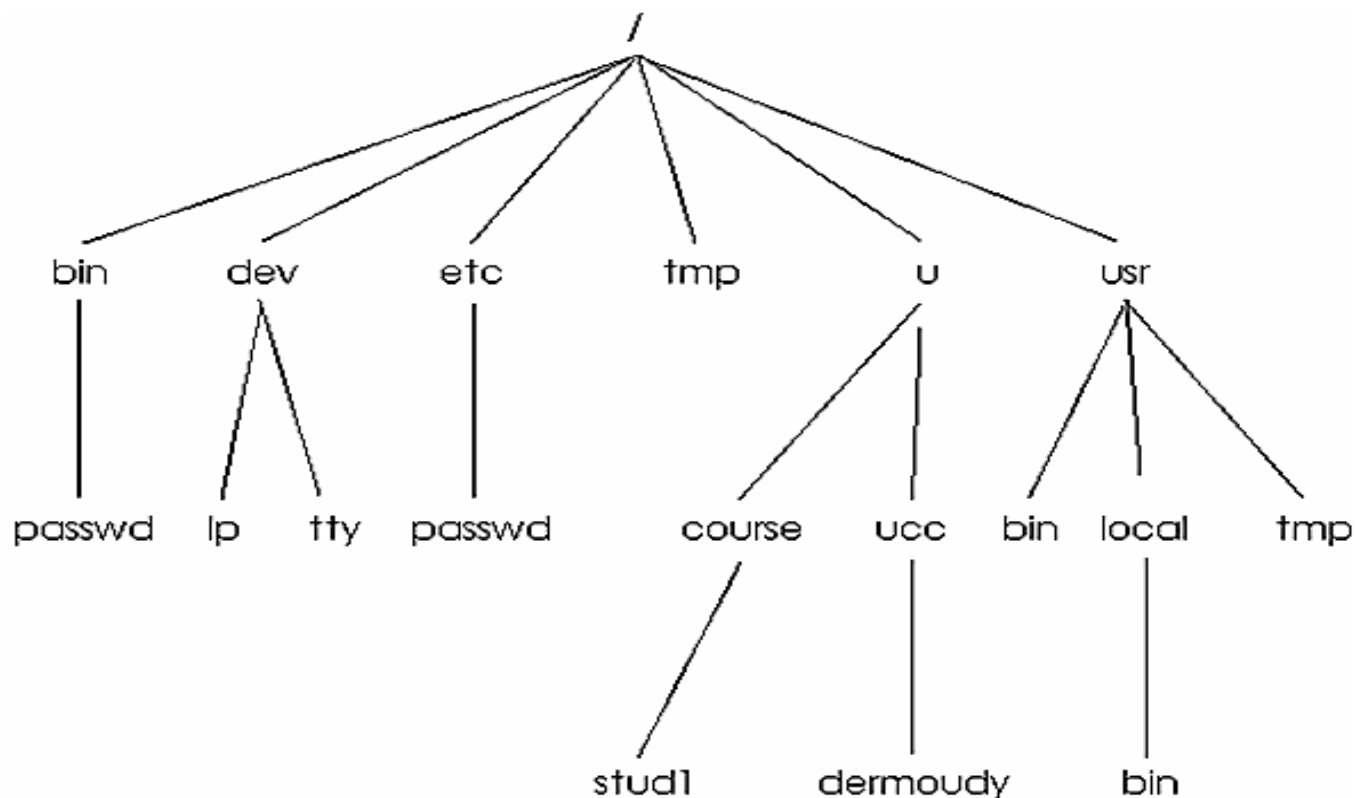
3. Sistema de archivos

Directorios y sistemas de archivos

En Linux y Unix todo es un fichero. Los directorios son ficheros, los ficheros son ficheros, y los dispositivos son ficheros. A veces a los dispositivos se les llama nodos, pero siguen siendo ficheros. Los sistemas de ficheros de Linux y Unix se organizan en una estructura jerárquica, de tipo árbol. El nivel más alto del sistema de ficheros es / o directorio raíz.

Todos los demás ficheros y directorios están bajo el directorio raíz. Por ejemplo, /home/pru/cheeses.odt muestra la ruta completa al fichero cheeses.odt que está en el directorio pru, que a su vez está bajo el directorio home, que por su parte está bajo el directorio raíz (/).

Por debajo del directorio raíz (/) hay un importante grupo de directorios común a la mayoría de las distribuciones de GNU/Linux. A continuación, hay una lista de los directorios que aparecen normalmente bajo el directorio raíz (/):



Actividad 03. Uso de la terminal y la línea de comandos en Linux

/	Directorio raíz o sistema de ficheros raíz. Todos los demás directorios se ramifican a partir de este. Linux no utiliza letras de unidad como Windows, sino que las particiones o discos se montan en un punto dentro del sistema de ficheros raíz.
/boot	Ficheros de configuración del arranque, núcleos y otros ficheros necesarios para el arranque (<i>boot</i>) del equipo.
/bin	Aplicaciones <i>binarias</i> importantes. Son ficheros ejecutables que son accesibles para todos los usuarios y constituyen los comandos más importantes.
/sbin	Se almacenan las <i>aplicaciones binarias</i> importantes del sistema. Son programas que solo puede ejecutar el administrador (<i>root</i>)
/lib	Librerías del sistema (<i>libraries</i>). Son bibliotecas de programas que son código compartido por muchos programas y que se almacenan en ficheros independientes.
/usr	Las aplicaciones y archivos a los que puede acceder la mayoría de los <i>usuarios</i>
/dev	Los ficheros de <i>dispositivo</i>
/etc	ficheros de configuración, scripts de arranque, <i>etc.</i>
/home	Directorios personales (<i>home</i>) para los diferentes usuarios.
/initrd	usado cuando se crea un proceso de arranque <i>initrd</i> personalizado.
/lost+found	proporciona un sistema de "perdido+encontrado" (<i>lost+found</i>) para los ficheros que existen debajo del directorio raíz (/)
/media	Las particiones montadas (cargadas) automáticamente en el disco duro y medios (<i>media</i>) extraíbles como CDs, cámaras digitales, etc.
/mnt	Los sistemas de archivos <i>montados</i> manualmente en el disco duro. Suelen ser unidades de red
/opt	Proporciona una ubicación donde instalar aplicaciones opcionales (de terceros)
/proc	Es un directorio dinámico especial que mantiene información sobre el estado del sistema, incluyendo los <i>procesos</i> actualmente en ejecución
/root	Es el directorio personal del usuario <i>root</i> (superusuario); también llamado "barra-root".
/srv	puede contener archivos que se <i>sirven</i> a otros sistemas
/sys	archivos del sistema (<i>system</i>)
/tmp	Se crean y almacenan los ficheros temporales (<i>temporary files</i>). La mayoría de las distribuciones limpian este directorio cada vez de se inicia el equipo.
/var	Contiene archivos <i>variables</i> como archivos de registros y bases de datos

Actividad 03. Uso de la terminal y la línea de comandos en Linux

4. Cómo entender la información al inicio de la terminal

Cuando abrimos la terminal vamos a ver algo como esto:

pru@pru-VirtualBox:~\$

Todo esto parece un mensaje encriptado, pero es más sencillo de lo que parece, vamos por partes:

- **pru** es el nombre del usuario que está activo. En tu caso aparecerá el nombre del usuario que esté activo en tu ordenador.
- **pru-VirtualBox** es el nombre que el sistema operativo le dio a tu máquina.
- **~ (virgulilla) es la carpeta del usuario:** /home/pru
- **\$** significa que somos un usuario normal.

Nota. El usuario root tiene el símbolo #

5. ¿Qué es un comando? Tipos y ayuda:

Un **comando** es un mensaje enviado al ordenador que provoca una respuesta en este sistema y se comporta como una **orden**, pues informa al dispositivo informático que debe ejecutar una acción según la indicación que pueda enviarse. Cada sistema operativo incorpora un determinado número de comandos básicos, que permiten ejecutar las tareas más simples con órdenes directas.

A continuación, conocerás todo lo relacionado con sistemas operativos basados en UNIX y sus comandos básicos en la terminal.

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Un **comando** puede significar cuatro cosas:

- **Un programa ejecutable**
- **Un comando de utilidad de la shell.** Esto es un programa en sí mismo, que puede tener funciones. Ejemplo **cd**
- **Una función de shell.** Son funciones de shell externas al comando de utilidad. Ejemplo **mkdir**
- **Un alias.** Un ejemplo es **ls**

Los siguientes comandos nos permiten obtener información sobre los comandos:

- `type <comando>`: Nos permite conocer qué tipo de comando es.
- `alias l="<secuencia de comandos>"`: Nos permite crear comandos. Son temporales, se borran al cerrar la terminal.
- `help <comando>`: Nos permite consultar un poco de documentación de un comando.
- `man <comando>`: De *manual*, nos permite conocer mucha más información de un comando.
- `info <comando>`: Similar al anterior, pero un poco resumido y con otro formato.
- `whatis <comando>`: Describe un comando en una sola línea. No funciona con todos.

6. Comandos básicos

- `pwd` – Indica el directorio activo o actual
- `cd` – Cambia el directorio actual
 - `cd /` Lleva al directorio raíz
 - `cd ~` Lleva al directorio del usuario (/home/pru)
 - `cd ..` Lleva al directorio anterior (a un nivel superior en la jeraquía)

Actividad 03. Uso de la terminal y la línea de comandos en Linux

- El símbolo / sirve para identificar directorios. Por ejemplo: `cd /home/pru/Escritorio` lleva al directorio Escritorio, que está dentro del directorio pru, que, a su vez, está dentro del directorio home
- `cd .` Lleva al directorio actual
- `cd ~` Lleva al directorio de usuario

Podemos desplazarnos de forma absoluta o relativa

- `ls` Muestra el contenido de un directorio:

```
miguelangel@DESKTOP-3R804MK:/$ ls
bin  dev  home  lib    lib64  lost+found  mnt  proc  run  snap  sys  usr
boot  etc  init  lib32  libx32  media      opt  root  sbin  srv   tmp  var
miguelangel@DESKTOP-3R804MK:/$
```

Para saber información adicional como la fecha de creación, el peso, los permisos, etc., utilizamos la opción `-l` que significa **Long**:

```
miguelangel@DESKTOP-3R804MK:/$ ls -l
total 684
lrwxrwxrwx 1 root root 7 Mar 24 17:40 bin -> usr/bin
drwxr-xr-x 2 root root 4096 Mar 24 17:47 boot
drwxr-xr-x 8 root root 2720 Apr 28 17:12 dev
drwxr-xr-x 97 root root 4096 Apr 28 17:12 etc
drwxr-xr-x 3 root root 4096 Apr 28 15:19 home
-rwxr-xr-x 2 root root 632096 Apr 23 19:19 init
```


Actividad 03. Uso de la terminal y la línea de comandos en Linux

Podemos añadir la opción **-h** que significa **Human Readable**:

```
miguelangel@DESKTOP-3R804MK:/$ ls -lh
total 684K
lrwxrwxrwx  1 root root    7 Mar 24 17:40 bin -> usr/bin
drwxr-xr-x  2 root root 4.0K Mar 24 17:47 boot
drwxr-xr-x  8 root root 2.7K Apr 28 17:12 dev
drwxr-xr-x 97 root root 4.0K Apr 28 17:12 etc
drwxr-xr-x  3 root root 4.0K Apr 28 15:19 home
-rwxr-xr-x  2 root root 618K Apr 23 19:19 init
```

Nota: para añadir más de una opción no hacemos esto `ls -l -h` sino esto `ls -lh`

- `file` Muestra información sobre el tipo de elemento que hayas seleccionado:

```
pru@pru-VirtualBox:/$ ls
bin      dev  initrd.img  lib64      mnt  root  srv  texto  var
boot     etc  initrd.img.old  lost+found  opt  run  swapfile  tmp  vmlinuz
cdrom    home lib          media      proc  sbin  sys      usr  vmlinuz.old

pru@pru-VirtualBox:/$ file texto
texto: empty

pru@pru-VirtualBox:/$ file bin
bin: directory

pru@pru-VirtualBox:/$ file initrd.img
initrd.img: symbolic link to boot/initrd.img-5.4.0-139-generic

pru@pru-VirtualBox:/$ file tmp
tmp: sticky, directory

pru@pru-VirtualBox:/$ file swapfile
swapfile: regular file, no read permission
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

- `mkdir` Crea carpetas:

```
pru@pru-VirtualBox:~$ ls
Descargas Desktop Documentos Escritorio Imágenes Música Plantillas Público Vídeos
pru@pru-VirtualBox:~$ mkdir pruebas
pru@pru-VirtualBox:~$ ls
Descargas Documentos Imágenes Plantillas Público
Desktop Escritorio Música pruebas Vídeos
```

- `rmdir` Elimina carpetas:

```
pru@pru-VirtualBox:~$ ls
Descargas Documentos Escritorio Imágenes Música Plantillas pruebas Público Vídeos
pru@pru-VirtualBox:~$ rmdir pruebas
pru@pru-VirtualBox:~$ ls
Descargas Documentos Escritorio Imágenes Música Plantillas Público Vídeos
pru@pru-VirtualBox:~$
```

- `touch` Crea archivos:

```
pru@pru-VirtualBox:~/pruebas$ ls
pru@pru-VirtualBox:~/pruebas$ touch texto1 texto2 texto3 texto4
pru@pru-VirtualBox:~/pruebas$ ls
texto1 texto2 texto3 texto4
```

- `cp` Copia archivos:

```
pru@pru-VirtualBox:~/pruebas$ ls
texto1 texto2 texto3 texto4
pru@pru-VirtualBox:~/pruebas$ cp texto4 texto5
pru@pru-VirtualBox:~/pruebas$ ls
texto1 texto2 texto3 texto4 texto5
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

- **mv** Mueve archivos:

```
pru@pru-VirtualBox:~/pruebas$ ls
texto1 texto2 texto3 texto4 texto5
pru@pru-VirtualBox:~/pruebas$ mv texto5 ..
pru@pru-VirtualBox:~/pruebas$ ls
texto1 texto2 texto3 texto4
pru@pru-VirtualBox:~/pruebas$ cd ..
pru@pru-VirtualBox:~$ ls
Descargas Documentos Imágenes Plantillas Público Vídeos
Desktop Escritorio Música pruebas texto5
```

- **rm** Elimina archivos:

```
pru@pru-VirtualBox:~$ ls
Descargas Documentos Imágenes Plantillas Público Vídeos
Desktop Escritorio Música pruebas texto5
pru@pru-VirtualBox:~$ rm texto5
pru@pru-VirtualBox:~$ ls
Descargas Documentos Imágenes Plantillas Público
Desktop Escritorio Música pruebas Vídeos
```

El comando **rm** tiene varias opciones. Las más usadas son:

- i (de interactive) te pregunta si estás seguro de eliminar el archivo
- r (de recursive) elimina todo lo que esté dentro de una carpeta
- f (de force) fuerza a borrar todo.

Actividad 03. Uso de la terminal y la línea de comandos en Linux

```
pru@pru-VirtualBox:~$ ls
Descargas Documentos Imágenes Plantillas pruebas1 Vídeos
Desktop Escritorio Música pruebas Público
pru@pru-VirtualBox:~$ ls pruebas
pruebas1 texto1 texto2 texto3 texto4
pru@pru-VirtualBox:~$ rm -r pruebas
pru@pru-VirtualBox:~$ ls
Descargas Documentos Imágenes Plantillas Público
Desktop Escritorio Música pruebas1 Vídeos
```

- Comandos para visualizar y editar archivos:

Comando	Función
head	Muestra las primeras 10 líneas
tail	Muestra las últimas 10 líneas
less	Muestra todo el contenido dentro de la consola
xdg-open	Abre el editor de texto por defecto para inspeccionar ese archivo
nautilus	Abre en la interfaz de ventanas la carpeta que selecciones

7. Uso de comodines (Wildcards):

Los **comodines o wildcards** son una serie de **caracteres especiales** que nos permiten encontrar patrones o realizar **búsquedas más avanzadas**. Es aplicable para archivos y directorios.

- **Buscar todo (*)**

Actividad 03. Uso de la terminal y la línea de comandos en Linux

El asterisco te ayuda a buscar toda la información dentro de una carpeta, pero puedes limitar su uso. Si por ejemplo quieres buscar los archivos que tengan una extensión “.png”, escribes:

```
ls -l *.png
```

También lo puedes poner al final, si quisieras buscar, todos los archivos que comiencen por unos caracteres específicos, entonces escribes esos caracteres y luego el asterisco.

Por ejemplo, si quisieras buscar todos los archivos que comiencen por “fotosDe”, habría que escribir:

```
ls -l fotoDe*
```

- **Buscar por cantidad de caracteres (?)**

Si dentro de tus archivos tuvieras una especie de código para guardar tus fotos, algo así como “foto1.png”, “foto2.png”, “foto3.png”, etc. En este caso, sabemos que primero tenemos el string “foto”, luego un solo número y por último la extensión “.png”.

Si quisieras buscar esas fotos escribirías:

```
ls -l foto?.png
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

- **Buscar por caracteres específicos ([])**

Si quieres buscar por varios caracteres específicos se usan corchetes. Para utilizarlos tienes que colocar dentro de los corchetes los caracteres que quieres buscar.

Por ejemplo, si quisieras buscar los archivos que comiencen por las letras “c” o “i”, entonces escribes:

```
ls -l [ci]*
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

8. Redirecciones

En la consola nosotros generamos una entrada cuando escribimos y una salida casi siempre que ejecutamos un comando.

A las entradas típicamente se les suele llamar **Standard Input** y a las salidas **Standard Output**, además se les suele abreviar como **stdin** y **stdout** respectivamente.

Los **file descriptors** son números que identifican un recurso. Funciona asociando un número con una acción, archivo o programa, en el caso de la shell tenemos 3 file descriptors:



Actividad 03. Uso de la terminal y la línea de comandos en Linux

Operador de redirección (>)

A veces queremos guardar la información de una salida porque nos puede interesar almacenar lo que esa salida contiene.

Si quieres que el Standard Output no vaya a la consola sino hacia un archivo, entonces puedes usar el operador > seguido del nombre del archivo en el que quieres guardar la salida.

```
ls -l > output.txt
```

Concatenar (>>)

Para añadir contenido a un fichero ya existente, realizas:

```
ls -l >> output.txt
```

Redirección de errores (2>)

El operador de redirección por defecto solo redirecciona el file descriptor 1 (es decir, el Standard Output). Pero, ¿qué tal si queremos redirigir un error? Pues tenemos que especificar que queremos el Standar Error, que tiene el file descriptor 2.

Vamos a generar un error ejecutando un comando que saldrá mal para redirigirlo a un archivo llamado “error.txt”.:

Actividad 03. Uso de la terminal y la línea de comandos en Linux

```
ls -lñ 2> error.txt
```

También podemos especificar que no importa lo que pase si me da un Standard Output o un Standard Error, igual tiene que guardar la salida en un archivo. Esto lo hacemos así:

```
ls -l > output.txt 2>&1
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

9. Encadenación de comandos

Operador	Función
;	Ejecuta de forma síncrona los comandos especificados
&	Ejecuta de forma asíncrona los comandos especificados
&&	Ejecuta el comando si el anterior se ejecutó correctamente
	Ejecuta el comando si el anterior o la combinación de los anteriores resultaron en verdadero

síncrona

```
ls; mkdir pruebas; cd pruebas; ls; cd ..; ls; date
```

```
pru@pru-VirtualBox:~$ ls; mkdir pruebas; cd pruebas; ls; cd ..; ls; date
Descargas Documentos Escritorio Imágenes Música Plantillas Público Vídeos
Descargas Documentos Escritorio Imágenes Música Plantillas pruebas Público Vídeos
jue feb 23 06:00:02 WET 2023
pru@pru-VirtualBox:~$
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

asíncrona

```
date & echo "hola" & cal
```

```
pru@pru-VirtualBox:~$ date & echo "Hola" & cal
[1] 1483
[2] 1484
Hola
jue feb 23 06:03:04 WET 2023
    Febrero 2023
do lu ma mi ju vi sa
      1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

[1]- Hecho          date
[2]+ Hecho          echo "Hola"
```

condicional

```
cd pruebas && mkdir pruebas2 && ls
```

```
pru@pru-VirtualBox:~$ ls
Descargas  Documentos  Escritorio  Imágenes  Música  Plantillas  pruebas  Público  Vídeos
pru@pru-VirtualBox:~$ cd pruebas && mkdir pruebas2 && ls
pruebas2
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

10. Gestión de permisos

Veamos, en primer lugar, los **TIPOS DE ARCHIVO**:

Atributo	Tipo de archivo
-	Archivo normal
d	Directorio
l	Link simbólico
b	Archivo de bloque especial

```
pru@pru-VirtualBox:~$ ls -l
total 36
drwxr-xr-x 2 pru pru 4096 feb 23 05:52 Descargas
drwxr-xr-x 2 pru pru 4096 feb 23 05:52 Documentos
drwxr-xr-x 3 pru pru 4096 feb 23 06:16 Escritorio
drwxr-xr-x 2 pru pru 4096 feb 23 05:52 Imágenes
lrwxrwxrwx 1 pru pru 30 feb 23 06:22 'mis cosas' -> '/home/pru/Escritorio/mis cosas'
drwxr-xr-x 2 pru pru 4096 feb 23 05:52 Música
drwxr-xr-x 2 pru pru 4096 feb 23 05:52 Plantillas
drwxrwxr-x 3 pru pru 4096 feb 23 06:09 pruebas
drwxr-xr-x 2 pru pru 4096 feb 23 05:52 Público
-rw-rw-r-- 1 pru pru 0 feb 23 06:22 texto.txt
drwxr-xr-x 2 pru pru 4096 feb 23 05:52 Vídeos
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Permisos de usuario

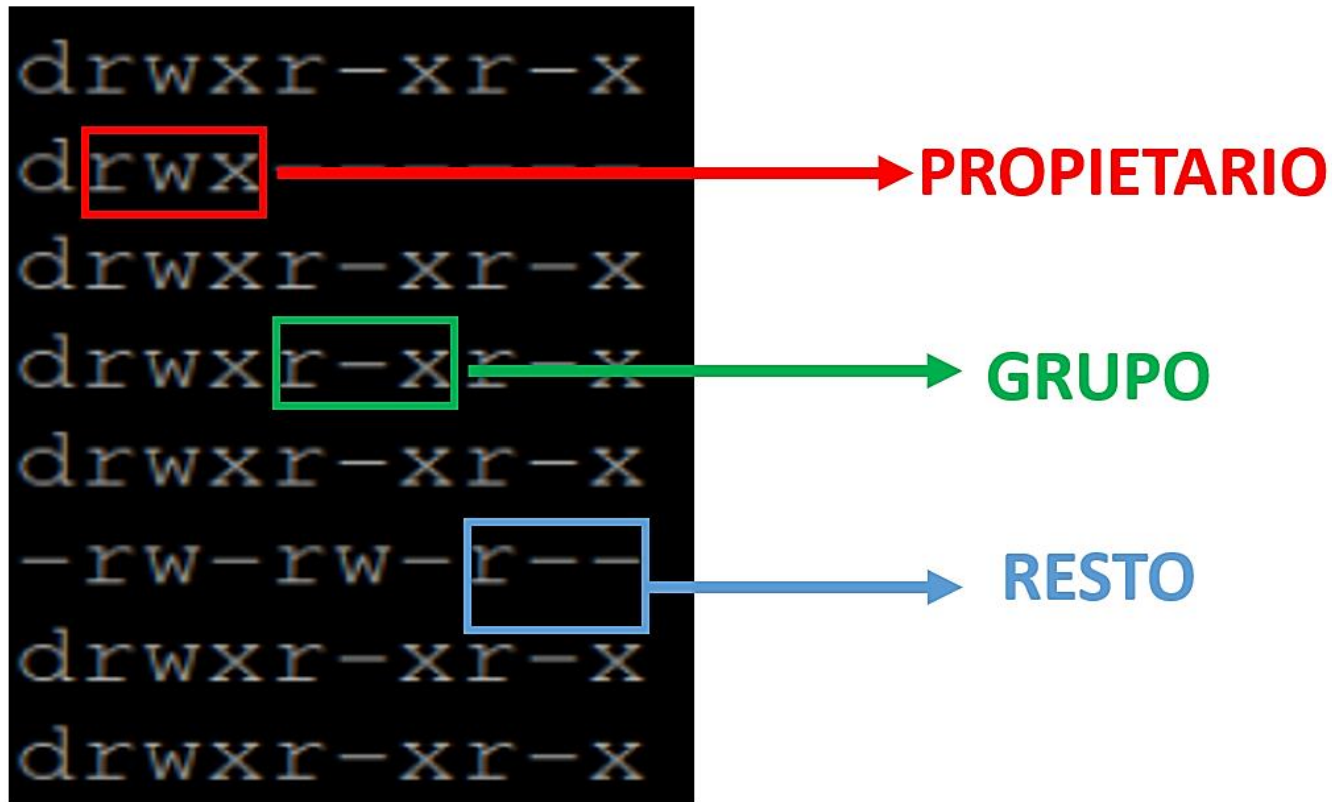
Tipos de usuario

Tipo	Descripción
Owner	El propietario del archivo, si no se ha cambiado, es quien lo creó y tiene mayor jerarquía sobre los otros 3. Le corresponden los primeros 3 caracteres de los permisos
Group	Se puede crear grupos de usuarios para darle a todos o varios los mismos permisos. A estos usuarios le corresponden el cuarto, quinto y sexto carácter de los permisos de usuarios y tienen mayor jerarquía que el último
World	También llamado “ otros ”, es cualquier otro usuario que no pertenezca a un grupo de usuario y tampoco sea el dueño, este tiene la menor jerarquía

Tipos de permiso

Símbolo	Tipo	Permiso
R	Lectura	El usuario Puede leer su contenido
W	Escritura	El usuario puede editar el contenido del archivo, también el nombre y los permisos
X	Ejecución	El usuario puede ejecutarlo en caso de que sea un programa

Permisos



Actividad 03. Uso de la terminal y la línea de comandos en Linux

Ejemplo:

```
pru@pru-VirtualBox:~$ ls -l
total 44
drwxr-xr-x 2 pru pru 4096 feb 21 05:02 Descargas
drwx----- 2 pru pru 4096 feb 21 05:02 Desktop
drwxr-xr-x 2 pru pru 4096 feb 22 09:21 Documentos
drwxr-xr-x 2 pru pru 4096 feb 21 10:35 Escritorio
drwxr-xr-x 2 pru pru 4096 feb 22 09:18 Imágenes
-rw-rw-r-- 1 pru pru 111 feb 22 06:19 miscarpetas.txt
```

**Tipo de
archivo**

Atributo	Tipo de archivo
-	Archivo normal
d	Directorio
l	Link simbólico
b	Archivo de bloque especial

Permisos

Símbolo	Permiso
R	Lectura
W	Escritura
X	Ejecución

Permisos

R	W	X	Resultado	Permiso
0	0	0	0	Ninguno
0	0	1	1	Ejecución
0	1	0	2	Escritura
0	1	1	3	Escritura y ejecución
1	0	0	4	Lectura
1	0	1	5	Lectura y ejecución
1	1	0	6	Lectura y escritura
1	1	1	7	Todos

Ejemplos permisos

PROPIETARIO

GRUPO

RESTO

-	-	-	-	-	-	-	-	-
0	0	0	0	0	0	0	0	0
0			0			0		

000

R	W	X	R	-	X	R	-	-
1	1	1	1	0	1	1	0	0
7			5			4		

754

R	W	X	R	W	X	R	W	X
1	1	1	1	1	1	1	1	1
7			7			7		

777

Permisos

Cambiar permisos de archivos

Sintáxis

chmod [**Usuario**][**Operador**][**Permiso**] [archivo directorio]

Usuario		
Propietario	Grupo	Otro
u	g	o

Operador	
+	Añade un permiso
-	Quita un permiso
=	Asigna un permiso

Permiso	
r	Lectura
w	Escritura
x	Ejecución

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Ejemplos:

Supongamos que queremos añadirle permiso de escritura al grupo, entonces tenemos que escribir lo siguiente:

chmod g+w archivo.txt

Puedes cambiar varios permisos de varios usuarios al mismo tiempo, por ejemplo, si quisieras agregar el permiso de escritura y ejecución al grupo y a otros, sería así:

chmod go+wx archivo.txt

Y si quieres permisos diferentes para cada usuario, solo sepáralos por comas:

chmod u+r,g=w archivo.txt

En ese comando se le añadió el permiso de lectura al dueño y de escritura al grupo. No agregues espacio en las comas o provocarás un error.

También puedes cambiar los permisos usando su forma octal, por ejemplo el conjunto de permisos rwxr-xr-x en su forma octal es 755:

chmod 755 archivo.txt

Permisos

Cambiar propietario de archivo

Sintaxis

chown [usuarioAlQuePertenece] [archivo]

Ejemplo:

Chown pru1 archivo.txt

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Comando	Función
whoami	Muestra el usuario con el que se está trabajando
su	Switch User Cambia al usuario al que le especifiques
chmod	Cambia los permisos de un archivo
chown	Change Owner Cambia el propietario de un archivo

Actividad 03. Uso de la terminal y la línea de comandos en Linux

11. Búsquedas

A veces necesitas localizar varios archivos del mismo tipo que ocupan espacio innecesario en tu disco duro.

Por ejemplo, algunos programas que funcionan desde la consola, como npm, guardan sus errores en archivos de extensión “.log” y si no estás pendiente de eliminarlos se van acumulando en tu disco duro.

¿Cuáles son los comandos de búsqueda en la terminal?

Para encontrar archivos de forma efectiva, usa el comando **find**, el cual buscará en la ruta que le indiques el tipo de archivos que necesitas. Su sintaxis es:

find [rutaDesdeDondeEmpezarBuscar] [opciones]

Segmentar por el nombre (-name)

Veamos un ejemplo, voy a buscar en mi carpeta home todos los archivos que tenga una extensión “.png”.

```
find -name *.png
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Segmentar por el tipo (-type)

También puedes segmentar por el tipo, si es un archivo o si es un directorio utilizando la opción **-type**, el cual acepta **f** para archivos, **d** para directorios y **l** para enlaces simbólicos.

Si quieres usar más de una opción lo separas por comas.

```
find -type f -name "f*"
```

Segmentar por tamaño (-size)

Con la opción **-size** podemos segmentar por tamaño ingresando el peso que queremos buscar. Esta opción tiene un uso un tanto especial. Primero que todo hay que colocar la unidad de peso: **c** para byte, **k** para Kilobyte, **M** para Megabyte y **G** para Gygabyte

Podemos especificar que sea ese peso en adelante con el símbolo **+** o de ese peso para abajo con el símbolo **-**.

```
find -size +2M
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Buscar vacíos (-empty)

Para buscar los archivos vacíos usamos la opción **-empty** que es fácil de usar, no tienes que especificarle nada, solo escribirla.

```
find -empty
```

Limitar la búsqueda (-maxdepth -mindepth)

Puede que no queramos buscar en absolutamente todas las carpetas del sistema, podemos limitar la profundidad de carpetas a la que el comando debe buscar, esto se hace con la opción **-maxdepth** seguido de la profundidad.

```
find -empty -maxdepth 1
```

Continuando, a veces ya conocemos más o menos la estructura de nuestras carpetas, así que nos queremos saltar niveles, por lo que le asignamos una profundidad mínima al comando.

```
find -empty -mindepth 3
```


Actividad 03. Uso de la terminal y la línea de comandos en Linux

12.Utilidades de red

Veamos algunos comandos básicos de utilidades de la red para que puedas obtener la información que necesites.

Configuración de tus dispositivos (ifconfig)

Ve a tu consola, escribe el comando ifconfig y miremos el resultado. Muestra información sobre los dispositivos e red de la máquina:

```
pru@pru-VirtualBox:~/Documentos$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
    inet6 fe80::5937:9499:8200:e9d3  prefixlen 64  scopeid 0x20<link>
    ether 08:00:27:5b:eb:03  txqueuelen 1000  (Ethernet)
    RX packets 1868  bytes 702133 (702.1 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1669  bytes 1555421 (1.5 MB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
    inet 127.0.0.1  netmask 255.0.0.0
    inet6 ::1  prefixlen 128  scopeid 0x10<host>
    loop txqueuelen 1000  (Bucle local)
    RX packets 210  bytes 25388 (25.3 KB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 210  bytes 25388 (25.3 KB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Enviar solicitudes a una página (ping)

A veces queremos saber si una página está disponible desde nuestra dirección IP. Para esto escribimos el comando seguido de la URL a la que queremos acceder.

El comando ping envía paquetes a esa página y evalúa el tiempo de respuesta.

Por defecto, el comando se ejecutará indefinidamente, así que tienes que detenerlo con **ctrl + c**:

```
pru@pru-VirtualBox: ~/Documentos$ ping www.google.es
PING www.google.es (142.250.185.3) 56(84) bytes of data.
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=1 ttl=115 time=36.3 ms
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=2 ttl=115 time=33.2 ms
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=3 ttl=115 time=33.2 ms
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=4 ttl=115 time=36.4 ms
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=5 ttl=115 time=33.6 ms
^C
--- www.google.es ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4008ms
rtt min/avg/max/mdev = 33.267/34.599/36.406/1.468 ms
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Limitar los paquetes enviados (-c)

Para limitar la cantidad de paquetes que enviamos, usamos la opción **-c** seguida del número de paquetes por enviar.

```
pru@pru-VirtualBox: ~/Documentos$ ping -c 6 www.google.es
PING www.google.es (142.250.185.3) 56(84) bytes of data.
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=1 ttl=115 time=37.8 ms
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=2 ttl=115 time=34.9 ms
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=3 ttl=115 time=35.1 ms
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=4 ttl=115 time=37.9 ms
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=5 ttl=115 time=35.2 ms
64 bytes from mad41s11-in-f3.1e100.net (142.250.185.3): icmp_seq=6 ttl=115 time=35.3 ms

--- www.google.es ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5014ms
rtt min/avg/max/mdev = 34.939/36.071/37.923/1.297 ms
```


Actividad 03. Uso de la terminal y la línea de comandos en Linux

13.Comprimiendo archivos tar y zip

Puedes aprender a crear archivos comprimidos .zip o .tar que vemos en nuestro sistema operativo. Estos encapsulan muchos archivos e incluso carpetas.

Comprimiendo archivos con formato .tar

El formato .tar es un tipo de compresión bastante usado en UNIX. Originalmente era utilizado para almacenar información en cintas magnéticas, así que está hecho especialmente para comprimir los archivos de forma lineal.

Para comprimir con este formato en la terminal usamos el comando **tar** que tiene ciertas opciones para aprender.

Sintaxis:

tar [opciones] [nombreDelArchivoComprimido] [archivoAComprimir]

Comprimir (-c)

Para comprimir un archivo utilizamos la opción -c. En todos los casos hay que usar la opción -f para indicar que estamos comprimiendo o descomprimiendo archivos.

```
tar -cf compressed.tar Documents/toCompress/
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Ver lo que está haciendo el comando (-v)

Si queremos ver lo que el comando está comprimiendo a medida que se va ejecutando, usamos la opción -v. Por cierto, la opción -v es de “Verbose” y muchos comandos la usan, también te la puedes encontrar como --verbose.

```
tar -cvf compressed.tar Documents/toCompress/
```

Comprimir con formato “.tar.gz” (-z)

El formato “.tar.gz” o también “.tgz” es una versión extendida del formato tradicional de compresión “.zip” que puede manejar y comprimir archivos más grandes.

Para manejar la compresión de archivos “.tar.gz” o “.tgz” se usa la opción -z además de tener que especificar en el nombre de archivo la extensión que quieres usar.

```
tar -czvf compressed.tar.gz Documents/toCompress/
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Descomprimir (-x)

Para descomprimir es mucho más sencillo, solo hay que especificar la opción **-x** y el archivo comprimido que se quiere descomprimir.

Si se quiere descomprimir un archivo de extensión “.tar.gz” o “.tgz” hay que especificar la opción **-z** también.

```
tar -xzvf compressed.tar.gz
```

Comprimiendo archivos .zip

Para comprimir usamos el comando zip con el nombre que quieres que tenga y lo que quieres comprimir.

Si quieres comprimir una carpeta con archivos dentro, tienes que especificar la opción **-r** de “recursive”.

```
zip -r compressed.zip Documents/toCompress/
```

Y para descomprimir es incluso más fácil, solo escribes el comando unzip seguido de lo que quieres descomprimir.

```
unzip compressed.zip
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Gestión de procesos

Puedes manejar los procesos como visualizar, eliminar o filtrar desde la terminal de un modo diferente a usar ctrl + alt +suprimir. Te explicamos cómo lograrlo.

Ver los procesos activos en la terminal (ps)

El comando **ps** muestra los procesos que están activos en una tabla muy sencilla de entender, donde en la primera columna tenemos el process ID y en la última el nombre.

```
pru@pru-VirtualBox:~$ ps
  PID TTY          TIME CMD
 5380 pts/3        00:00:00 bash
 5390 pts/3        00:00:00 ps
pru@pru-VirtualBox:~$ ls
```


Actividad 03. Uso de la terminal y la línea de comandos en Linux

Ver procesos más detallados (top)

Si quieres ver una lista más detallada de los procesos con su consumo en CPU y en RAM, además del usuario que lo activó, usamos el comando top.

```
top - 09:22:43 up 3:23, 1 user, load average: 0,54, 0,23, 0,12
Tareas: 176 total, 1 ejecutar, 129 hibernar, 0 detener, 0 zombie
%Cpu(s): 0,1 usuario, 0,1 sist, 0,0 adecuado, 99,8 inact, 0,0 en espera, 0,0 hard
KiB Mem : 8148308 total, 6138532 libre, 636688 usado, 1373088 búfer/caché
KiB Intercambio: 969960 total, 969960 libre, 0 usado. 7214276 dispon Mem
```

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
5416	pru	20	0	44428	3924	3332	R	0,7	0,0	0:00.04	top
10	root	20	0	0	0	0	I	0,3	0,0	0:03.15	rcu_sched
972	pru	20	0	183784	2884	2524	S	0,3	0,0	1:22.46	VBoxClient
4520	root	20	0	0	0	0	I	0,3	0,0	0:00.38	kworker/3:1-eve
4988	root	20	0	0	0	0	I	0,3	0,0	0:00.14	kworker/1:11-ev
5358	pru	20	0	353560	25592	20628	S	0,3	0,3	0:00.98	lxtask
1	root	20	0	159468	8816	6760	S	0,0	0,1	0:01.91	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.03	kthreadd
3	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	rcu_par_gp
6	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	kworker/0:0H
8	root	0	-20	0	0	0	I	0,0	0,0	0:00.00	mm_percpu_wq
9	root	20	0	0	0	0	S	0,0	0,0	0:00.12	ksoftirqd/0
11	root	rt	0	0	0	0	S	0,0	0,0	0:00.62	migration/0
12	root	-51	0	0	0	0	S	0,0	0,0	0:00.00	idle_inject/0
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0,0	0,0	0:00.00	cpuhp/1

Aquí podemos filtrar por user. Si presionas la tecla “u” podrás escribir el nombre de usuario por el cual quieres buscar y si presionas la tecla “h” te mostrará un cuadro de ayuda para más opciones. Para salir presiona “q”.

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Matar un proceso (kill)

Para matar un proceso usamos el comando kill seguido del PID del proceso que queremos matar.

Por ejemplo, con el comando top, vemos que el proceso 7232 es el navegador Firefox:

PID	USUARIO	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	HORA+	ORDEN
7232	pru	20	0	3035544	268012	147896	S	10,3	3,3	0:06.69	firefox
708	root	20	0	997808	103836	42420	S	4,3	1,3	1:30.26	Xorg
4921	pru	20	0	1095156	39168	27288	S	1,7	0,5	0:05.69	lxterminal
972	pru	20	0	183784	2884	2524	S	0,7	0,0	1:25.89	VBoxClient
4963	pru	20	0	353520	25532	20556	S	0,7	0,3	0:04.28	lxtask
198	root	0	-20	0	0	0	I	0,3	0,0	0:00.99	kworker/0:1H-kb
1028	pru	20	0	397332	29328	23700	S	0,3	0,4	0:03.74	xfce4-power-man
1037	pru	20	0	721124	30544	24696	S	0,3	0,4	0:02.17	nm-applet
1045	pru	20	0	401248	22360	18088	S	0,3	0,3	0:02.43	light-locker
3854	root	20	0	0	0	0	I	0,3	0,0	0:00.72	kworker/u8:2-ev
5358	pru	20	0	353560	25680	20692	S	0,3	0,3	0:02.91	lxtask
5585	root	20	0	0	0	0	I	0,3	0,0	0:00.14	kworker/0:0-eve
6990	pru	20	0	353568	25648	20652	S	0,3	0,3	0:00.68	lxtask
1	root	20	0	159468	8820	6760	S	0,0	0,1	0:02.01	systemd
2	root	20	0	0	0	0	S	0,0	0,0	0:00.03	kthreadd

Para finalizar el proceso:

```
pru@pru-VirtualBox:~$ kill 7232
```

Actividad 03. Uso de la terminal y la línea de comandos en Linux

Htop

Htop es un visor de procesos y una aplicación de modo de texto para la supervisión del sistema en tiempo real, similar a top. Es fácil de usar y muestra una lista completa de los procesos que se ejecutan.

```
1  [                               0.0%]   Tasks: 106, 163 thr; 1 running
2  [                               0.0%]   Load average: 0.17 0.13 0.10
3  [|                             0.7%]   Uptime: 03:43:02
4  [|                             0.7%]
Mem[|||||||]                     392M/7.77G]
Swp[                               ]      0K/947M]
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
5358	pru	20	0	345M	25680	20692	S	0.7	0.3	0:04.54	lxtask
992	pru	20	0	179M	2884	2524	S	0.7	0.0	1:28.61	/usr/bin/VBoxClient --d
4963	pru	20	0	345M	25532	20556	S	0.7	0.3	0:05.92	lxtask
6990	pru	20	0	345M	25656	20652	S	0.7	0.3	0:02.29	lxtask
708	root	20	0	955M	82188	34160	S	0.0	1.0	1:39.49	/usr/lib/xorg/Xorg -cor
972	pru	20	0	179M	2884	2524	S	0.0	0.0	1:28.62	/usr/bin/VBoxClient --d
734	root	20	0	955M	82188	34160	S	0.0	1.0	0:20.99	/usr/lib/xorg/Xorg -cor
7561	pru	20	0	29800	4696	3420	R	0.0	0.1	0:00.95	htop
1014	pru	20	0	844M	46060	25932	S	0.0	0.6	0:09.00	lxpanel --profile Lubun
4921	pru	20	0	1090M	41388	28608	S	0.0	0.5	0:08.74	lxterminal
1037	pru	20	0	704M	30544	24696	S	0.0	0.4	0:02.37	nm-applet
1031	pru	20	0	329M	15416	13364	S	0.0	0.2	0:02.30	xfce4-power-manager
1028	pru	20	0	388M	29328	23700	S	0.0	0.4	0:04.00	xfce4-power-manager
1045	pru	20	0	391M	22360	18088	S	0.0	0.3	0:02.64	light-locker
1030	pru	20	0	558M	21684	17212	S	0.0	0.3	0:03.18	update-notifier
1128	pru	20	0	453M	29008	23940	S	0.0	0.4	0:02.67	/usr/lib/x86_64-linux-g
440	root	20	0	45244	5344	4804	S	0.0	0.1	0:00.28	/sbin/wpa_supplicant -u
1062	pru	20	0	49916	4104	3644	S	0.0	0.1	0:00.26	/usr/bin/dbus-daemon --
1010	pru	20	0	417M	35976	17560	S	0.0	0.4	0:02.27	openbox --config-file /
1076	pru	20	0	215M	6860	6156	S	0.0	0.1	0:02.15	/usr/lib/at-spi2-core/a
1	root	20	0	155M	8820	6760	S	0.0	0.1	0:02.01	/sbin/init splash

F1Help F2Setup F3SearchF4FilterF5Tree F6SortByF7Nice F8Nice F9Kill F10Quit