



Tecnicatura Universitaria en Software Libre



Administración de GNU/Linux I

Unidad 2
El sistema operativo

Autor
Leonardo Martinez

Contenido

1 El Sistema Operativo y su soporte físico	4
1.1 Particiones	4
1.1.1 Esquema de particionado MBR	5
1.1.2 Esquema de particionado GPT	6
1.1.3 <i>GPT</i> y gestores de arranque	6
1.1.4 Software de particionado	7
1.1.5 Modificación del particionado de un disco rígido	8
1.2 Sistemas de archivo	19
2 Directorios del sistema	20
2.1 Rutas y nombres de archivos	20
2.2 Estructura general de los directorios	21
2.3 Control de Acceso	24
2.3.1 Permisos de Archivos y Directorios	24
2.3.2 Cambiar los permisos de archivos y directorios	27
2.3.3 Cambiar dueño y grupo de archivos y directorios	28
3 Gestión de procesos	29
3.1 El Sistema de Archivos <code>/proc</code>	30
3.2 Herramientas	30
3.2.1 El comando <code>ps</code>	31
3.2.2 El comando <code>pstree</code>	33
3.2.3 El comando <code>kill</code>	33
3.2.4 El comando <code>top</code>	35
3.3 Demonios	35
3.3.1 Monitorear que hacen los demonios	36
4 Comandos Básicos	36
4.1 Ayuda sobre los comandos	37
4.2 Información de sistema, memoria, espacio en disco	38
4.2.1 El comando <code>uname</code>	38
4.2.2 El comando <code>uptime</code>	39
4.2.3 El comando <code>date</code>	40
4.2.4 El comando <code>df</code>	41
4.2.5 El comando <code>du</code>	42
4.2.6 El comando <code>free</code>	46
4.3 Navegación del árbol de directorios y gestión de archivos	47
4.3.1 Nombres de archivos y directorios	47
4.3.2 El comando <code>pwd</code>	47

4.3.3	El comando <code>ls</code>	47
4.3.4	El comando <code>cd</code>	51
4.3.5	El comando <code>mkdir</code>	52
4.3.6	El comando <code>rmdir</code>	53
4.3.7	El comando <code>cp</code>	53
4.3.8	El comando <code>mv</code>	54
4.3.9	El comando <code>rm</code>	55
4.3.10	El comando <code>locate</code>	56
4.4	Visualización y modificación de archivos de texto	58
4.4.1	El comando <code>file</code>	58
4.4.2	El comando <code>cat</code>	59
4.4.3	Los comandos <code>head</code> y <code>tail</code>	61
4.4.4	El comando <code>less</code>	62
4.4.5	El editor <code>vi</code>	62
5	Bibliografía	62

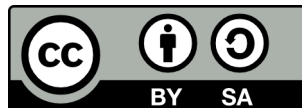
Copyright © 2016.

Autor: Leonardo Martinez

¡Copia este texto!

Los textos que componen este trabajo se publican bajo formas de licenciamiento que permiten la copia, la redistribución y la realización de obras derivadas, siempre y cuando éstas se distribuyan bajo las mismas licencias libres y se cite la fuente. El copyright de los textos individuales corresponde a los respectivos autores.

Este trabajo está licenciado bajo un esquema Creative Commons Atribución Compartir Igual (CC-BY-SA) 4.0 Internacional. <http://creativecommons.org/licenses/by-sa/4.0/deed.es>



1 El Sistema Operativo y su soporte físico

Todos los archivos del sistema operativo deben estar disponibles para su correcto funcionamiento, por lo tanto se necesita un dispositivo físico de soporte para darle existencia a los archivos. Existen diversos dispositivos de esta naturaleza, los más comunes son los discos rígidos mecánicos o de estado sólido, discos compactos (CDs/DVDs), memorias del tipo *flash*, etc.

Para que estos dispositivos puedan darle soporte y puedan contener los archivos del sistema operativo, se deben realizar una serie de configuraciones sobre los mismos.

1.1 Particiones

Las **particiones** son divisiones lógicas en la unidad de almacenamiento. Las características de estas divisiones lógicas dependen del **esquema de particiones** que se utilice. Los más conocidos y difundidos son **MBR** ¹ (*Master Boot Record*) y **GPT** ² (*GUID Partition Table*).

Prácticamente todo tipo de discos magnéticos y memorias flash (como pendrives) pueden particionarse. En el caso de los sistemas *UNIX* y *UNIX-like*, como es el caso de *GNU/Linux*, las particiones de datos son montadas en un directorio perteneciente a un único árbol jerárquico.

En Windows, las particiones reconocidas son identificadas con una letra seguida por dos puntos (por ejemplo, *C:*), aunque existe la posibilidad de montar particiones en directorios, no es una funcionalidad muy utilizada.

Crear más de una partición en un disco rígido trae beneficios:

- Separar los archivos del *sistema operativo* de los propios de los usuarios, permitiendo definir estrategias de respaldo (*backup*) separadas y dedicadas por cada tipo de servicio.
- Tener el espacio de la memoria virtual de intercambio (*área de swap*) o paginado separada de los archivos.
- Separar los archivos de *cache* y *logs*. Estos servicios crecen en forma dinámica y muy rápido. Si no se tiene cuidado se corre el riesgo de que se algún servicio del sistema que tenga algún problema genere muchos registros en los archivos de *logs* y, como resultado llene el disco provocando un fallo general.
- Tener un esquema de *multi-boot* con diferentes *sistemas operativos* en cada partición.

Las desventajas de utilizar varias particiones, a diferencia del modelo de partición única, son:

- Reducir el espacio total disponible para almacenamiento en el disco rígido, ya que obliga a tener áreas de administración en cada una de las particiones.
- Reducir el rendimiento general del disco rígido ya que al operar con archivos en diferentes particiones obliga al cabezal de lectura del disco a recorrer más distancia para leer las cabeceras de cada partición.
- Aumentar la fragmentación debido a que disminuye el tamaño promedio de los bloques libres continuos en cada partición.

- Mover archivos de una partición a otra requiere la copia de los bytes, en cambio si el movimiento de archivos es dentro de la misma partición sólo se requiere modificar la información **meta-data** en el registro.

1.1.1 Esquema de particionado MBR

Independientemente del *sistema de archivo* definido para una partición, existen tres *tipos de particiones* en el esquema **MBR**:

Partición primaria

Son las divisiones crudas o primarias del disco, se puede definir hasta un máximo de cuatro de éstas. Esta configuración se especifica en una **tabla de particiones**. Un disco rígido completamente formateado que se ve como un solo dispositivo desde el *sistema operativo* en realidad es una única partición primaria que ocupa todo el espacio disponible del disco y posee un sistema de archivo.

Partición extendida

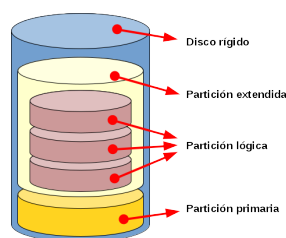
También conocida como *partición secundaria*, es un tipo de partición que funciona como una *partición primaria*; pero con una funcionalidad diferente, puede contener múltiples *unidades lógicas* en su interior. Fue diseñada con la idea de romper la limitación de cuatro *particiones primarias* en un solo disco rígido. Solo puede existir una partición de este tipo por disco, y su única tarea es contener *particiones lógicas*, siendo el único tipo de partición que no soporta un *sistema de archivo* directamente.

Partición lógica

Este tipo de partición puede ocupar una porción de la partición extendida en la que está definida o la totalidad del espacio disponible. Al ser formateada con un tipo específico de *sistema de archivo* el sistema operativo reconoce las *particiones lógicas* o su *sistema de archivo*. Por definición en la especificación técnica, puede haber un máximo de 32 *particiones lógicas* en una *partición extendida*. Sin embargo, algunos sistemas operativos restringen esa cantidad por cuestiones técnicas y/o rendimiento.

Podemos decir entonces que un dispositivo de almacenamiento con una tabla de particiones basada en el esquema **MBR** se le pueden definir **cuatro** particiones como máximo pudiendo ser todas *primarias* o un esquema de **tres primarias** y **una extendida**. Sabiendo que esta última puede contener varias *particiones lógicas*.

MBR almacena la información del sector de la partición con valores LBA (Logical Block Addressing) de 32 *bits*. Esta longitud de LBA junto con los 512 *bytes* del tamaño del sector limita el tamaño máximo del disco que una partición puede manejar en 2 *TiB* (1 *TiB* *tebibyte* = 2^{40} bytes).



*Ejemplo de particionamiento de un disco rígido con tabla de particiones **MBR** y con **una** partición primaria y **una** partición extendida conteniendo **tres** particiones lógicas.*

1.1.2 Esquema de particionado GPT

GUID Partition Table (GPT) es un nuevo formato de particionado que forma parte de la especificación *Unified Extensible Firmware Interface* ³ (UEFI), y utiliza un identificador único global para los dispositivos.

Resuelve una serie de limitaciones que presenta **MBR**, la cantidad máxima de particiones, la estructura de lista enlazada de los metadatos, el tamaño máximo de *2TiB* manejable por una partición.

El esquema de particionado **GPT** utiliza *GUID (UUID)* ⁴ para identificar los tipos de particiones evitando las colisiones. Al proporcionar un *GUID* único de disco y un *GUID* único para cada partición se obtiene un buen sistema de archivos independiente referenciando a las particiones y discos.

En este esquema de particionado no hay necesidad de particiones extendidas y lógicas. Por defecto, la tabla **GPT** tiene espacio para la definición de 128 particiones.

Al utilizar *64-bit* LBA, el tamaño máximo del disco manejable por una partición es de *2 ZiB* (1 ZiB *zebibyte* = 2^{70} bytes).

También se almacena una copia de seguridad del encabezado y de la tabla de particiones al final del disco que ayuda en la restauración del esquema en caso de daños.

1.1.3 GPT y gestores de arranque

Sistemas UEFI

GPT es parte de la especificación UEFI, por lo tanto el uso de este esquema de particionado es obligatorio.

Sistemas BIOS

- *GRUB* ⁵ requiere una partición de inicio *BIOS* de 2 *MiB* (1 *MiB mebibyte* = 2^{20} bytes), que no contenga sistema de archivos, para incrustar su archivo *core.img*. Esto se debe a que no hay espacio post-MBR para poder insertar discos GPT.
- *GRUB Legacy* y *LILO* no soportan **GPT**.

1.1.4 Software de particionado

En todos los sistemas operativos hay muchas aplicaciones que se pueden utilizar para realizar el particionado de los discos rígidos. Durante el proceso de instalación de **Xubuntu** se utiliza el software **GNU Parted CLI** que utiliza los parámetros que se ingresan durante el proceso de particionado.

La *Wikipedia* ⁶ nos ofrece un listado comparativo de software de particionado para los distintos tipos de esquemas y sistemas operativos.

Las más utilizadas son *fdisk*, *gdisk* ⁷ y *parted* ⁸.

Las aplicaciones de particionado se deben utilizar para gestionar discos rígidos que no están siendo utilizados por el sistema operativo o aplicaciones que están ejecutándose. Por lo tanto, si vamos a modificar el esquema de particiones del disco rígido donde tenemos el directorio raíz del sistema operativo, debemos utilizar un *LiveCD* que nos permita la manipulación de las particiones.

Uno de los *LiveCD* más utilizado para esta tarea es el *GNOME Partition Editor* ⁹, más conocido como *Gparted*. La imagen ISO del *LiveCD* se puede descargar desde la página del sitio ¹⁰ y la documentación sobre el funcionamiento y utilización de la herramienta se puede ver en la página de manual ¹¹.

El particionado es un paso indispensable en el proceso de instalación, consiste en dividir el espacio disponible en los discos rígidos según los datos que serán almacenados en él y el uso propuesto para el equipo.

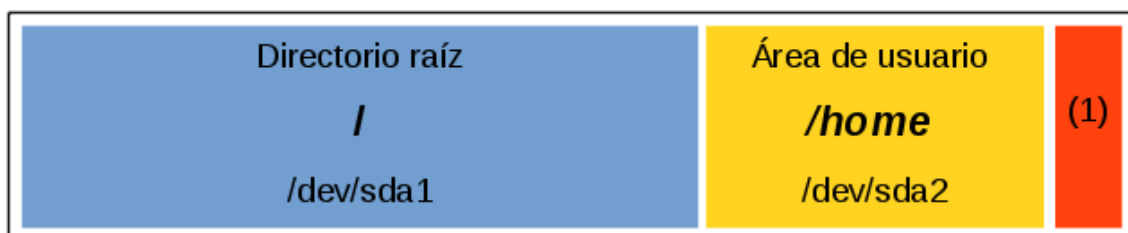
Este paso también incluye elegir los sistemas de archivo que serán utilizados. Todas estas decisiones influirán en el rendimiento, la seguridad de los datos y la administración del equipo.

El particionado es tradicionalmente difícil para usuarios nuevos. Se necesita definir las **particiones** en el que se almacenarán los sistemas de archivo de *GNU/Linux* y la memoria virtual (**swap**). Esta tarea es más complicada si el equipo ya posee otro sistema operativo que desea conservar. Hay que asegurarse de modificar la o las particiones existentes sin causar daños.

Durante el proceso de instalación de la máquina virtual que utilizaremos en la materia, se definió un esquema de particionado simple, separando el área de archivos de los usuarios (directorio */home*) de la partición que se usa para los archivos del sistema operativo.

El disco virtual de *15Gb* definido en la configuración de la *máquina virtual*, por cuestiones de técnicas que escapan al desarrollo de la materia, son presentados al sistema operativo como **16106 Mb**. Las particiones creadas en el proceso son tres:

- Partición primaria de *10000 Mb* identificada como **/dev/sda1** y montada en el directorio **/** (raíz) para el sistema operativo.
- Partición primaria de *5000 Mb* identificada como **/dev/sda2** y montada en el directorio **/home** para el área de archivos de usuarios.
- Partición primaria de *1106 Mb* identificada como **/dev/sda3** y como área de intercambio o *swap*.



(1) Área de intercambio o **swap** (/dev/sda3)

Particionamiento del disco rígido antes de la modificación.

1.1.5 Modificación del particionado de un disco rígido

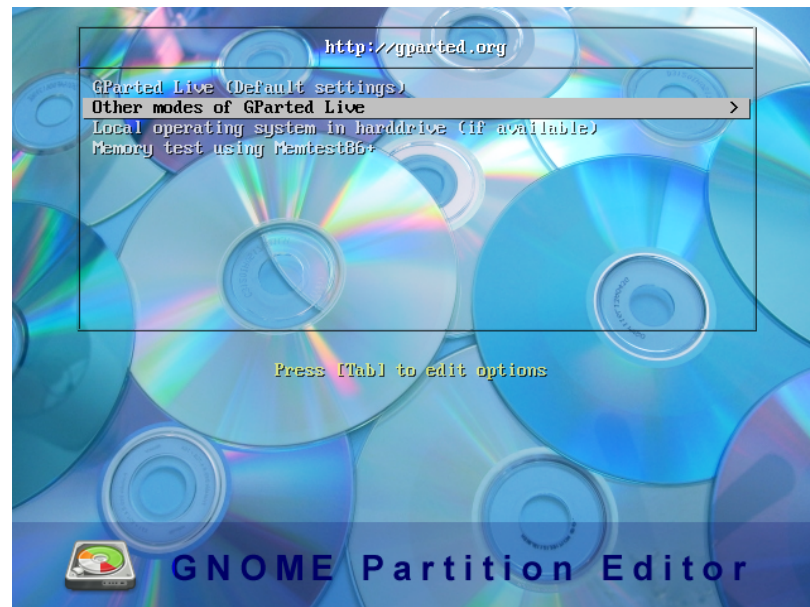
Lo que haremos es achicar la partición del área de usuarios **/home** (/dev/sda2) y crear una cuarta partición primaria para **datos**.



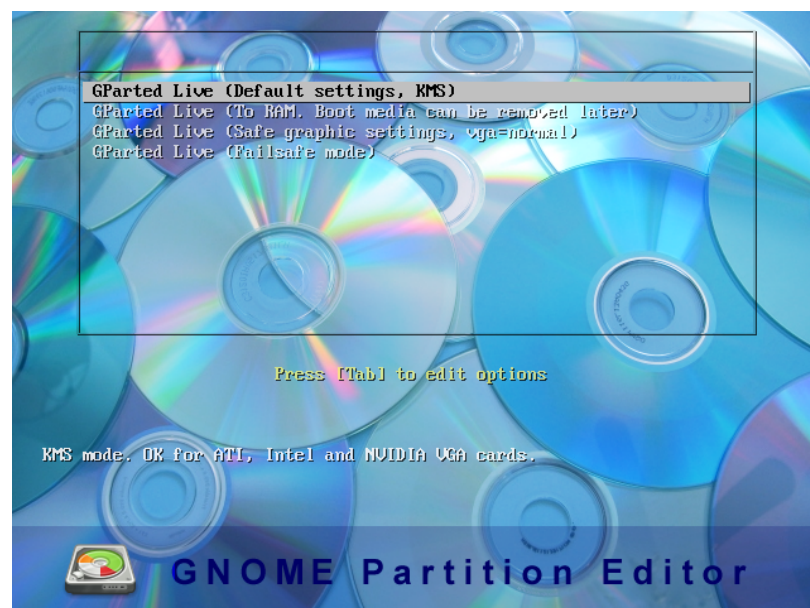
(1) Área de intercambio o **swap** (/dev/sda3)

Particionamiento del disco rígido después de la modificación.

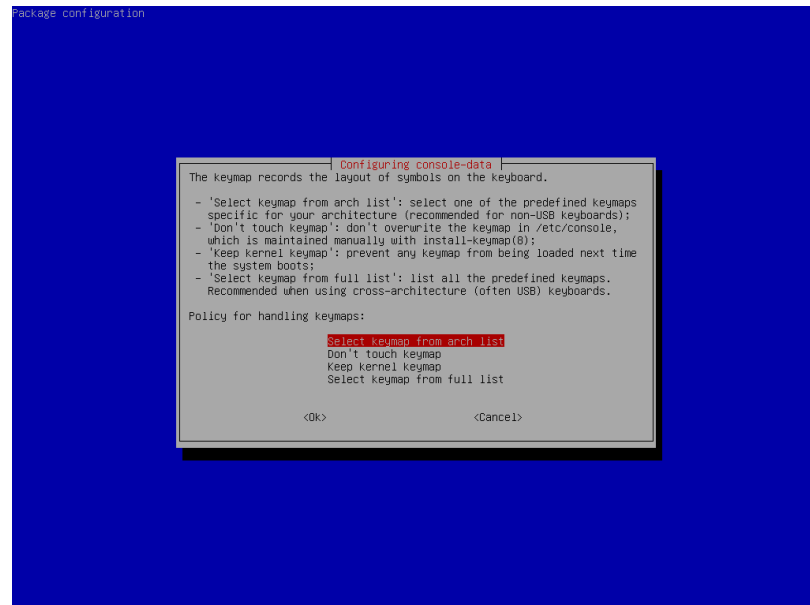
El primer paso será asignar el archivo ISO de *GParted* descargado a la unidad de CD/DVD de la máquina virtual e iniciarla. Por un problema con *VirtualBox* no se puede utilizar el inicio estándar de *GParted LiveCD*, por lo tanto, en la primer pantalla se debe seleccionar la opción **Other modes of GParted Live**.



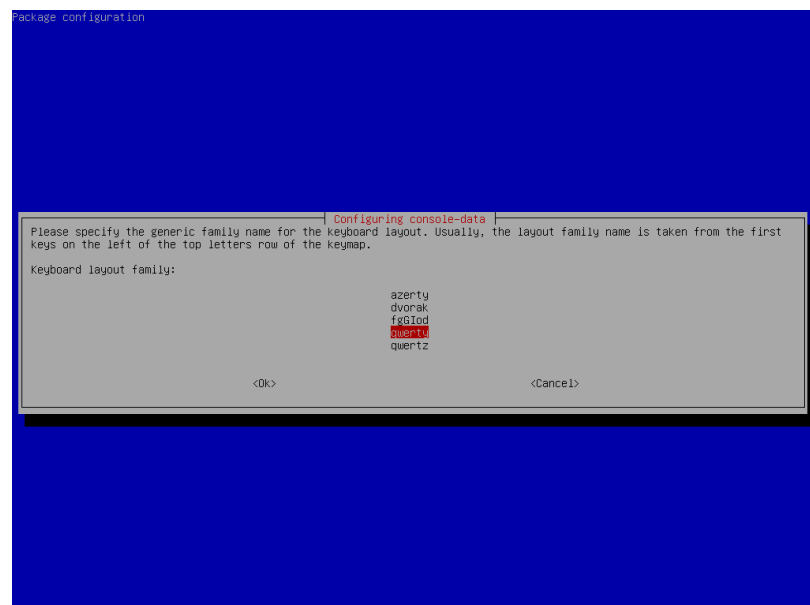
En la siguiente pantalla seleccionar la opción **GParted Live (Default settings, KMS)**



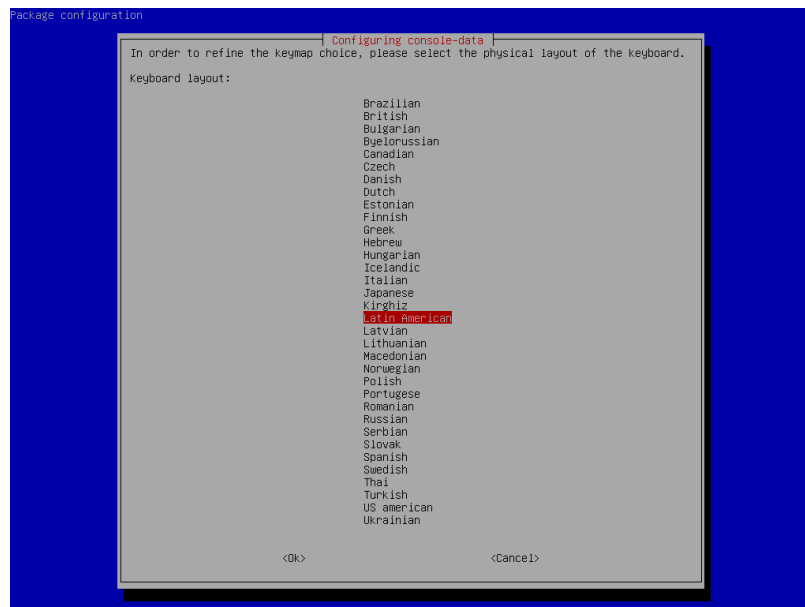
Una vez iniciado, en las pantallas de configuración de **GParted LiveCD** se define el tipo de teclado que se utilizará seleccionando **Select keyboard from arch list**.



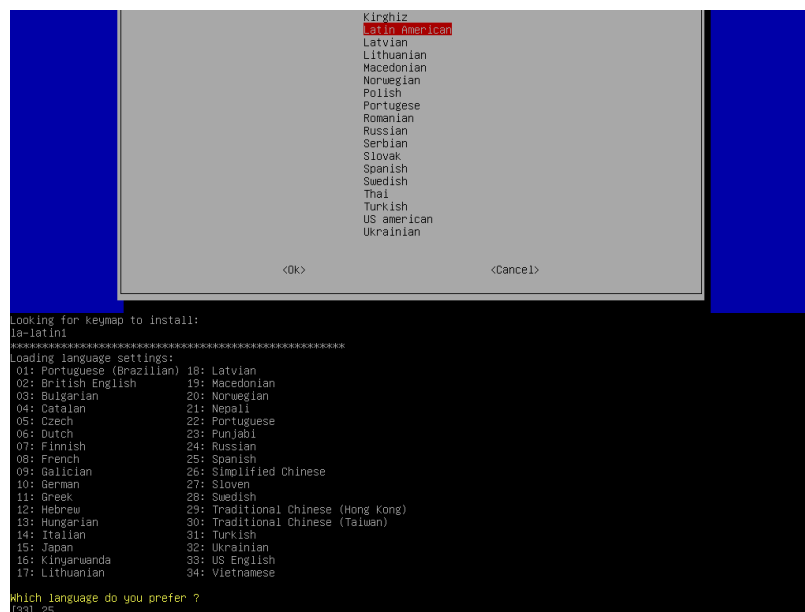
Luego se elige **qwerty** como la familia de disposición de teclas.



El tipo de teclado variará según el que disponga en la computadora física, en este caso se opta por el modelo **Latin American**.



A continuación se muestra el listado de idiomas disponibles para el sistema operativo de **GParted LiveCD**. La opción **25** corresponde al *español*.



Luego se selecciona el modo *[0]* para iniciar **GParted** en el entorno gráfico.

```

Ukrainian
<Ok> <Cancel>

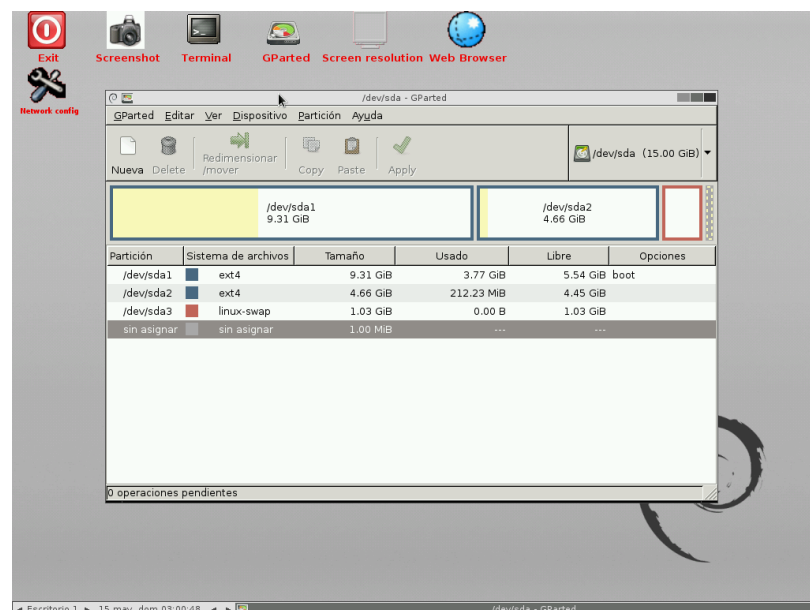
Looking for keymap to install:
la-latin1
*****
Loading language settings:
01: Portuguese (Brazilian) 16: Latvian
02: British English 19: Macedonian
03: Bulgarian 20: Norwegian
04: Catalan 21: Nepali
05: Czech 22: Portuguese
06: Dutch 23: Punjabi
07: Finnish 24: Russian
08: French 25: Spanish
09: Galician 26: Simplified Chinese
10: German 27: Slovene
11: Greek 28: Swedish
12: Hebrew 29: Traditional Chinese (Hong Kong)
13: Hungarian 30: Traditional Chinese (Taiwan)
14: Italian 31: Turkish
15: Japanese 32: Ukrainian
16: Kinyarwanda 33: US English
17: Lithuanian 34: Vietnamese

Which language do you prefer ?
[33] 25
Language selected es_ES
Generating es_ES locale by "localedef -f UTF-8 -i es_ES es_ES.UTF-8"... done!
Setting locale in /etc/default/locale...
done!
*****
//NOTE// Later we will enter graphical environment if you choose '0'. However, if graphical environment (X-window) fails to start, you can:
Run "sudo Forcevideo" to configure it again. Choose 1024x768, 800x600 or 640x480 as your resolution and the driver for your VGA card, etc. Most of the time you can accept the default values if you have no idea about them.
If failing to enter graphical environment, and it does not return to text mode, you can reboot again, and choose '1' here to config X manually.
-----
Which mode do you prefer ?
(0) Continue to start X to use GParted automatically
(1) Run "Forcevideo" to config X manually
(2) Enter command line prompt
[0]

```

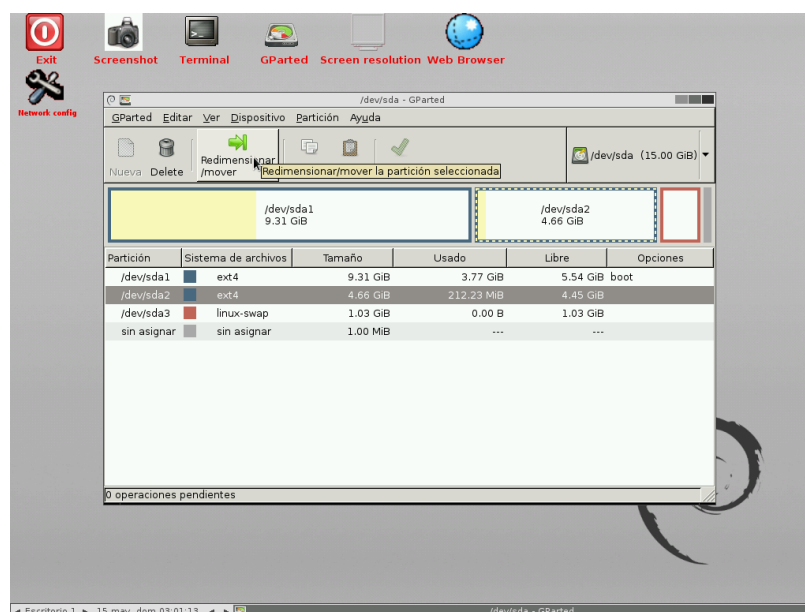
Una vez iniciado, vemos que estamos en uno de los tantos entornos gráficos disponibles para *GNU/Linux*, en este caso **Fluxbox**, de características más simples ya que solo necesitamos utilizar unas pocas herramientas. Incluso ya aparece iniciada la aplicación *GParted*. Esta nos muestra, de manera clara, el esquema de particionado del disco del equipo.

En cada uno de los recuadros que representan las particiones se observan dos porciones: la del extremo derecho es la estimación de lo que representa el espacio en uso, la porción de la derecha es el espacio disponible, que no contiene datos.

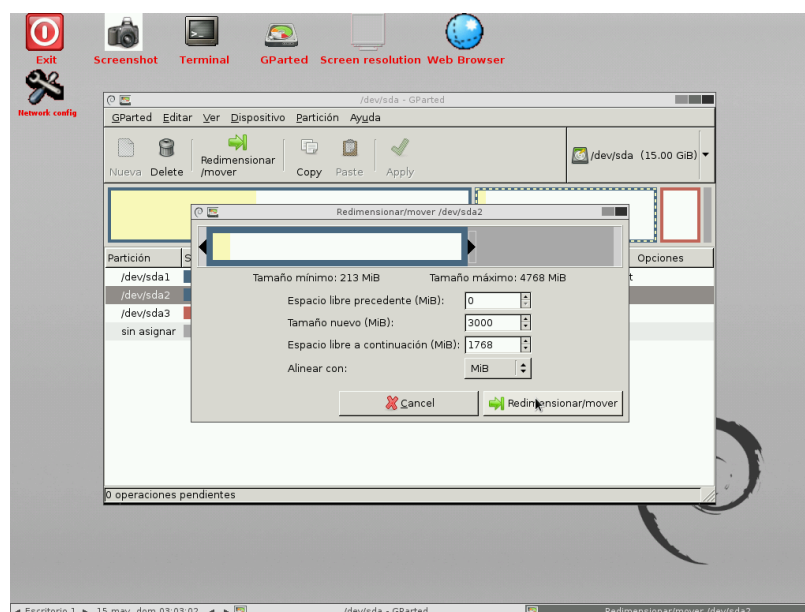


Para mostrar el funcionamiento de la aplicación vamos a achicar la partición `/dev/sda2` montada en el directorio `/home` y crear una nueva partición que estará montada en el directorio `/datos`.

El primer paso es seleccionar la partición que vamos a modificar y presionar el botón **Redimensionar/mover**.

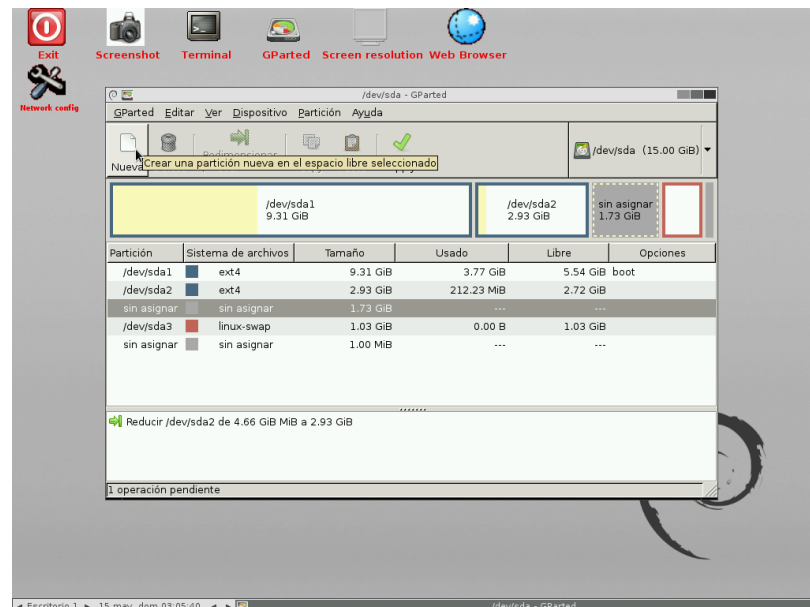


En la pantalla de modificación, la tarea se puede realizar de dos maneras, editando el campo *Tamaño Nuevo (MB)* ingresando el nuevo valor, o más dinámicamente desplazando con el mouse la flecha de la derecha hasta la posición requerida.

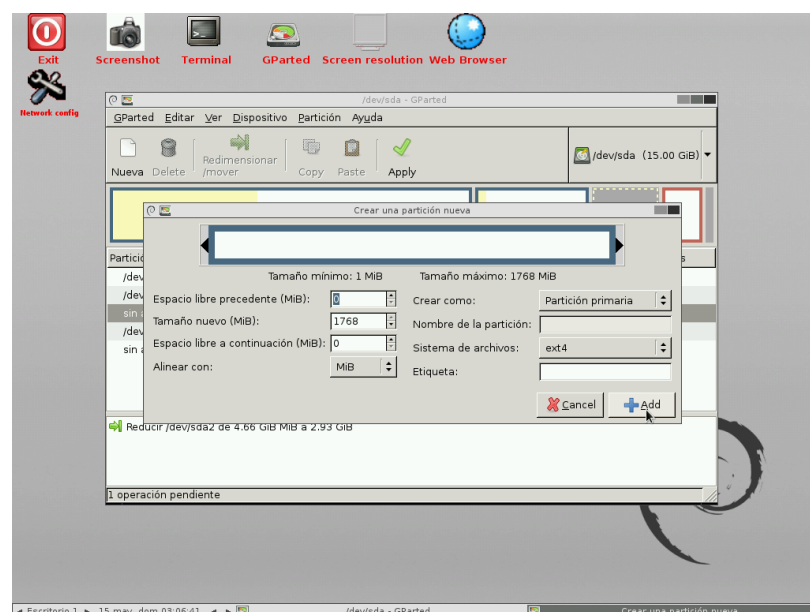


Luego de realizada la tarea vemos que la partición está definida con el nuevo tamaño y existe un espacio con la denominación *Sin asignar*. La operación de redimensionamiento, en realidad, todavía no se hizo efectiva, en la parte inferior de la pantalla de la aplicación, vemos que figura una operación pendiente.

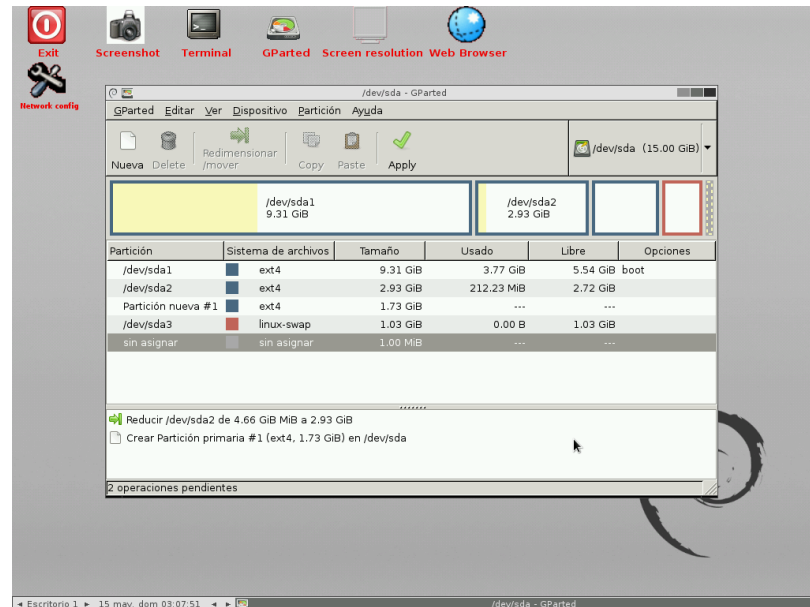
Continuamos con nuestro objetivo seleccionando el espacio *Sin asignar* y presionando el botón **Nueva**.



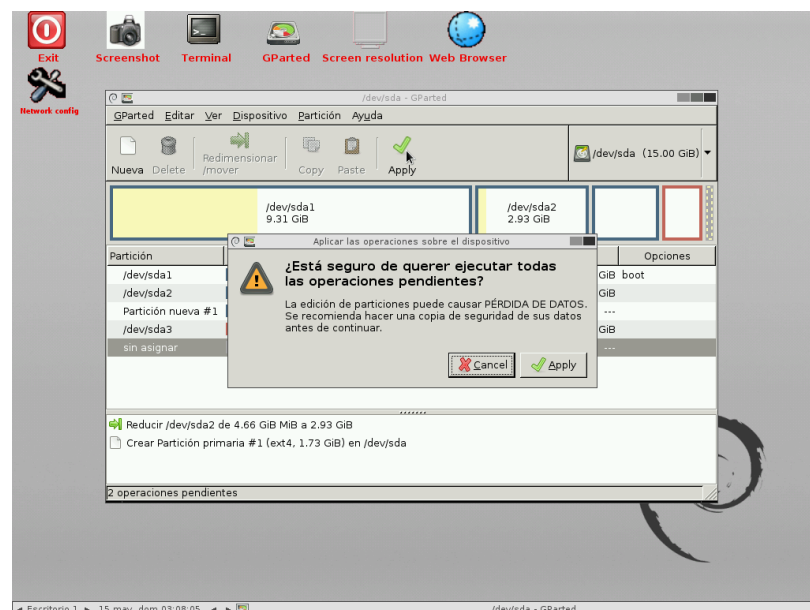
En la pantalla de definición de la nueva partición marcamos todo el espacio disponible, seleccionamos que la partición será del tipo *primaria* y tendrá un sistema de archivo *ext4*.



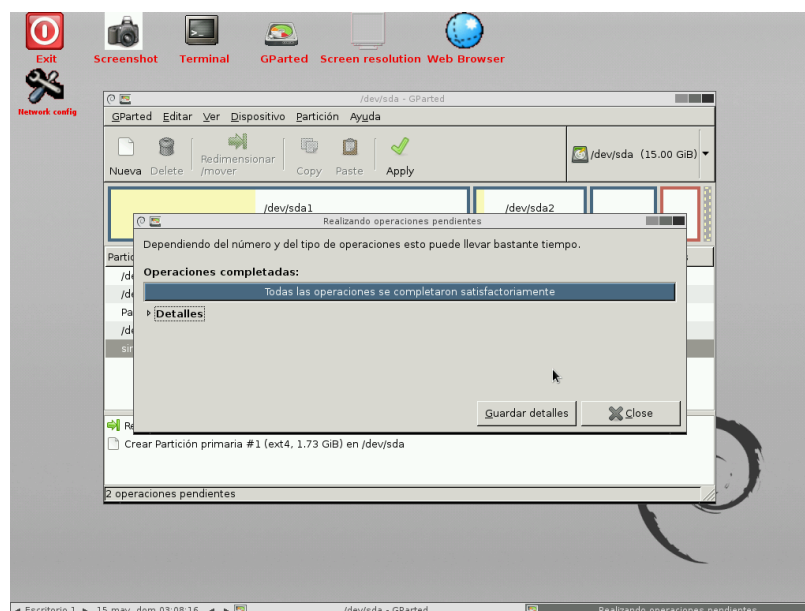
Con las dos operaciones pendientes, de redimensionamiento y de creación de la nueva partición, vemos cómo quedará el disco rígido.



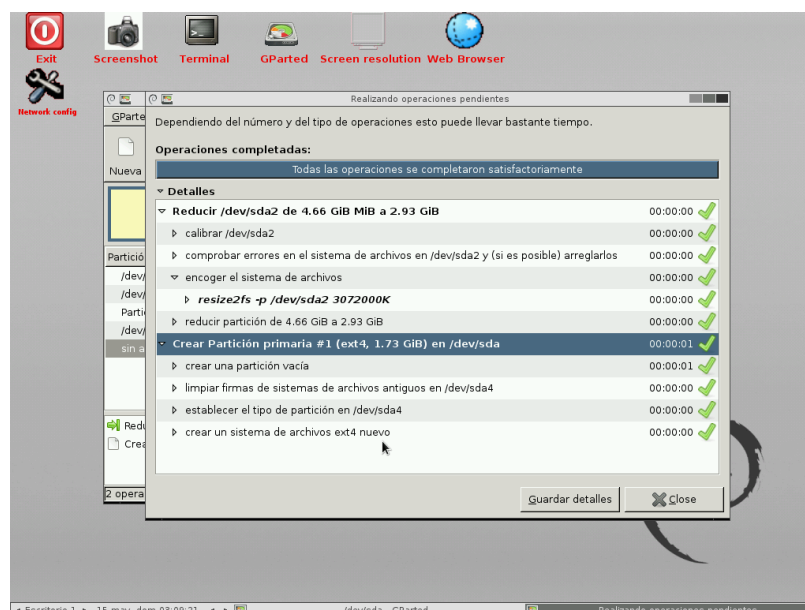
Para hacer efectivos los cambios, se debe presionar el botón **Apply**. La aplicación hará la correspondiente advertencia ya que estaremos modificando el esquema de la tabla de particiones del disco.



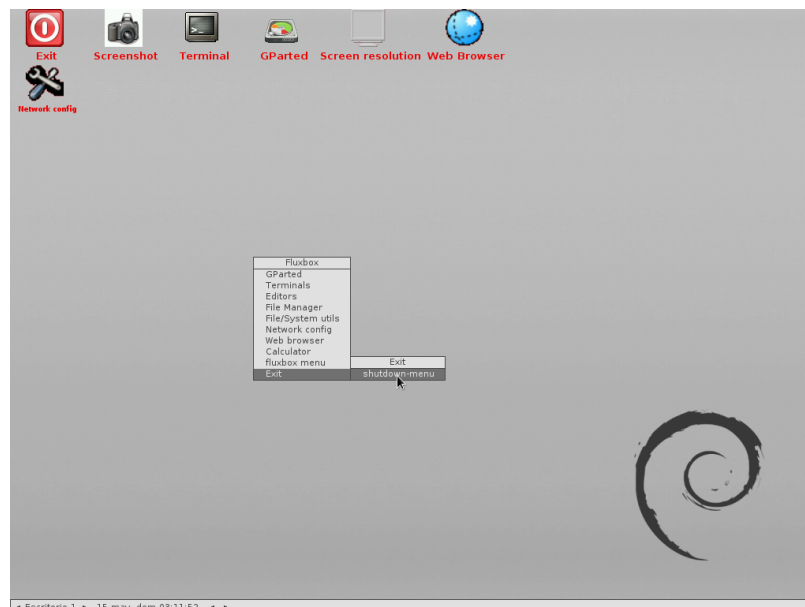
El mensaje que esperamos ver luego de realizar tareas de este tipo: *Todas las operaciones se completaron satisfactoriamente*. Para ver más información sobre las tareas realizadas se debe presionar sobre la palabra **Detalles**.



Los detalles de los procesos realizados nos permite ver, más minuciosamente, lo que la aplicación realiza. Por ejemplo, en la pantalla siguiente se puede ver claramente el comando utilizado para encoger la partición `/dev/sda2`.



Una vez cerrada la aplicación, debemos apagar el equipo, para ello se debe presionar con el botón derecho del mouse sobre el escritorio, luego seleccionar *Exit* y *shutdown-menu*.

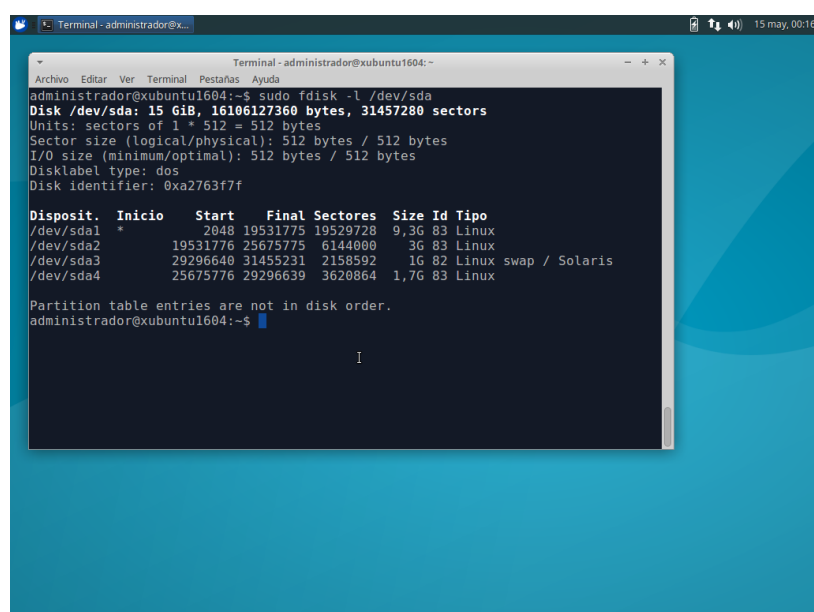


Con el equipo apagado, se quita la imagen ISO del **GParted LiveCD** de la unidad virtual.

Para ver cómo ve el sistema operativo la nueva configuración del disco rígido, iniciamos la *máquina virtual* y abrimos la aplicación *Terminal*.

Ejecutamos el comando:

```
$ sudo fdisk -l /dev/sda
```

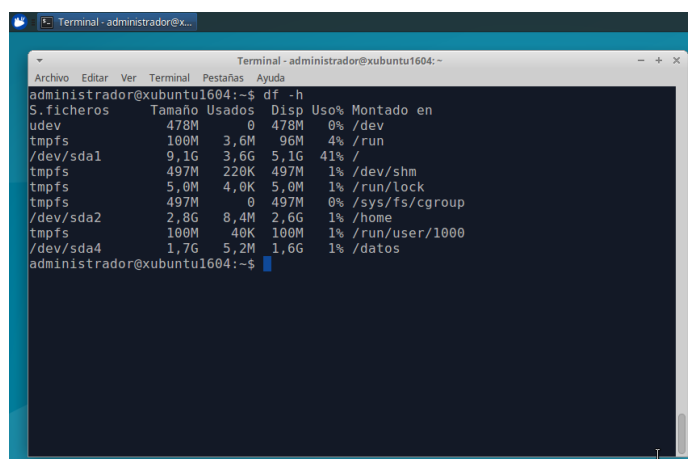


Para tener disponible el espacio de la nueva partición, debemos realizar una serie de pasos que veremos en detalle en una próxima unidad de la materia.

Por ahora basta con ejecutar los comandos en la consola y una breve modificación en un archivo de configuración.

Ejecutar los siguientes comandos en una terminal:

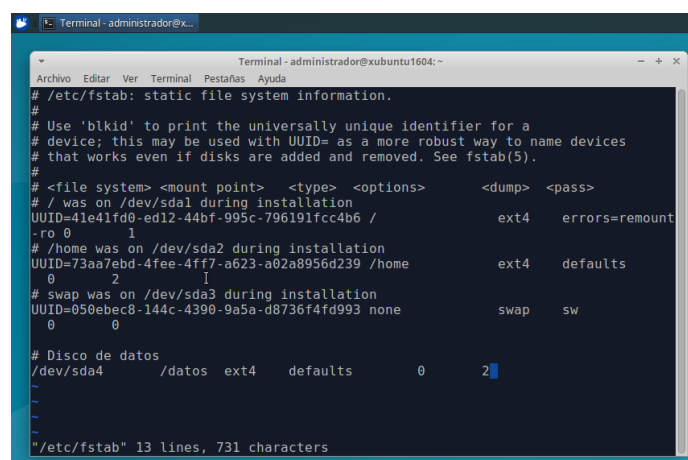
```
$ sudo mkdir /datos
$ sudo mount /dev/sda4 /datos
```



```
administrador@xubuntu1604:~$ df -h
S.ficheros      Tamaño Usados  Disp Uso% Montado en
udev            478M    0    478M   0% /dev
tmpfs           100M    3,6M   96M   4% /run
/dev/sda1        9,1G    3,6G   5,1G  41% /
tmpfs           497M    220K  497M   1% /dev/shm
tmpfs           5,0M    4,0K   5,0M   1% /run/lock
tmpfs           497M    0    497M   0% /sys/fs/cgroup
/dev/sda2        2,8G    8,4M   2,6G   1% /home
tmpfs           100M    40K   100M   1% /run/user/1000
/dev/sda4        1,7G    5,2M   1,6G   1% /datos
administrador@xubuntu1604:~$
```

Para que el montaje de la nueva partición sea persistente y esté disponible cuando se inicia el sistema operativo editamos el archivo `/etc/fstab` agregando las siguientes líneas al final del archivo:

```
# Disco de datos
/dev/sda4 /datos ext4 defaults 0 2
```



```
administrador@xubuntu1604:~$ cat /etc/fstab
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda1 during installation
UUID=41e41fd0-ed12-44bf-995c-796191fcc4b6 / ext4 errors=remount-ro 0 1
# /home was on /dev/sda2 during installation
UUID=73aa7ebd-4fee-4ff7-a623-a02a8956d239 /home ext4 defaults 0 2
# swap was on /dev/sda3 during installation
UUID=050ebec8-144c-4390-9a5a-d8736f4fd993 none swap sw 0 0
# Disco de datos
/dev/sda4 /datos ext4 defaults 0 2
" /etc/fstab" 13 lines, 731 characters
```

1.2 Sistemas de archivo

Cada *partición* es el soporte para poder contener datos, tanto los archivos del sistema operativo como los que utilizan y generan los usuarios, pero para ello hay que definir que tipo de **sistema de archivo** tendrán. El espacio del disco rígido que no se asigna, no conforma una *partición* por lo tanto no puede contener un **sistema de archivo**.

Un **sistema de archivo** ^{12 13} define la forma en la que se organizan los datos en el disco rígido. Entre los tipos de **sistema de archivo**, algunos de los más conocidos son *FAT*, *NTFS*, *FAT32*, *EXT2*, *EXT3*, *EXT4*, *XFS*, *Btrfs*, *ReiserFS* cada uno de los cuales tiene sus características y capacidades particulares y se pueden optar por cualquiera de ellos de acuerdo a las necesidades de uso. Incluso se puede definir un sistema de archivo diferente para cada partición de un disco rígido.

Algunos son más robustos, otros más efectivos, *ReiserFS* es particularmente eficiente para leer muchos archivos pequeños; *XFS*, en cambio, trabaja más rápido con archivos grandes. *ext4*, el sistema de archivo predeterminado para *Debian*, es un buen compromiso basado en las tres versiones anteriores de sistemas de archivo utilizados en *GNU/Linux* históricamente (*ext*, *ext2* y *ext3*) ya que supera algunas limitaciones de *ext3* y es particularmente apropiado para discos rígidos de gran capacidad.

Un sistema de archivo con *registros* (como *ext3*, *ext4*, *btrfs*, *reiserfs* o *xfs*) toma medidas especiales que posibilitan volver a un estado consistente anterior luego de una interrupción abrupta sin analizar completamente el disco entero, como era necesario con el sistema *ext2*. Esta funcionalidad se lleva a cabo manteniendo un registro que describe las operaciones a realizar antes que sean ejecutadas. Si se interrumpe una operación será posible *reproducirla* desde el *registro*. Por el otro lado, si la interrupción ocurre durante una actualización del registro, simplemente se ignora el último cambio solicitado; los datos almacenados podrían perderse pero, como los datos en el disco no han cambiado, se mantuvieron coherentes. Esto es nada más y nada menos que un mecanismo transaccional aplicado al sistema de archivo.

Todo *sistema operativo* necesita guardar multitud de archivos: configuración del sistema, registro de actividades, de usuarios, etc. Existen diferentes **sistemas de archivo** caracterizados por su estructura, fiabilidad, arquitectura y rendimiento. *GNU/Linux* es capaz de leer y escribir archivos en la casi totalidad de los **sistemas de archivo** aunque tiene su propio sistema optimizado para sus funcionalidades, como por ejemplo *ext4* o *ReiserFS*.

El tipo de **sistema de archivo** *ext4* es una mejora compatible con *ext3* que a su vez es una evolución del *ext2* que es el más típico y extendido. Su rendimiento es muy bueno, incorpora todo tipo de mecanismos de seguridad y adaptación, es muy fiable e incorpora una tecnología denominada *journaling* (registro), que permite recuperar fácilmente errores en el sistema cuando, por ejemplo, hay un corte de energía o el equipo sufre una parada no prevista. *Ext4* soporta volúmenes de hasta 1024 PiB (1 PiB *pebibyte* = 2^{50} bytes), mejora el uso de CPU y el tiempo de lectura y escritura.

ReiserFS es otro de los sistemas utilizados en *GNU/Linux* que incorpora nuevas tecnologías de diseño y que le permiten obtener mejores prestaciones y utilización del espacio libre. Cualquiera de estos tipos de **sistemas de archivo** puede ser seleccionado en el proceso de instalación, sin embargo, es recomendable utilizar *ext4* para la mayoría de las instalaciones.

Para ver el tipo de **sistema de archivo** que tiene cada partición del sistema, se puede ejecutar el comando:

```
$ df -Th
S.ficheros      Tipo      Tamaño Usados  Disp  Uso% Montado en
/dev/sdal       ext4       19G    5,7G   12G   33% /
udev            devtmpfs   10M      0    10M    0% /dev
tmpfs           tmpfs      605M    8,4M   596M   2% /run
tmpfs           tmpfs      1,5G    740K   1,5G   1% /dev/shm
tmpfs           tmpfs      5,0M    4,0K   5,0M   1% /run/lock
tmpfs           tmpfs      1,5G      0    1,5G   0% /sys/fs/cgroup
none            vboxsf     96G    88G    8,5G  92% /media/sf_Downloads
tmpfs           tmpfs      303M    8,0K   303M   1% /run/user/119
tmpfs           tmpfs      303M    16K    303M   1% /run/user/1000
```

2 Directorios del sistema

El **Estándar de Jerarquía del Sistema de archivos** (o *FHS*, del inglés *Filesystem Hierarchy Standard*) es una norma que define los directorios principales y sus contenidos en el sistema operativo *GNU/Linux* y otros sistemas de la familia Unix. Se diseñó originalmente en 1994 para estandarizar el sistema de archivos de las distribuciones de *GNU/Linux*, basándose en la tradicional organización de directorios de los sistemas Unix. En 1995 se amplió el ámbito del estándar a cualquier Unix que se adhirió voluntariamente.

Los sistemas Unix en general, y por lo tanto también *GNU/Linux*, presentan una estructura de archivos estándar que, a diferencia de otros sistemas operativos, considera a todos los componentes un archivo, logrando una independencia del dispositivo físico. El *kernel* es un conjunto de archivos, las librerías son archivos, el directorio es un archivo y los dispositivos son archivos. Estos últimos, por esta misma condición, no necesitan una forma particular de identificarlos como es el caso de *DOS* y *Windows*, que utilizan una letra.

Esta versatilidad de direccionar todo en una jerarquía de archivos permite incorporar a la estructura de archivos los dispositivos de almacenamiento removibles e incluso recursos de archivos remotos en otros sistemas.

El **sistema de archivo** *ext2/3/4* fue diseñado para manejar de forma óptima archivos pequeños y de forma aceptable los ficheros grandes, si bien se pueden configurar los parámetros del **sistema de archivo** para optimizar el trabajo con este tipo de archivos.

La jerarquía del sistema de archivos de *GNU/Linux* es similar a la estructura de la raíz de un árbol. Comienza con un directorio particular denominado *directorio root* o *directorio raíz* y se ramifica en una estructura definida.

2.1 Rutas y nombres de archivos

En los sistemas de archivos jerárquicos, usualmente, se declara la ubicación precisa de un archivo con una cadena de texto llamada **ruta** (*path*, en inglés). La nomenclatura para rutas varía ligeramente de sistema en sistema, pero mantienen por lo general una misma estructura. Una ruta viene dada por una sucesión de nombres de directorios y subdirectorios, ordenados jerárquicamente de izquierda a derecha y separados por algún carácter especial

que en el caso de los sistemas operativos del tipo *Unix*, y por supuesto en *GNU/Linux*, es una barra diagonal (/) y puede terminar en el nombre de un archivo presente en la última rama de directorios especificada.

Por ejemplo, la ruta para el archivo *alumnos.txt* ubicado en el directorio *Documentos* del usuario *lmartinez* se podría describir de la siguiente manera:

```
/home/lmartinez/Documentos/alumnos.txt
```

Donde / representa el directorio raíz donde está montado el sistema de archivos, y contiene a toda la jerarquía de directorios. La cadena de texto `/home/lmartinez/Documentos/` representa la ruta de directorios en la estructura jerárquica y la cadena `alumnos.txt` el nombre del archivo.

Los nombres de los archivos son a criterio de los usuarios y tienen pocas restricciones, no es necesario la utilización de extensiones, aunque se utilizan convencionalmente para una mejor comprensión acerca del contenido de los archivos y la asociación con aplicaciones y el tratamiento que éstas le dan a los mismos, desde la visualización del listado de archivos en la consola o en el explorador de archivos, hasta los editores de texto y planillas de cálculo.

Restricciones para los nombres de archivos

Característica	Observación
Bytes máx.	255
Bytes permitidos	Todos excepto el carácter nulo <i>NUL</i> \0, la / y los nombres de archivos especiales . y ..

Los nombres especiales . y .. son referencias relativas al directorio en el que se está ubicado. El . hace referencia al directorio actual y .. hace referencia al directorio inmediato anterior en la jerarquía o *ruta*.

En el ejemplo, al posicionarnos en el directorio donde está alojado el archivo `alumnos.txt` el nombre especial . hace referencia al directorio `Documentos` y el nombre especial .. hace referencia al directorio `lmartinez`.

2.2 Estructura general de los directorios

En el sistema de ficheros de UNIX (y similares), existen varias sub-jerarquías de directorios que poseen múltiples y diferentes funciones de almacenamiento y organización en todo el sistema. Estos directorios pueden clasificarse en:

Estáticos

Contiene archivos que no cambian sin la intervención del administrador (usuario `root`), sin embargo, pueden ser leídos por cualquier otro usuario. (por ejemplo, `/bin`, `/sbin`, `/opt`, `/boot`, `/usr/bin`)

Dinámicos

Contiene archivos que son cambiantes, y pueden leerse y escribirse (algunos sólo por el usuario dueño y el superusuario `root`). Contienen configuraciones, documentos, etc. Para estos directorios, es recomendable una copia de seguridad con frecuencia, o mejor aún, deberían ser montados en una partición aparte en el mismo disco, como por ejemplo, montar el directorio `/home` en otra partición del mismo disco, independiente de la partición principal del sistema; de esta forma, puede repararse el sistema sin afectar o borrar los documentos de los usuarios. (por ejemplo, `/var/mail`, `/var/spool`, `/var/run`, `/var/lock`, `/home`)

Compartidos

Contiene archivos que se pueden encontrar en un ordenador y utilizarse en otro, o incluso compartirse entre usuarios.

Restringidos

Contiene ficheros que no se pueden compartir, solo son modificables por el administrador. (por ejemplo, `/etc`, `/boot`, `/var/run`, `/var/lock`)

En la mayoría de las distribuciones *GNU/Linux* se sigue el estándar **FHS** ^{14 15}, por lo tanto, en el primer nivel de la jerarquía de directorios tenemos los siguientes:

Directorio	Descripción
<code>/</code>	Jerarquía primaria, la <i>raíz</i> o <i>root</i> , y directorio raíz o contenedor de todo el sistema de jerarquía.
<code>/bin/</code>	Aplicaciones binarias de comando que son esenciales para que estén disponibles para una sesión de usuario único, o bien, para todos los usuarios (multiusuario).
<code>/boot/</code>	Archivos cargadores de arranque (por ejemplo, los núcleos, el <i>initrd</i>). A menudo en una partición o disco aparte.
<code>/dev/</code>	Contiene los enlaces a dispositivos esenciales (por ejemplo, <code>/dev/null</code>), incluso a los que no se les ha asignado un directorio. También contiene los enlaces a dispositivos que son virtuales y también a los que no proporcionan almacenamiento (micrófonos, impresoras, etc). Se trata de la parte de más bajo nivel del sistema operativo hacia el hardware, aunque es extremadamente útil para tener un acceso directo a los dispositivos.
<code>/etc/</code>	Archivos de configuración del sistema y de las aplicaciones que hay instaladas.
<code>/home/</code>	Contiene los directorios de trabajo de todos los usuarios, excepto el del superusuario root . Contiene archivos guardados por el usuario y ajustes personales. A menudo se instala en un disco o partición separada. Cada usuario tiene su propio directorio dentro de esta carpeta.

Directorio	Descripción
<code>/lib/</code>	Contiene todas las bibliotecas (mal traducidas como librerías) esenciales compartidas de los programas alojados, es decir, los binarios en <code>/bin/</code> y <code>/sbin/</code> . Contiene también las bibliotecas para el núcleo.
<code>/media/</code>	Contiene los puntos de montaje de medios de almacenamiento extraíbles, tales como lectores de CD-ROM, pendrives (memorias USB), e incluso sirve para montar otras particiones del mismo disco rígido utilizadas por otro sistema operativo.
<code>/mnt/</code>	Es una directorio semejante a <code>/media</code> . Sirve para montar discos rígidos y particiones de forma temporal en el sistema; no necesita contraseña, a diferencia del directorio <code>/media</code> .
<code>/opt/</code>	Contiene aplicaciones estáticas opcionales que pueden ser compartidas entre los usuarios. Dichas aplicaciones no guardan sus configuraciones en este directorio, cada usuario puede tener una configuración diferente de una misma aplicación, de manera que se comparte la aplicación pero no las configuraciones de los usuarios, las cuales se guardan en su respectivo directorio en <code>/home</code> .
<code>/proc/</code>	Sistema de archivos virtual que contiene principalmente archivos de texto, que documentan al núcleo y el estado de los procesos en archivos de texto para consulta por parte de los procesos.
<code>/root/</code>	Directorio personal del usuario root . Funciona como las carpetas en <code>/home</code> , pero en este caso, es solo para el superusuario.
<code>/sbin/</code>	Sistema de binarios esenciales, comandos y programas exclusivos del superusuario, (por ejemplo, <code>init</code> , <code>route</code> , <code>ifup</code>). Un usuario puede ejecutar estas aplicaciones de comandos si tiene los permisos suficientes.
<code>/srv/</code>	Lugar específico para los datos que son servidos por el sistema, como ser los datos y <i>scripts</i> de servidores web, datos ofrecidos por el servicio FTP o repositorios de los sistemas de control de versión.
<code>/sys/</code>	Evolución de <code>/proc/</code> . Sistema de archivos virtual que documenta al núcleo pero localizados de forma jerarquizada. En <code>/proc/</code> se disponen de forma anárquica. Su nombre correcto es <i>Sysfs</i> .
<code>/tmp/</code>	Archivos temporales. Aquí se guardan los archivos temporales guardados por las aplicaciones del sistemas.
<code>/usr/</code>	Jerarquía secundaria de los datos de usuario; contiene la mayoría de las utilidades y aplicaciones multiusuario, es decir, accesibles para todos los usuarios. Contiene los archivos compartidos, pero que no obstante son de sólo lectura. Este directorio puede incluso ser compartido con otras computadoras de red local.

Directorio	Descripción
/var/	Archivos variables, tales como logs, archivos spool, bases de datos, archivos de e-mail temporales, y algunos archivos temporales en general. Generalmente actúa como un registro del sistema. Contiene la información que ayuda a encontrar los orígenes de un problema.

No se deben borrar estos directorios a pesar de que parezca que no se utilizan, son necesarios para el buen funcionamiento del sistema y muchas aplicaciones pueden no instalarse o dar errores si los directorios estándar no se encuentran definidos.

2.3 Control de Acceso

Al considerar la accesibilidad de un archivo por parte de un usuario se deben tener en cuenta los tipos de nivel de usuario y los tipos de permisos.

Utilizando los atributos de propiedad y de grupo, en combinación con el *UID* de los usuarios y el *GID* los grupos, se puede asignar una combinación de permisos posibles para cada uno de los archivos en el sistema de archivos.

Si el usuario que quiere acceder al archivo es el dueño del mismo se utiliza el nivel de permisos de dueño (*owner*). En cambio, si el usuario no es el dueño pero esta en el grupo asociado al archivo, se evalúa el nivel de grupo (*group*), si el usuario no es el propietario y tampoco está en el grupo asociado, se determina la accesibilidad contemplando el nivel denominado *otros* (*others*).

El superusuario *root* tiene acceso irrestricto a todos los archivos del sistema operativo.

2.3.1 Permisos de Archivos y Directorios

Cada usuario puede pertenecer a uno o más grupos. Al momento de ser creado el usuario en el sistema, se debe asignarlo a un grupo, en caso de no hacerlo explícito, será incorporado a un grupo definido por defecto en la configuración del sistema. En algunas distribuciones de *GNU/Linux*, este procedimiento consiste en crear en forma automática un grupo con el nombre idéntico al del usuario, otras optan por asignarlo a un grupo con una denominación genérica, por ejemplo *users*.

El administrador del sistema tiene la facultad de agregar al usuario a otros grupos de acuerdo a las necesidades de uso de los servicios disponibles que tiene el usuario. El uso de estos grupos permite establecer una política de acceso organizada gestionando los accesos mediante el uso de grupos y garantizar el acceso a los usuarios administrando la pertenencia a grupos.

Para cada objeto (archivo) que se encuentre en el sistema, **GNU/Linux** almacena información administrativa acerca de cada uno de ellos en la **tabla de inodos**. Entre la información que contiene la *tabla de inodos* se encuentra lo siguiente:

- Puntero a la posición física en el disco rígido.
- Nombre del archivo.

- La identificación del dueño (*UID*).
- La identificación del grupo (*GID*).
- La identificación del dispositivo que contiene el archivo.
- El modo del archivo y las reglas de acceso o permisos.
- Tamaño del archivo en bytes.
- Fecha y hora del último acceso.
- Fecha y hora de la última modificación.
- Fecha y hora de la última modificación del inodo.
- Cantidad de enlaces al archivo.

Como *GNU/Linux* es un sistema operativo multiusuario, se debe garantizar la protección de la información que genera cada usuario del sistema, para lo cual se estableció un mecanismo que permite que un usuario no pueda ver la información creada por otro a menos que éste, o el superusuario, le otorgue los permisos necesarios. Cuando un usuario crea un archivo, este recibe los atributos asociados al dueño y al grupo principal de éste y los permisos del directorio contenedor.

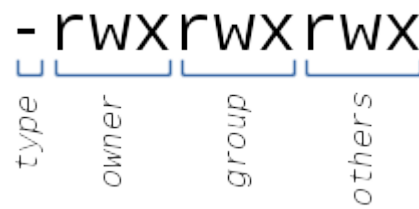
Existen tres tipos de permisos que se pueden asociar a un archivo:

- *Lectura*, identificado con la letra **r** (*read*)
- *Escritura*, identificado con la letra **w** (*write*)
- *Ejecución*, identificado con la letra **x** (*execute*)

Estos permisos se pueden fijar para tres clases de usuario: el propietario del archivo (*owner*), el grupo de usuarios al que pertenece el archivo (*group*) y para el resto de los usuarios del sistema (*others*).

El permiso de *lectura* permite que un usuario pueda leer el contenido del archivo o en el caso de que el archivo sea un directorio, la posibilidad de ver el contenido del mismo. El permiso *escritura* permite al usuario modificar y escribir el archivo y en el caso de un directorio permite crear nuevos archivos o directorios en él o eliminar archivos o directorios. El permiso de *ejecución* permite al usuario ejecutar el archivo (programa o script) y, en el caso de los directorios permite al usuario posicionarse en él.

La representación de estos permisos se presenta como una combinación de nueve *bits* donde cada grupo de tres *bits*, mostrados como **rwX**, indica los permisos para las clases de usuario *owner*, *group* y *others* respectivamente. El primer *bit* indica el tipo de archivo, con un **-** para los archivos regulares y una **d** cuando se trata de un directorio.



Cuando un archivo tiene asociado un permiso determinado, se dice que el *bit* está *encendido* y muestra la letra correspondiente, en caso contrario se representa con un guión (-).

Algunas consideraciones

- Para ejecutar un comando el *bit* de ejecución `x` debe estar encendido. Para algunos tipos de archivos, también se requiere el *bit* de lectura `r` para su ejecución.
- Como el nombre del archivo no es parte del archivo en sí, es una función del directorio que lo contiene, para borrar o renombrar el archivo se necesita permiso de escritura en el directorio.
- Para mover un archivo de un directorio a otro se requiere permisos de escritura en ambos directorio.
- El bit de ejecución de un directorio no significa *ejecutar*, significa *buscar*. Si este *bit* no está encendido el usuario no puede buscar en el directorio aunque el sea el propietario del mismo.

Para ver los permisos que tiene asociado un archivo se puede utilizar el comando `ls -lh` en la consola:

```
leonardo@tarod:~/Documentos$ ls -lh
total 4372
-rw-r----- 1 leonardo users 4,3M jun 20 23:10 funciones.pdf
-rwxr-xr-x 1 leonardo users 4,2K jun 20 23:11 install.sh
leonardo@tarod:~/Documentos$
```

En este ejemplo se observa el listado de dos archivos, donde la primera cadena de caracteres representa la asignación de permisos del archivo.

Archivo	Permisos
funciones.pdf	Lectura y escritura para el usuario <i>leonardo</i> (<code>rw-</code>), Lectura para los usuarios del grupo <i>users</i> (<code>r--</code>), ningún permiso para el resto de los usuarios del sistema.
install.sh	Lectura, escritura y ejecución para el usuario <i>leonardo</i> (<code>rwx</code>), Lectura y ejecución para los usuarios del grupo <i>users</i> (<code>r-x</code>), Lectura y ejecución para el resto de los usuarios del sistema (<code>r-x</code>).

Para gestionar los permisos de archivos y directorios se utiliza el comando `chmod` (*change mode*), para gestionar el usuario propietario de archivos y directorios el comando `chown` (*change ownership*) y, por último, para gestionar el grupo propietario de los archivos y directorios se usa el comando `chgrp` (*change group*).

2.3.2 Cambiar los permisos de archivos y directorios

Para cambiar los permisos del archivo `install.sh` y habilitar la edición por parte de los integrantes del grupo `users` se utiliza el comando `chmod` con los parámetros `g+w`:

```
leonardo@tarod:~/Documentos$ ls -lh *.sh
total 4372
-rwxr-xr-x 1 leonardo users 4,2K jun 20 23:11 install.sh

leonardo@tarod:~/Documentos$ chmod g+w install.sh

leonardo@tarod:~/Documentos$ ls -lh *.sh
total 4372
-rwxrwxr-x 1 leonardo users 4,2K jun 20 23:11 install.sh
```

Además de asignar los permisos explícitamente mediante los parámetros indicados por letras, se puede utilizar una codificación compuesta por tres números, uno para cada rango de usuario, la escala valores va de 0 a 7 donde cada uno representa una combinación de los permisos `rwX` de acuerdo a la siguiente tabla:

Codificación de permisos para archivos y directorios

Binario	Númérico	Letras
000	0	---
001	1	--X
010	2	-W-
011	3	-WX
100	4	r--
101	5	r-X
110	6	rw-
111	7	rwX

El comando análogo al ejemplo anterior utilizando los parámetros numéricos es el siguiente:

```
leonardo@tarod:~/Documentos$ chmod 775 install.sh
```

2.3.3 Cambiar dueño y grupo de archivos y directorios

Si lo que se pretende es cambiar es el propietario del archivo `funciones.pdf` para darle los permisos al usuario `mario`, se usa el comando `chown` de la siguiente manera:

```
leonardo@tarod:~/Documentos$ ls -lh *.pdf
-rw-r----- 1 leonardo users 4,3M jun 20 23:10 funciones.pdf

leonardo@tarod:~/Documentos$ chown mario funciones.pdf

leonardo@tarod:~/Documentos$ ls -lh *.pdf
-rwxrwxr-x 1 mario users 4,3M jun 20 23:10 funciones.pdf
```

Ahora, si se quiere cambiar el grupo propietario del mismo archivo para que los permisos correspondan al grupo de desarrollo denominado `desa`, existen dos alternativas. La primera es utilizar el comando `chgrp` de la siguiente manera:

```
leonardo@tarod:~/Documentos$ ls -lh *.pdf
-rw-r----- 1 mario users 4,3M jun 20 23:10 funciones.pdf

leonardo@tarod:~/Documentos$ chgrp desa funciones.pdf

leonardo@tarod:~/Documentos$ ls -lh *.pdf
-rwxrwxr-x 1 mario desa 4,3M jun 20 23:10 funciones.pdf
```

También podemos definir el grupo propietario utilizando el comando `chown`:

```
leonardo@tarod:~/Documentos$ ls -lh *.pdf
-rw-r----- 1 mario users 4,3M jun 20 23:10 funciones.pdf

leonardo@tarod:~/Documentos$ chown :desa funciones.pdf

leonardo@tarod:~/Documentos$ ls -lh *.pdf
-rwxrwxr-x 1 mario desa 4,3M jun 20 23:10 funciones.pdf
```

El comando `chown` espera como parámetro la combinación `<usuario>:<grupo>`. Si se explicita solamente el `<usuario>`, es lo único que cambiará; si se explicita el `<grupo>` precedido por el caracter `:` solo se cambiará el `<grupo>`.

Esto da a entender que existe otra alternativa más completa y que simplifica el trabajo ya que, con una sola ejecución del comando `chown` podemos cambiar tanto el usuario como el grupo propietario del archivo como se observa en el caso siguiente:

```
leonardo@tarod:~/Documentos$ ls -lh *.pdf
-rw-r----- 1 leonardo users 4,3M jun 20 23:10 funciones.pdf

leonardo@tarod:~/Documentos$ chown mario:desa funciones.pdf

leonardo@tarod:~/Documentos$ ls -lh *.pdf
-rwxrwxr-x 1 mario desa 4,3M jun 20 23:10 funciones.pdf
```

De esta manera se obtiene el mismo resultado pero utilizando un solo comando.

3 Gestión de procesos

Una tarea de usuario es una actividad que debe realizar el sistema operativo que bien puede ser la ejecución de un comando, una orden, editar un archivo, etc. Para ello, el sistema operativo debe ejecutar un programa adecuado para realizar esta tarea, normalmente se denomina *programa ejecutable*, ya que contiene las instrucciones máquina para realizar esta tarea.

Cuando el *programa ejecutable* se carga en memoria se le llama **proceso**, o programa en ejecución, ya que contiene, además del programa ejecutable, todas las estructuras de datos para que se pueda realizar esta tarea. Cuando finaliza su ejecución se libera toda la memoria utilizada.

Un «proceso» es la representación en memoria de un programa en ejecución. Incluye toda la información necesaria para la ejecución apropiada del programa (el código en sí pero también los datos que tiene en memoria, la lista de archivos que ha abierto, las conexiones de red que ha establecido, etc.) Un único programa puede ser instanciado en varios procesos inclusive bajo el mismo «ID» de usuario.

Este proceso se puede suspender, bloquear, continuar su ejecución de acuerdo a las necesidades del usuario y a lo que ordene el sistema operativo. Una derivación de este concepto de proceso es que, en los procesadores modernos, un proceso puede ser dividido en varias sub tareas llamados hilos o *threads*.

Los *Unix* son sistemas operativos multitareas que ejecutan procesos e hilos (*threads*) mediante una técnica llamada multiprogramación, que permite ejecutar más de un **proceso/hilo** a la vez en forma concurrente y de forma más eficiente que si lo ejecutáramos en forma secuencial, ya que se puede solapar la ejecución de entrada/salida de un proceso con la ejecución en CPU de otro proceso. Para identificar de manera inequívoca cada proceso, el núcleo del sistema les asigna un número denominado **PID** (*process identification*) necesario para administrar y referenciar el proceso.

Cuando el núcleo supera su fase de inicialización, ejecuta el primer proceso: `init`, al que se le asigna el **PID** número 1. Este proceso solo rara vez es útil por sí mismo.

Un proceso puede clonarse a sí mismo mediante un método llamado **bifurcación** (*fork*). El núcleo reserva un espacio de memoria idéntico al del proceso padre y se crea un nuevo proceso para utilizarlo. En ese momento, la única diferencia entre estos dos procesos es su **PID**. El nuevo proceso se denomina *proceso hijo* y al proceso original, cuyo **PID** no cambió, se lo denomina *proceso padre*.

A veces, el *proceso hijo* continúa su vida de forma independiente al *proceso padre*, con sus propios datos copiados del *proceso padre*. Sin embargo, en muchos casos, el *proceso hijo* ejecuta otro programa. Con unas pocas excepciones, simplemente se reemplaza su memoria con aquella del nuevo programa y comienza la ejecución del mismo.

Este es el mecanismo utilizado por el proceso con **PID 1** para iniciar servicios adicionales y ejecutar la secuencia completa de inicio del sistema. En algún punto, uno de los procesos de la descendencia de `init` inicia una interfaz gráfica en la que los usuarios pueden iniciar sesión.

Cuando un proceso finaliza la tarea para la que fue iniciado, termina. El núcleo recupera la memoria asignada a este proceso y no le asignará más divisiones de tiempo de ejecución. Se le informa al *proceso padre* sobre la finalización de su *proceso hijo*, lo que permite a un proceso esperar que se complete una tarea que delegó a un *proceso hijo*. Este comportamiento es fácil de identificar en la consola. Cuando se ingresa un comando, sólo vuelve el *prompt* cuando finaliza la ejecución de dicho comando.

La mayoría de las consolas permiten ejecutar programas en segundo plano, sólo es cuestión de agregar el símbolo `&` al final del comando. Se mostrará el *prompt* inmediatamente, lo que puede llevar a problemas si la orden necesita mostrar datos por su cuenta.

Los procesos son susceptibles de recibir señales con el fin de informarles de algún evento, como salir de la cola de ejecución, ser eliminados, cambiarles la prioridad de ejecución, etc.

Los procesos que se encuentren en ejecución en un determinado momento serán, en general, de diferente naturaleza. Se pueden categorizar como:

Procesos de sistema

Ya sean procesos asociados al funcionamiento local de la máquina, *kernel*, o bien procesos (denominados *daemons* - *demonios*) asociados al control de diferentes servicios. Por otro lado pueden ser locales, o de red, debido a que se ofrece el servicio actuando como servidor o se reciben los resultados del servicio en el caso de actuar de clientes. La mayoría de estos procesos de sistema aparecerán asociados al usuario *root* o en su defecto a usuarios definidos específicamente para el servicio en cuestión.

Procesos del usuario administrador

Cuando se utiliza el usuario *root*, los procesos interactivos o aplicaciones lanzadas también aparecerán como procesos asociados al usuario *root*.

Procesos de usuarios del sistema

Son los procesos asociados a la ejecución de las aplicaciones de los usuarios, ya sea tareas interactivas en modo texto o en modo gráfico.

3.1 El Sistema de Archivos `/proc`

El sistema de archivos `/proc` es una ventana para observar el funcionamiento del *kernel* en memoria. De allí se puede extraer valiosísima información sobre los procesos que están ejecutándose en el sistema. En el directorio `/proc` existe un número de subdirectorios correspondiente a cada **PID**. Dentro de estos directorios enumerados existen archivos y directorios que nos dan información acerca del proceso asociado a ese **PID**.

3.2 Herramientas

Las herramientas más conocidas en los sistemas operativos *GNU/Linux* para gestionar los procesos son `ps`, `kill` y `top`.

Con ellas se puede obtener el listado de los procesos en ejecución en el sistema, qué usuarios lo ejecutaron y otro tipo de información de utilidad para la administración de los procesos.

3.2.1 El comando `ps`

Este comando se utiliza en la consola para visualizar los procesos en ejecución en el sistema. Puede mostrar una gran variedad de información que permitirá saber qué ocurre en el sistema y cómo se están utilizando los recursos.

El uso más simple del comando es ejecutarlo sin parámetros, en este caso, la información que presenta corresponde a los procesos que tiene corriendo el usuario.

```
leonardo@tarod:~$ ps
  PID TTY          TIME CMD
 1831 pts/1    00:00:00 zsh
 1851 pts/1    00:00:00 bash
 4245 pts/1    00:00:00 ps
leonardo@tarod:~$
```

Las cuatro columnas presentadas corresponden al **PID**, a la *Terminal* (**TTY**) asociada al proceso, el tiempo de uso de *CPU* acumulado por el proceso y por último, el comando o programa (**CMD**) que inició el proceso.

Esta información, si bien es útil, poco dice sobre el estado del sistema. Para ahondar un poco más en el funcionamiento, se debe solicitar más información sobre los procesos del usuario y sobre aquellos que están ejecutándose para mantener el sistema funcionando.

Esto se logra corriendo el comando `ps` con los parámetros adecuados para obtener la información necesaria.

Por ejemplo, el parámetro `-e` muestra todos los procesos corriendo en el sistema y el parámetro `-f` muestra la información completa acerca de cada uno de los procesos listados.

```
leonardo@tarod:~$ ps -ef
UID          PID  PPID  C STIME TTY          TIME CMD
root           1      0  0 jul05 ?        00:00:01 /sbin/init
root           2      0  0 jul05 ?        00:00:00 [kthreadd]
root           3      2  0 jul05 ?        00:00:02 [ksoftirqd/0]
root           5      2  0 jul05 ?        00:00:00 [kworker/0:0H]
root           6      2  0 jul05 ?        00:00:00 [kworker/u4:0]

[ ... ]

leonardo 1076  1065  0 jul05 ?        00:00:00 gnome-session
leonardo 1671  1660  0 jul05 ?        00:00:00 gnome-pty-helper
leonardo 1672  1660  0 jul05 pts/0    00:00:00 /bin/zsh
leonardo 1756  1258 30 jul05 ?        00:31:56 firefox-esr
leonardo 1830  1827  0 jul05 ?        00:00:00 gnome-pty-helper
leonardo 1831  1827  0 jul05 pts/1    00:00:00 zsh
leonardo 1851  1831  0 jul05 pts/1    00:00:00 bash
leonardo 4427  1851  0 01:20 pts/1    00:00:00 ps -ef
leonardo@tarod:~$
```


En listado comienza con el proceso `/sbin/init` que, como se mencionó anteriormente, tiene el **PID 1**. Se observan procesos de los usuarios `root` y `leonardo`.

La tercer columna tiene un dato por demás importante, indica el **PID** del proceso padre.

Otra combinación de parámetros muy utilizada que lista los procesos de todos los usuarios con información añadida es la que forman los parámetros `a`, `u` y `x`:

Parámetro	Descripción
<code>-a</code>	Lista los procesos de todos los usuarios.
<code>-u</code>	Lista información del usuario que lo está corriendo, utilización de CPU y memoria, etc.
<code>-x</code>	Lista procesos de todas las terminales y usuarios.

La información más relevante que ofrece este listado es el uso de *CPU* y *Memoria RAM* del proceso.

```
leonardo@tarod:~$ ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1 110824  5228 ?        Ss   jul05    0:01 /sbin/init
root         2  0.0  0.0      0      0 ?        S    jul05    0:00 [kthreadd]
root         3  0.0  0.0      0      0 ?        S    jul05    0:03 [ksoftirqd/0]
root         5  0.0  0.0      0      0 ?        S<   jul05    0:00 [kworker/0:0H]
root         6  0.0  0.0      0      0 ?        S    jul05    0:00 [kworker/u4:0]
root         7  0.1  0.0      0      0 ?        S    jul05    0:11 [rcu_sched]
root         8  0.0  0.0      0      0 ?        S    jul05    0:00 [rcu_bh]
root         9  0.0  0.0      0      0 ?        S    jul05    0:00 [migration/0]
root        10  0.0  0.0      0      0 ?        S    jul05    0:00 [watchdog/0]
root        11  0.0  0.0      0      0 ?        S    jul05    0:00 [watchdog/1]
root        12  0.0  0.0      0      0 ?        S    jul05    0:00 [migration/1]
root        13  0.0  0.0      0      0 ?        S    jul05    0:02 [ksoftirqd/1]

[...]

leonardo  1258  7.0  9.3 1739468 288852 ?        Sl   jul05    9:28 /usr/bin/gnome-shell
leonardo  1671  0.0  0.0  14692   1728 ?        S    jul05    0:00 gnome-pty-helper
leonardo  1672  0.0  0.1  47124   5864 pts/0    Ss+  jul05    0:00 /bin/zsh
leonardo  1756 33.0 17.4 1449976 540212 ?        Sl   jul05   44:18 firefox-esr
leonardo  1830  0.0  0.0  14692   1780 ?        S    jul05    0:00 gnome-pty-helper
leonardo  1831  0.0  0.1  44776   5440 pts/1    Ss   jul05    0:00 zsh
leonardo  1851  0.0  0.1  24332   6152 pts/1    S    jul05    0:00 bash
root      4553  0.0  0.0      0      0 ?        S    01:35    0:00 [kworker/0:2]
root      4638  0.0  0.0      0      0 ?        S    01:40    0:00 [kworker/0:1]
root      4815  0.0  0.0      0      0 ?        S    01:45    0:00 [kworker/0:0]
leonardo  5013  0.0  0.0  19100   2556 pts/1    R+   01:50    0:00 ps -aux
leonardo@tarod:~$
```

3.2.2 El comando `pstree`

Una forma interesante de visualizar el *árbol* de procesos es utilizar el comando `pstree`. El resultado es un listado de los procesos que están corriendo en el sistema presentado como una estructura de árbol graficada, donde las ramas conectan *procesos* padres con hijos.

```
leonardo@tarod:~$ pstree
systemd+-ModemManager+-{gdbus}
      +-{gmain}
      -NetworkManager+-dhclient
          +-{NetworkManager}
          +-{gdbus}
          +-{gmain}
      -accounts-daemon+-{gdbus}
          +-{gmain}
      -cron
      -cups-browsed
      -cupsd
      -exim4
      -gdm3+-Xorg---2*[{Xorg}]
          -gdm-session-wor+-gnome-session+-applet.py---{gmain}
              +-evolution-alarm+-{cal-client-dbus}
                  +-{dconf worker}
                  +-{evolution-alarm}
                  +-{gdbus}
              -gnome-settings--+-{dconf worker}
                  +-{gdbus}
                  +-{gmain}
                  +-{pool}
              -gnome-shell+-{JS GC Helper}
                  +-{JS Sour~ Thread}
                  +-{threaded-ml}
              -ssh-agent
              +-{dconf worker}
              +-{gdbus}
              +-{gmain}
              +-{gdbus}
              +-{gmain}
          +-{gdbus}
          +-{gdm SIGUSR1 cat}
          +-{gmain}
      -gedit+-gnome-pty-helpe
          +-zsh
          +-{dconf worker}
          +-{gdbus}
          +-{gedit}
          +-{gmain}
      -nautilus+-python+-{gdbus}
          +-{gmain}
          +-{dconf worker}
          +-{gdbus}
          +-{gmain}
      -zeitgeist-daemo---{gdbus}
      -zeitgeist-fts+-cat
          +-{gdbus}
          +-{gmain}
```

3.2.3 El comando `kill`

El comando `kill` permite eliminar procesos del sistema mediante el envío de señales al proceso como, por ejemplo, la señal de terminación `kill -9 PID_del_proceso`. El código 9 corresponde a la señal `SIGKILL`.

Resulta útil para procesos con comportamiento inestable o programas interactivos que han dejado de responder. Podemos ver una lista de las señales válidas en el sistema con el comando:

```
leonardo@tarod:~$ kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
6) SIGABRT     7) SIGBUS     8) SIGFPE     9) SIGKILL    10) SIGUSR1
11) SIGSEGV    12) SIGUSR2    13) SIGPIPE    14) SIGALRM    15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD   18) SIGCONT    19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU   23) SIGURG     24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH   29) SIGIO      30) SIGPWR
31) SIGSYS     34) SIGRTMIN  35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
leonardo@tarod:~$
```

En el ejemplo anterior de ejecución del comando `ps -aux` se observa la información del proceso **firefox-esr** correspondiente al navegador de internet que el usuario está utilizando en ese momento. Ante un comportamiento inestable o falta de respuesta del programa, se debe terminar o *matar* ese proceso.

```
leonardo@tarod:~$ ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
[...]
leonardo  1756 33.0  17.4 1449976 540212 ?        Sl    jul05   44:18 firefox-esr
[...]
```

El **PID** del proceso es **1756**, por lo tanto, para *matar* el proceso, se debe ejecutar el comando:

```
leonardo@tarod:~$ kill -9 1756
```

Si el programa o proceso que no responde es una aplicación gráfica y todavía se tiene control del entorno gráfico se puede recurrir al comando `xkill`.

Para utilizar este comando, se debe presionar la combinación de teclas **<Alt>+<F2>** y en la ventana de comando ingresar el comando y ejecutarlo. El puntero del mouse se transformará en una cruz. Lo que resta es presionar en cualquier punto de la ventana de la aplicación que se quiere *matar*.

3.2.4 El comando `top`

Es una herramienta interactiva que muestra los procesos en ejecución en tiempo real. La lista se puede ordenar de acuerdo al recurso que se pretende monitorear, el ordenamiento predeterminado es según la cantidad de procesador utilizada y se puede obtener mediante la tecla **P**. Entre otros criterios de ordenación se encuentran:

- según la cantidad de memoria ocupada (tecla **M**)
- según el tiempo total de uso de procesador (tecla **T**)
- según el identificador de proceso (tecla **N**).

Esta herramienta permite la gestión de los procesos en forma interactiva, por ejemplo, presionando la tecla **k** e ingresando el **PID** se *mata* el proceso.

```

top
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
top - 18:27:22 up 1:41, 3 users, load average: 0.54, 0.42, 0.34
Tasks: 166 total, 1 running, 163 sleeping, 0 stopped, 2 zombie
%Cpu(s): 2.1 us, 4.7 sy, 0.0 ni, 93.1 id, 0.2 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 3092800 total, 1854768 used, 1238032 free, 82824 buffers
KiB Swap: 1438716 total, 0 used, 1438716 free, 683884 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM    TIME+  COMMAND
 1195 leonardo  20   0 1752180 311012 56672 S   9.0 10.1   6:17.96 gnome-shell
 566 root      20   0 543108 113912 43740 S   4.3  3.7   5:56.29 Xorg
1083 leonardo  20   0 114740  3176  2792 S   1.0  0.1   0:27.30 VBoxClient
   7 root      20   0      0      0      0 S   0.3  0.0   0:08.60 rcu_sched
 504 debian--  20   0  72136  36724  6116 S   0.3  1.2   0:15.02 tor
 3353 root      20   0      0      0      0 S   0.3  0.0   0:02.55 kworker/0:0
 3713 leonardo  20   0 412256  28172 20868 S   0.3  0.9   0:03.54 gnome-terminal-
 4003 leonardo  20   0 23652  2924  2420 R   0.3  0.1   0:00.19 top
   1 root      20   0 176356  5244  3104 S   0.0  0.2   0:01.31 systemd
   2 root      20   0      0      0      0 S   0.0  0.0   0:00.00 kthreadd
   3 root      20   0      0      0      0 S   0.0  0.0   0:02.44 ksoftirqd/0
   5 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 kworker/0:0H
   8 root      20   0      0      0      0 S   0.0  0.0   0:00.00 rcu_bh
   9 root      rt    0      0      0      0 S   0.0  0.0   0:00.02 migration/0
  10 root      rt    0      0      0      0 S   0.0  0.0   0:00.21 watchdog/0
  11 root      rt    0      0      0      0 S   0.0  0.0   0:00.20 watchdog/1
  12 root      rt    0      0      0      0 S   0.0  0.0   0:00.02 migration/1
  13 root      20   0      0      0      0 S   0.0  0.0   0:01.74 ksoftirqd/1
  15 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 kworker/1:0H
  16 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 khelper
  17 root      20   0      0      0      0 S   0.0  0.0   0:00.00 kdevtmpfs
  18 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 netns
  19 root      20   0      0      0      0 S   0.0  0.0   0:00.00 khungtaskd
  20 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 writeback
  21 root      25   5      0      0      0 S   0.0  0.0   0:00.00 ksmd
  22 root      39  19      0      0      0 S   0.0  0.0   0:00.00 khugepaged
  23 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 crypto
  24 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 kintegrityd

```

3.3 Demonios

Un **demonio** es un proceso iniciado automáticamente durante la secuencia de inicio del sistema. Continúa ejecutándose en segundo plano para realizar tareas de mantenimiento o proveer servicios a otros procesos. Esta *tarea en segundo plano* es realmente arbitraria y no tiene un rol especial desde el punto de vista del sistema.

Simplemente son procesos, muy similares a otros procesos, que se ejecutarán cuando le corresponda a su división de tiempo. Esta distinción es sólo para los humanos: se dice de un proceso que ejecuta sin interacción de un usuario que se ejecuta *en segundo plano* o *como un demonio*.

Un atributo clave de un proceso **demonio** es que no es parte de un terminal, esto se puede comprobar con el comando `ps` que mostrará un signo de pregunta (?) en la columna **TTY**. Esto implica que el **demonio** no recibe señales asociadas con el terminal.

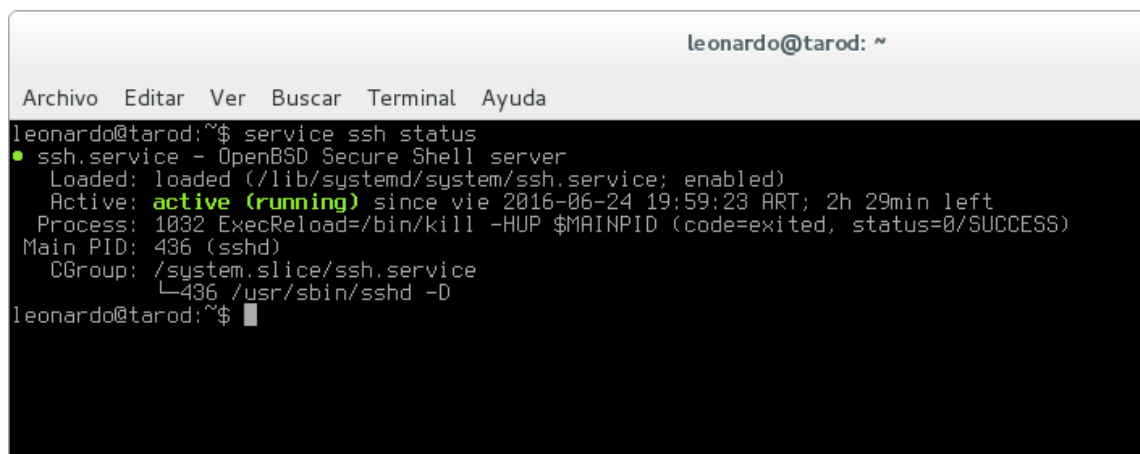
Normalmente los **demonios** esperan a un evento para actuar, así como a una señal, un archivo siendo creado, un *timeout* o entrada de datos desde una conexión de red. Cuando el evento ocurre el **demonio** se despierta, le da servicio al evento y vuelve a dormir, generalmente el demonio da inicio a un proceso hijo para que maneje el evento y, el **demonio** pueda escuchar y prepararse para otro evento.

3.3.1 Monitorear que hacen los demonios

Qué hace un demonio es, de alguna manera, complicado de entender ya que no interactúa directamente con el administrador. Para controlar que un *demonio* realmente esté trabajando se debe probarlo. Por ejemplo, se puede verificar que el *demonio* Apache (servidor web) esté levantado con una petición HTTP.

Para permitir dichas pruebas cada *demonio* generalmente graba todo lo que hace, así como también los errores que encuentra, en lo que se llaman *archivos de registro* o *registros de sistema*. Los registros se almacenan en `/var/log/` o alguno de sus subdirectorios.

Los sistemas *GNU/Linux* tienen herramientas que permiten gestionar los *demonios* asociados a los servicios del sistema. El comando `service` es uno de los más utilizados.



```
leonardo@tarod:~$ service ssh status
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled)
   Active: active (running) since vie 2016-06-24 19:59:23 ART; 2h 29min left
     Process: 1032 ExecReload=/bin/kill -HUP $MAINPID (code=exited, status=0/SUCCESS)
    Main PID: 436 (sshd)
      CGroup: /system.slice/ssh.service
              └─436 /usr/sbin/sshd -D
leonardo@tarod:~$
```

4 Comandos Básicos

Para desplazarse por la línea de comando y realizar las actividades necesarias para administrar un equipo con GNU/Linux, es necesario conocer los comandos y sus parámetros.

Existe un conjunto mínimo de comandos básicos que todo administrador de sistemas GNU/Linux debe conocer y manejar con soltura. Este grupo de comandos no es estricto, se puede ampliar de acuerdo a las necesidades de administración de la infraestructura de la cual es responsable el *Administrador de Sistemas*.

En secciones anteriores se presentaron los comandos `ps`, `pstree` para gestionar procesos. También se mostró el uso de `ls`, `chmod`, `chown` y `chgrp` para la gestión de los permisos de archivos y directorios.

4.1 Ayuda sobre los comandos

Todos los comandos de la terminal de *GNU/Linux* tienen la documentación de ayuda necesaria para entender cómo se utilizan cada uno de ellos. La herramienta por excelencia para consultar la ayuda es el comando `man` que permite la consulta del manual de *GNU/Linux*, agrupado en varias secciones, correspondientes a comandos de administración, de formatos de ficheros, de comandos de usuario, de llamadas de lenguaje C, y otras categorías. Para obtener la ayuda asociada a un comando se debe ejecutar:

```
leonardo@tarod$ man <comando>
```

Por ejemplo la ayuda del comando `ps` se obtiene con:

```
leonardo@tarod$ man ps
```

```
man ps

Archivo  Editar  Ver  Buscar  Terminal  Ayuda

PS(1)                                          User Commands                                          PS(1)

NAME
  ps - report a snapshot of the current processes.

SYNOPSIS
  ps [options]

DESCRIPTION
  ps displays information about a selection of the active processes.  If you want a repetitive update of the
  selection and the displayed information, use top(1) instead.

  This version of ps accepts several kinds of options:

  1  UNIX options, which may be grouped and must be preceded by a dash.
  2  BSD options, which may be grouped and must not be used with a dash.
  3  GNU long options, which are preceded by two dashes.

  Options of different types may be freely mixed, but conflicts can appear.  There are some synonymous options,
  which are functionally identical, due to the many standards and ps implementations that this ps is compatible
  with.

  Note that "ps -aux" is distinct from "ps aux".  The POSIX and UNIX standards require that "ps -aux" print all
  processes owned by a user named "x", as well as printing all processes that would be selected by the -a
  option.  If the user named "x" does not exist, this ps may interpret the command as "ps aux" instead and print
  a warning.  This behavior is intended to aid in transitioning old scripts and habits.  It is fragile, subject
  to change, and thus should not be relied upon.

  By default, ps selects all processes with the same effective user ID (euid=EUID) as the current user and
  associated with the same terminal as the invoker.  It displays the process ID (pid=PID), the terminal
  associated with the process (tname=TTY), the cumulated CPU time in [DD-]hh:mm:ss format (time=TIME), and the
  executable name (ucmd=CMD).  Output is unsorted by default.

Manual page ps(1) line 1 (press h for help or q to quit)
```

La mayoría de los parámetros se indican con un `-` delante, en el caso de utilizar varios parámetros en un comando, se pueden agrupar detrás de un solo `-` como se vió en el caso de uso de `ps -aux`. Los parámetros que se identifican con palabras completas son precedidos por dos signos `--` como el parámetro `--help` disponible la mayoría de los comandos.

Hay dos parámetros que se utilizan en la mayoría de los comandos, el primero es `-h` que permite mostrar los datos de una manera más comprensible utilizando las unidades de *Kb*, *Mb* y *Gb* en contraste con la unidad estándar de bloques de 1024 bytes.

El otro parámetro es `-R` que hace recursiva la aplicación del comando a los directorios incluidos en el directorio actual y sus hijos hasta el último nivel del árbol. En algunos comandos puede ser que el parámetro sea `-r` (en minúscula), para verificarlo se debe consultar la página de manual con el comando `man`.

4.2 Información de sistema, memoria, espacio en disco

Algunos comandos permiten conocer el entorno del sistema operativo y de la sesión de terminal que se está utilizando.

4.2.1 El comando `uname`

Para obtener información acerca del sistema utilizamos este comando que, con los parámetros disponibles podemos especificar la consulta. El resultado es una cadena de texto con valores ordenados. Por defecto, el comando sin parámetros devuelve el nombre del *kernel* al igual que la salida de información provista por el parámetro `-s`.

```
administrador@xubuntu1604:~$ uname
Linux
administrador@xubuntu1604:~$
```

El parámetro `-a` o `-all` muestra la totalidad de la información que puede dar este comando.

```
administrador@xubuntu1604:~$ uname -a
Linux xubuntu1604 4.4.0-28-generic #47-Ubuntu SMP Fri Jun 24 10:09:13 UTC 2016
x86_64 x86_64 x86_64 GNU/Linux
administrador@xubuntu1604:~$
```

El parámetro `-s` o `--kernel-name` informa el nombre del *kernel*.

```
administrador@xubuntu1604:~$ uname -s
Linux
administrador@xubuntu1604:~$
```

El parámetro `-n` o `--nodename` informa el nombre de red del equipo.

```
administrador@xubuntu1604:~$ uname -n
xubuntu1604
administrador@xubuntu1604:~$
```

El parámetro `-r` o `--kernel-release` muestra el *release* o *versión* del *kernel*.

```
administrador@xubuntu1604:~$ uname -r
4.4.0-28-generic
administrador@xubuntu1604:~$
```

El parámetro `-v` o `--kernel-version` informa la versión del *kernel*.

```
administrador@xubuntu1604:~$ uname -v
#47-Ubuntu SMP Fri Jun 24 10:09:13 UTC 2016
administrador@xubuntu1604:~$
```

El parámetro `-m` o `--machine` informa la denominación del *hardware* del equipo.

```
administrador@xubuntu1604:~$ uname -m
x86_64
administrador@xubuntu1604:~$
```

El parámetro `-p` o `--processor` muestra el tipo de procesador.

```
administrador@xubuntu1604:~$ uname -p
x86_64
administrador@xubuntu1604:~$
```

El parámetro `-i` o `--hardware-platform` informa el tipo de plataforma de hardware.

```
administrador@xubuntu1604:~$ uname -i
x86_64
administrador@xubuntu1604:~$
```

El parámetro `-o` o `--operating-system` muestra la denominación del sistema operativo.

```
administrador@xubuntu1604:~$ uname -o
GNU/Linux
administrador@xubuntu1604:~$
```

4.2.2 El comando `uptime`

Muchas veces es importante saber el tiempo que lleva funcionando el sistema. Para ello existe el comando `uptime` que muestra una línea de información con datos relevantes como la hora de consulta, el tiempo que hace que el sistema está prendido, la cantidad de usuarios y la carga del sistema.

```
administrador@xubuntu1604:~$ uptime
22:48:55 up 3:15, 3 users, load average: 0,00, 0,00, 0,00
administrador@xubuntu1604:~$
```

Este comando tiene dos parámetros que hacen más legibles la información, el primero de ellos es el parámetro `-s` (*since - desde cuando*) que indica la fecha y hora de inicio del sistema.

```
administrador@xubuntu1604:~$ uptime -s
2016-07-09 19:33:30
administrador@xubuntu1604:~$
```


El otro parámetro es `-p`, que muestra la cantidad de tiempo en años, días, horas y minutos.

```
administrador@xubuntu1604:~$ uptime -p
up 3 hours, 16 minutes
administrador@xubuntu1604:~$
```

4.2.3 El comando `date`

Muchas veces es necesaria la información de fechas, para ver cuál es la fecha del sistema, para nombrar un archivo, para comparar antigüedad de un archivo, asignar el valor de una variable de entorno.

Este comando permite tanto visualizar la fecha del sistema como también configurarla.

La utilización de parámetros es muy común y generalmente necesaria para obtener la información necesaria.

La ejecución sin parámetros devuelve una cadena con el día de la semana, el mes, el día del mes, la hora, la zona horaria y el año.

```
administrador@xubuntu1604:~$ date
dom jul 10 22:11:28 ART 2016
```

Esta información es completa pero, si se necesita para incorporarla en un texto o en un línea de un archivo de registro, para completar el nombre de un archivo de backup, y muchas otras tareas, es conveniente manipular el resultado. Esto se logra utilizando los parámetros adecuados.

Para obtener la fecha con el formato **ISO 8601** utilizamos el parámetro `-I`:

```
administrador@xubuntu1604:~$ date -I
2016-07-10
```

Para obtener la fecha con el formato **RFC 2822** utilizamos el parámetro `-R`:

```
administrador@xubuntu1604:~$ date -R
Sun, 10 Jul 2016 22:11:06 -0300
```

Para obtener la fecha con la zona horaria **UTC** utilizamos el parámetro `-u`:

```
administrador@xubuntu1604:~$ date -u
lun jul 11 01:11:43 UTC 2016
```

Para obtener la fecha con un formato personalizado podemos utilizar el carácter `+` seguido de los parámetros de formato especiales, algunos presentan información completa:

```
administrador@xubuntu1604:~$ date +%D
07/10/16
```

```
administrador@xubuntu1604:~$ date +%F
2016-07-10
```

```
administrador@xubuntu1604:~$ date +%T
17:13:35
```

```
administrador@xubuntu1604:~$ date +%R
17:13
```

Pero otros permiten utilizar elementos individuales para crear una salida que se pueda, por ejemplo, con el año, el mes, el día, la hora, los minutos y los segundos para concatenar con el nombre de un archivo:

```
administrador@xubuntu1604:~$ date +%Y%m%d%H%M%S
20160710172434
```

También se puede obtener una visualización tan vistosa como la siguiente:

```
$ date +'Son las %H horas con %M minutos y %S segundos del %A, %d de %B del año %Y'
Son las 17 horas con 17 minutos y 51 segundos del domingo, 10 de julio del año 2016
```

Para conocer qué otros parámetros y combinaciones se pueden utilizar se puede ver la página de ayuda provista por el comando `man`.

4.2.4 El comando `df`

A medida que se crean y escriben archivos en el disco rígido el espacio disponible disminuye, por lo tanto, es necesario monitorear el uso del espacio en cada partición para evitar problemas de llenado de disco con la consecuente falla del sistema operativo.

Este comando informa la distribución de espacio de disco en los sistemas de archivo de las particiones. Al ejecutarlo sin parámetros presenta una salida con información completa de todos los sistemas de Archivos:

```
administrador@xubuntu1604:~$ df
S.ficheros      bloques de 1K  Usados Disponibles  Uso% Montado en
udev            488724        0      488724      0% /dev
tmpfs           101632      3676      97956      4% /run
/dev/sda1       9480420 3777184    5198612    43% /
tmpfs           508152       164      507988      1% /dev/shm
tmpfs           5120         4        5116      1% /run/lock
tmpfs           508152        0      508152      0% /sys/fs/cgroup
/dev/sda2       4674688    12592    4401592      1% /home
tmpfs           101632       36      101596      1% /run/user/1000
administrador@xubuntu1604:~$
```

En este caso, la información más relevante es la de los sistemas de archivos (o ficheros) `/dev/sda1`, montada en el directorio raíz `/` que contiene los archivos del sistema operativo y `/dev/sda2`, montada en el directorio `/home` que contiene los archivos de los usuarios del sistema.

En la información se puede ver claramente el espacio total del sistema de archivos, el espacio ocupado y el espacio disponible. El porcentaje de uso es un dato que ayuda a una mayor comprensión del estado del sistema de archivos.

Toda mejora en la presentación de la información ayuda a la comprensión del estado de los sistemas de archivos. Como se mostró en la sección **1.2. Sistemas de Archivos**, se pueden utilizar parámetros `T` y `h` en tal sentido.

```
administrador@xubuntu1604:~$ df -Th
S.ficheros      Tipo      Tamaño Usados  Disp Uso% Montado en
udev            devtmpfs  478M    0      478M  0% /dev
tmpfs           tmpfs     100M    3,6M   96M    4% /run
/dev/sda1       ext4      9,1G    3,7G   5,0G   43% /
tmpfs           tmpfs     497M    164K   497M    1% /dev/shm
tmpfs           tmpfs     5,0M    4,0K   5,0M    1% /run/lock
tmpfs           tmpfs     497M    0      497M    0% /sys/fs/cgroup
/dev/sda2       ext4      4,5G    13M    4,2G    1% /home
tmpfs           tmpfs     100M    36K    100M    1% /run/user/1000
administrador@xubuntu1604:~$
```

Para una mejor visualización, se puede acotar la consulta sobre cuáles sistemas de archivos se pretende la información agregando como parámetro la denominación del sistema de archivos:

```
administrador@xubuntu1604:~$ df -Th /dev/sda1
S.ficheros      Tipo      Tamaño Usados  Disp Uso% Montado en
/dev/sda1       ext4      9,1G    3,7G   5,0G   43% /
administrador@xubuntu1604:~$

administrador@xubuntu1604:~$ df -Th /dev/sda2
S.ficheros      Tipo      Tamaño Usados  Disp Uso% Montado en
/dev/sda2       ext4      4,5G    13M    4,2G    1% /home
administrador@xubuntu1604:~$
```

Otra opción es solicitar la información de todos los sistemas de archivos de un tipo determinado, por ejemplo `ext4` utilizando el parámetro `-t`:

```
administrador@xubuntu1604:~$ df -Tht ext4
S.ficheros      Tipo      Tamaño Usados  Disp Uso% Montado en
/dev/sda1       ext4      9,1G    3,7G   5,0G   43% /
/dev/sda2       ext4      4,5G    13M    4,2G    1% /home
administrador@xubuntu1604:~$
```

4.2.5 El comando `du`

A medida que se crean y escriben archivos en el disco rígido el espacio disponible disminuye, por lo tanto, es necesario monitorear el uso del espacio en cada partición para evitar problemas de llenado de disco con la consecuente falla del sistema operativo.

Este comando informa el espacio que ocupa un archivo o un directorio y todos los archivos y subdirectorios contenidos.

Por defecto el comando `du` reporta el tamaño en el uso de bloques lógicos del sistema operativo, o en pedazos de 1024 bytes. Esto no es de mucha ayuda para interpretar fácilmente la información.

```
administrador@xubuntu1604:~$ du Desarrollo/manuales_admlinux1.tar.gz
53392    Desarrollo/manuales_admlinux1.tar.gz
```

El uso de los parámetros disponibles permite mejorar la visualización de los datos. El más interesante de ellos es el parámetro `-h` o su equivalente `--human-readable` que informa los tamaños en Kb, Mb y Gb. Este parámetro es, salvo raras ocasiones, utilizado para todas las consultas.

```
administrador@xubuntu1604:~$ du -h Desarrollo/manuales_admlinux1.tar.gz
53M     Desarrollo/manuales_admlinux1.tar.gz
```

Se puede utilizar para conocer el espacio utilizado por un archivo individual como se muestra en el ejemplo pero no es el caso de mayor uso. El uso más habitual es identificar el uso de espacio en los directorios.

Si solo se necesita el total de uso, el parámetro indicado es `-s`.

```
administrador@xubuntu1604:~$ du -hs Desarrollo
128M    Desarrollo
```

Si el comando se ejecuta sin indicar un directorio o archivo se toma como referencia el directorio actual identificado con `..`.

```
administrador@xubuntu1604:~$ du -hs
1,1G    .
```

Otro parámetro interesante es el que permite indicar la profundidad en el árbol de jerarquía de la cual se pretende la información. El parámetro es `--max-depth=n` y siempre requiere un valor de profundidad `n`.

```
administrador@xubuntu1604:~$ du -h --max-depth=1
520K    ../gimp-2.8
182M    ../Descargas
15M     ../Documentos
4,0K    ../Plantillas
12K     ../dbus
430M    ../.cache
4,4M    ../.config
60K     ../.ipython
3,6M    ../.oh-my-zsh.old
80K     ../.ninja_ide
128M    ../Desarrollo
13M     ../.local
16K     ../.gconf
16K     ../.gnupg
36K     ../.pki
4,0K    ../Videos
4,0K    ../Escritorio
```

```

16K      ./ssh
8,0K     ./gnome2
15M      ./Imágenes
210M     ./software
20K      ./gitkraken
12M      ./oh-my-zsh
4,0K     ./Público
49M      ./mozilla
4,0K     ./gnome2_private
80K      ./thumbnails
4,0K     ./Música
1,1G     .

```

A medida que se incrementa el valor de profundidad se observa que la cantidad de información es mayor.

```

administrador@xubuntu1604:~$ du -h --max-depth=2
4,0K     ./gimp-2.8/modules
4,0K     ./gimp-2.8/envIRON
4,0K     ./gimp-2.8/levels
4,0K     ./gimp-2.8/fractalexplorer
4,0K     ./gimp-2.8/tmp
4,0K     ./gimp-2.8/brushes
4,0K     ./gimp-2.8/tool-options
4,0K     ./gimp-2.8/interpreters
4,0K     ./gimp-2.8/themes
4,0K     ./gimp-2.8/gfig
4,0K     ./gimp-2.8/fonts
4,0K     ./gimp-2.8/gflare
4,0K     ./gimp-2.8/dynamics
4,0K     ./gimp-2.8/scripts
4,0K     ./gimp-2.8/gradients
4,0K     ./gimp-2.8/tool-presets
4,0K     ./gimp-2.8/plug-ins
4,0K     ./gimp-2.8/gimpress-ionist
4,0K     ./gimp-2.8/patterns
4,0K     ./gimp-2.8/palettes
520K     ./gimp-2.8
2,8M     ./Descargas/gparted
4,3M     ./Descargas/test
182M     ./Descargas
15M      ./Documentos/avales
15M      ./Documentos
4,0K     ./Plantillas
8,0K     ./dbus/session-bus
12K      ./dbus
8,0K     ./cache/mc
4,0K     ./cache/libgweather
288K     ./cache/fontconfig
391M     ./cache/mozilla
4,0K     ./cache/GitKraken
8,0K     ./cache/gegl-0.2
24K      ./cache/webkit
16K      ./cache/folks
4,0K     ./cache/gnome-screenshot
24K      ./cache/geocode-glib
20M      ./cache/thumbnails

```

```

8,0K    ../.cache/gnome-control-center
52K     ../.cache/evolution
20K     ../.cache/wocky
16K     ../.cache/rhythmbox
4,0K    ../.cache/pocket
72K     ../.cache/opera
732K    ../.cache/gstreamer-1.0
44K     ../.cache/telepathy
19M     ../.cache/tracker
8,0K    ../.cache/totem
430M    ../.cache
8,0K    ../.config/mc
504K    ../.config/GitKraken
8,0K    ../.config/gtk-2.0
16K     ../.config/rabbitvcs
8,0K    ../.config/yelp
2,7M    ../.config/libreoffice
84K     ../.config/pulse
24K     ../.config/nautilus
8,0K    ../.config/enchant
16K     ../.config/Notepadqq
20K     ../.config/dconf
16K     ../.config/gnome-control-center
8,0K    ../.config/goa-1.0
8,0K    ../.config/gtk-3.0
4,0K    ../.config/eog
8,0K    ../.config/ReText project
12K     ../.config/evolution
832K    ../.config/opera
52K     ../.config/inkscape
12K     ../.config/evince
8,0K    ../.config/gnome-session
8,0K    ../.config/Empathy
4,0K    ../.config/tracker
8,0K    ../.config/totem
28K     ../.config/gedit
92K     ../.config/vlc
8,0K    ../.config/ibus
4,4M    ../.config
4,0K    ../.ipython/extensions
44K     ../.ipython/profile_default
4,0K    ../.ipython/nbextensions
60K     ../.ipython
8,0K    ../.oh-my-zsh.old/cache
592K    ../.oh-my-zsh.old/.git
576K    ../.oh-my-zsh.old/themes
2,3M    ../.oh-my-zsh.old/plugins
32K     ../.oh-my-zsh.old/tools
68K     ../.oh-my-zsh.old/lib
8,0K    ../.oh-my-zsh.old/templates
8,0K    ../.oh-my-zsh.old/log
20K     ../.oh-my-zsh.old/custom
3,6M    ../.oh-my-zsh.old
72K     ../.ninja_ide/addins
80K     ../.ninja_ide
5,5M    ../Desarrollo/E_S1_2016
20K     ../Desarrollo/ppnp
11M     ../Desarrollo/tusl-adm-gnu-linux-1
620K    ../Desarrollo/estilos

```

```

6,8M    ./Desarrollo/ADM-GNU-LINUX-I
100K    ./Desarrollo/ejemplos
52M     ./Desarrollo/manuales adminlinux1
128M    ./Desarrollo
13M     ././local/share
13M     ././local
12K     ././gconf/apps
16K     ././gconf
16K     ././gnupg
32K     ././pki/nssdb
36K     ././pki
4,0K    ./Videos
4,0K    ./Escritorio
16K     ././ssh
4,0K    ././gnome2/accels
8,0K    ././gnome2
15M     ./Imágenes
210M    ./software/tor-browser_es-ES
210M    ./software
20K     ././gitkraken
820K    ././oh-my-zsh/cache
6,9M    ././oh-my-zsh/.git
580K    ././oh-my-zsh/themes
3,1M    ././oh-my-zsh/plugins
32K     ././oh-my-zsh/tools
92K     ././oh-my-zsh/lib
8,0K    ././oh-my-zsh/templates
4,0K    ././oh-my-zsh/log
20K     ././oh-my-zsh/custom
12M     ././oh-my-zsh
4,0K    ./Público
4,0K    ././mozilla/extensions
49M     ././mozilla/firefox
49M     ././mozilla
4,0K    ././gnome2_private
76K     ././thumbnails/normal
80K     ././thumbnails
4,0K    ./Música
1,1G    .

```

4.2.6 El comando `free`

Este comando es relativamente simple, presenta en la consola el uso de memoria del sistema. Posee pocos parámetros y, como en la mayoría de los comandos, el más utilizado es `-h`.

```

administrador@xubuntu1604:~$ free
              total        used        free      shared  buff/cache   available
Mem:          1016308        118480        337068         11180         560760         721232
Intercambio:    1078268            0        1078268

administrador@xubuntu1604:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:           992M         115M         329M         10M         547M         704M
Intercambio:    1,0G            0B         1,0G

administrador@xubuntu1604:~$

```

4.3 Navegación del árbol de directorios y gestión de archivos

Dentro de las actividades de la administración de sistemas es imprescindible desplazarse por el árbol de directorios y gestionar los archivos.

Todos los usuarios del sistema tienen un directorio denominado **HOME**, este directorio es el punto de arranque al iniciar sesión y donde se encuentran los archivos de personalización del usuario y los archivos que este genera en el sistema para llevar a cabo sus tareas del día a día.

Este directorio está protegido con los permisos correspondientes para que solo el usuario propietario tenga acceso al mismo y su contenido. La excepción es el usuario **root** que, como *superusuario* tiene acceso irrestricto a todos los archivos del sistema.

En la mayoría de las distribuciones *GNU/Linux*, los directorios **HOME** de los usuarios se encuentran en el directorio `/home` y el nombre del directorio personal se identifica con el nombre de usuario.

Entonces, de acuerdo a lo visto anteriormente, la ruta del directorio **HOME** del usuario está dada por la cadena `/home/<usuario>`. La excepción a esta regla es la ruta del directorio **HOME** del usuario **root** cuya ruta es `/root`.

4.3.1 Nombres de archivos y directorios

Como se describió anteriormente, los nombres de los archivos son a criterio de los usuarios y tienen pocas restricciones, no es necesario la utilización de extensiones, aunque se utilizan convencionalmente para una mejor comprensión acerca del contenido de los archivos y la asociación con aplicaciones y el tratamiento que éstas le dan a los mismos. En el caso de los directorios corren las mismas condiciones en cuanto a la denominación que le pueden dar los usuarios.

Un caso especial es la utilización del carácter `.` al inicio del nombre de un archivo o un directorio. En este caso se le agrega la propiedad, al archivo o al directorio de permanecer oculto. Generalmente se utiliza para archivos o directorios de configuraciones personales del usuario para las aplicaciones.

4.3.2 El comando `pwd`

Para desplazarnos por los directorios del sistema lo primero que se debe conocer es dónde se está ubicado. Esta información la da el comando `pwd`.

```
administrador@xubuntu1604:~$ pwd
/home/administrador
administrador@xubuntu1604:~$
```

4.3.3 El comando `ls`

Para poder desplazarse entre los directorios es necesario un conocimiento previo de los nombres de los directorios. El comando `ls` nos permite listar el contenido de los directorios.


```
administrador@xubuntu1604:~$ ls
Descargas  Documentos  Escritorio  Imágenes  Música  Plantillas  Público  Vídeos
administrador@xubuntu1604:~$
```

Este comando tiene una lista de parámetros que se combinan para lograr una visualización del listado de archivos y directorios adecuada a las necesidades del usuario.

Para ver los archivos *ocultos* el parámetro es `-a`:

```
administrador@xubuntu1604:~$ ls -a
.                .config          Imágenes          Vídeos
..               Descargas        .local            .Xauthority
.bash_history    .dmrc            Música            .Xdefaults
.bash_logout     Documentos       Plantillas        .xscreensaver
.bashrc          Escritorio       .profile          .xsession-errors
binarios         .gnupg           Público           .xsession-errors.old
.cache           .ICEauthority    .sudo_as_admin_successful
administrador@xubuntu1604:~$
```

En este caso se observan los directorios `.` y `..`, el primero hace referencia al directorio actual y el segundo al directorio padre en la jerarquía del árbol de directorios.

Otro de los parámetros más utilizados es `-l` que presenta los archivos y directorios en formato de una sola columna, incluyendo los datos de los permisos del fichero, el número de enlaces que tiene, el nombre del propietario, el del grupo al que pertenece, el tamaño (en bytes), una marca de tiempo, y el nombre del fichero. De forma predeterminada, la marca de tiempo que se muestra es la de la última modificación.

```
administrador@xubuntu1604:~$ ls -l
total 32
lrwxrwxrwx 1 administrador administrador 14 jul 11 18:30 binarios -> /usr/local/bin
drwxr-xr-x 2 administrador administrador 4096 may 11 23:59 Descargas
drwxr-xr-x 2 administrador administrador 4096 may 11 23:59 Documentos
drwxr-xr-x 2 administrador administrador 4096 may 11 23:59 Escritorio
drwxr-xr-x 2 administrador administrador 4096 may 11 23:59 Imágenes
drwxr-xr-x 2 administrador administrador 4096 may 11 23:59 Música
drwxr-xr-x 2 administrador administrador 4096 may 11 23:59 Plantillas
drwxr-xr-x 2 administrador administrador 4096 may 11 23:59 Público
drwxr-xr-x 2 administrador administrador 4096 may 11 23:59 Vídeos
administrador@xubuntu1604:~$
```

Podemos combinar los parámetros `-a` y `-l` para obtener el listado completo de archivos del directorio.

```
administrador@xubuntu1604:~$ ls -la
total 96
drwxr-xr-x 14 administrador administrador 4096 jul 11 18:30 .
drwxr-xr-x  4 root          root          4096 may 11 23:49 ..
-rw-r----- 1 administrador administrador 2225 jul 11 12:31 .bash_history
-rw-r--r--  1 administrador administrador  220 may 11 23:49 .bash_logout
-rw-r--r--  1 administrador administrador 3771 may 11 23:49 .bashrc
lrwxrwxrwx  1 administrador administrador  14 jul 11 18:30 binarios -> /usr/local/bin
```

```

drwx----- 10 administrador administrador 4096 jul 11 00:00 .cache
drwxr-xr-x  9 administrador administrador 4096 may 11 23:59 .config
drwxr-xr-x  2 administrador administrador 4096 may 11 23:59 Descargas
-rw-r--r--  1 administrador administrador  26 may 11 23:59 .dmrc
drwxr-xr-x  2 administrador administrador 4096 may 11 23:59 Documentos
drwxr-xr-x  2 administrador administrador 4096 may 11 23:59 Escritorio
drwx-----  3 administrador administrador 4096 jul 11 00:32 .gnupg
-rw-----  1 administrador administrador  0 jul 11 00:32 .ICEauthority
drwxr-xr-x  2 administrador administrador 4096 may 11 23:59 Imágenes
drwx-----  3 administrador administrador 4096 may 11 23:59 .local
drwxr-xr-x  2 administrador administrador 4096 may 11 23:59 Música
drwxr-xr-x  2 administrador administrador 4096 may 11 23:59 Plantillas
-rw-r--r--  1 administrador administrador 675 may 11 23:49 .profile
drwxr-xr-x  2 administrador administrador 4096 may 11 23:59 Público
-rw-r--r--  1 administrador administrador  0 may 12 00:00 .sudo_as_admin_succe
ssful
drwxr-xr-x  2 administrador administrador 4096 may 11 23:59 Vídeos
-rw-----  1 administrador administrador  56 jul 10 22:40 .Xauthority
-rw-r--r--  1 administrador administrador 1600 may 11 23:49 .Xdefaults
-rw-r--r--  1 administrador administrador  14 may 11 23:49 .xscreensaver
-rw-----  1 administrador administrador 359 jul 11 00:32 .xsession-errors
-rw-----  1 administrador administrador 369 jul  9 23:15 .xsession-errors.old
administrador@xubuntu1604:~$

```

El parámetro `-R` de análisis recursivo presenta la información de la siguiente manera:

```

administrador@xubuntu1604:~$ ls -R /var/log
.:
alternatives.log      btmp                gpu-manager.log     syslog.1
alternatives.log.1    btmp.1             hp                  syslog.2.gz
appport.log           cups               installer           syslog.3.gz
appport.log.1         dist-upgrade       kern.log            unattended-upgrades
apt                   dmesg              kern.log.1          upstart
auth.log              dpkg.log           kern.log.2.gz       wtmp
auth.log.1            dpkg.log.1         lastlog             wtmp.1
auth.log.2.gz         faillog            lightdm             Xorg.0.log
boot.log              fontconfig.log     speech-dispatcher   Xorg.0.log.old
bootstrap.log         fsck                syslog

./apt:
history.log  history.log.1.gz  term.log  term.log.1.gz

./cups:
access_log  access_log.1  access_log.2.gz  access_log.3.gz  error_log  page_log

./dist-upgrade:

./fsck:
checkfs  checkroot

./hp:
tmp

./hp/tmp:

./installer:
casper.log  debug  initial-status.gz  media-info  partman  syslog  version

```

```

./lightdm:
lightdm.log          lightdm.log.3.gz          x-0-greeter.log.2.gz  x-0.log.1.gz
lightdm.log.1.gz    x-0-greeter.log          x-0-greeter.log.3.gz  x-0.log.2.gz
lightdm.log.2.gz    x-0-greeter.log.1.gz    x-0.log                x-0.log.3.gz
ls: no se puede abrir el directorio './speech-dispatcher': Permiso denegado

./unattended-upgrades:
unattended-upgrades.log      unattended-upgrades-shutdown.log
unattended-upgrades.log.1.gz

./upstart:
administrador@xubuntu1604:/var/log$

```

Hay dos parámetros que se pueden usar para generar listas y utilizarlas con otro comando que trabaje en forma iterada con cada uno de los elementos.

Uno de los parámetros es el `-1` que permite visualizar los archivos y directorios en una sola columna.

```

administrador@xubuntu1604:~$ ls -1
binarios
Descargas
Documentos
Escritorio
Imágenes
Música
Plantillas
Público
Videos
administrador@xubuntu1604:~$

```

El otro comando es el `-m` que presenta los elementos en una sola línea separados por una coma.

```

administrador@xubuntu1604:~$ ls -m
binarios, Descargas, Documentos, Escritorio, Imágenes, Música, Plantillas,
Público, Videos
administrador@xubuntu1604:~$

```

Al comando `ls` se le puede dar como parámetro una cadena de texto como patrón de búsqueda:

```

administrador@xubuntu1604:~$ ls /var/log/*.log
/var/log/alternatives.log  /var/log/bootstrap.log      /var/log/kern.log
/var/log/apport.log        /var/log/dpkg.log           /var/log/Xorg.0.log
/var/log/auth.log          /var/log/fontconfig.log
/var/log/boot.log          /var/log/gpu-manager.log
administrador@xubuntu1604:~$

```

En este caso se lista los elementos incluidos en el directorio `/var/log` cuyo nombre contenga la cadena `.log` al final. El carácter `*` reemplaza a cualquier cantidad de caracteres que formen parte del nombre.

4.3.4 El comando `cd`

Este es el comando para navegar por el árbol de directorios del sistema, tiene una sintaxis muy simple y no tiene parámetros especiales.

Al ejecutar el comando `cd` (*change directory*), se le dice al sistema que se necesita cambiar de ubicación, cambiar de directorio de trabajo. El único parámetro que necesita es la *ruta* de la nueva ubicación.

```
administrador@xubuntu1604:~$ cd /var/log
administrador@xubuntu1604:/var/log$ pwd
/var/log
administrador@xubuntu1604:/var/log$
```

En el ejemplo se *cambia de directorio* de trabajo a la ubicación `/var/log`. Esto se puede comprobar ejecutando el comando `pwd`, aunque en la mayoría de las distribuciones GNU/Linux se indica en el *prompt* la ubicación actual.

En el ejemplo, la ubicación original era la carpeta **HOME** del usuario representada por el caracter `~`.

Hay parámetros especiales que permiten desplazarse sin necesidad de indicar la *ruta* completa. Uno de ellos es la referencia al directorio padre `..`, disponible en todos los directorios del sistema, que cambia al directorio inmediato superior en la jerarquía.

```
administrador@xubuntu1604:/var/log$ cd ..
administrador@xubuntu1604:/var$
```

Otro de los parámetros especiales es el caracter `-` que permite volver al directorio anterior en el que se estaba trabajando. Por ejemplo, si el directorio actual es `/var/log` y se necesita cambiar al directorio `/home/administrador`, realizar alguna tarea y luego volver al directorio `/var/log` la operatoria sería la siguiente:

```
administrador@xubuntu1604:/var/log$ cd /home/administrador/
administrador@xubuntu1604:~$

[...] tarea [...]

administrador@xubuntu1604:~$ cd -
/var/log
administrador@xubuntu1604:/var/log$
```

Otra particularidad del comando `cd` es que sin importar en cuál directorio del sistema se está trabajando, siempre se puede volver al directorio **HOME** ejecutando el comando sin parámetro.

```
administrador@xubuntu1604:/var/log$ cd
administrador@xubuntu1604:~$ pwd
/home/administrador
administrador@xubuntu1604:~$
```

El cambio de directorio de trabajo se puede realizar utilizando como parámetro la *ruta absoluta*, esto es describiendo la *ruta* indicando el camino completo por la jerarquía del árbol desde el directorio *raíz* /, o describiendo la *ruta relativa* al directorio de trabajo actual.

Algunos ejemplos:

```
administrador@xubuntu1604:/var/log$ cd cups/
administrador@xubuntu1604:/var/log/cups$ cd ../apt
administrador@xubuntu1604:/var/log/apt$ cd ../../spool
administrador@xubuntu1604:/var/spool$
```

4.3.5 El comando `mkdir`

A medida que el usuario realiza sus tareas diarias, surge la necesidad de organizar los archivos generados, esta organización se logra creando directorios específicos para dichas tareas. Si bien lo más lógico es crear la estructura de directorios desde la interfaz gráfica, puede darse el caso de tener que hacerlo desde la línea de comando.

Un trabajo que suele hacer el *Administrador de Sistemas* para una tarea imprescindible, el respaldo de la información de los usuarios. Un trabajo que se puede hacer en forma remota conectándose al equipo o local.

Para crear un directorio, es necesario que el usuario que ejecuta el comando tenga permisos de escritura en el directorio donde va a residir el nuevo. Al comando se le pasa como parámetro el nombre del nuevo directorio.

```
administrador@xubuntu1604:~$ ls
binarios  Documentos  Imágenes  Plantillas  Vídeos
Descargas Escritorio  Música    Público
administrador@xubuntu1604:~$ mkdir Respaldo
administrador@xubuntu1604:~$ ls
binarios  Documentos  Imágenes  Plantillas  Respaldo
Descargas Escritorio  Música    Público     Vídeos
administrador@xubuntu1604:~$
```

Este comando tiene un parámetro que cumple una funcionalidad importante, es el `-p`. Al utilizar este parámetro, el comando crea todos los directorios necesarios para construir la rama de directorios solicitada.

Así, para crear el directorio `/home/administrador/Respaldo/201607/Documentos` donde `201607` no existe, se utiliza el comando siguiente:

```
administrador@xubuntu1604:~$ tree Respaldo
Respaldo/
0 directories, 0 files
administrador@xubuntu1604:~$ mkdir -p Respaldo/201607/Documentos
administrador@xubuntu1604:~$ tree Respaldo
Respaldo/
|-- 201607
    |-- Documentos
2 directories, 0 files
administrador@xubuntu1604:~$
```

4.3.6 El comando `rmdir`

Este comando se utiliza exclusivamente para eliminar directorios vacíos. También se puede utilizar el parámetro `-p` para borrar una estructura de directorios donde todos están vacíos. El uso es simple, se utiliza como parámetro el nombre del directorio que se va a eliminar.

4.3.7 El comando `cp`

Entre las tareas más habituales dentro de la administración de *GNU/Linux* está la copia de archivos y directorios. El comando `cp` cumple esta función.

Para copiar un archivo es necesario indicarle al comando dos parámetros, el *origen* y el *destino*. El primero es el o los archivos o directorios que se quiere copiar y el segundo indica donde se hará la copia. En ambos casos indicando la *ruta absoluta* o necesaria para acceder a los elementos.

Para copiar un archivo en el mismo directorio se debe ejecutar:

```
administrador@xubuntu1604:~/Documentos$ ls topsecret*
topsecret.pdf
administrador@xubuntu1604:~/Documentos$ cp topsecret.pdf topsecret_copia_respaldo.pdf
administrador@xubuntu1604:~/Documentos$ ls topsecret*
topsecret_copia_respaldo.pdf  topsecret.pdf
administrador@xubuntu1604:~/Documentos$
```

Hay que tener en cuenta que en este caso los nombres deben ser diferentes.

Para copiar un grupo de archivos se puede utilizar el caracter `*` para indicar un cadena de caracteres cualquiera.

```
administrador@xubuntu1604:~$ cp Documentos/*.pdf Respaldo/201607/Documentos/
administrador@xubuntu1604:~$ tree Respaldo/
Respaldo/
|-- 201607
|   |-- Documentos
|       |-- classified.pdf
|       |-- confidential.pdf
|       |-- default.pdf
|       |-- default-testpage.pdf
|       |-- form_english.pdf
|       |-- form_russian.pdf
|       |-- secret.pdf
|       |-- standard.pdf
|       |-- topsecret_copia_respaldo.pdf
|       |-- topsecret.pdf
|       |-- unclassified.pdf
2 directories, 11 files
administrador@xubuntu1604:~$
```

Para realizar una copia de directorios se usa el parámetro `-R` o `-r` para que la tarea sea recursiva.

```
administrador@xubuntu1604:~$ cp -R Respaldo Respaldo2
administrador@xubuntu1604:~$ ls
binarios  Documentos  Imágenes  Plantillas  Respaldo  Vídeos
Descargas Escritorio  Música    Público     Respaldo2
administrador@xubuntu1604:~$ tree Respaldo
Respaldo
|-- 201607
|   |-- Documentos
|   |   |-- classified.pdf
|   |   |-- confidential.pdf
|   |   |-- default.pdf
|   |   |-- default-testpage.pdf
|   |   |-- form_english.pdf
|   |   |-- form_russian.pdf
|   |   |-- secret.pdf
|   |   |-- standard.pdf
|   |   |-- topsecret_copia_respaldo.pdf
|   |   |-- topsecret.pdf
|   |   |-- unclassified.pdf
|   |-- 2 directories, 11 files
administrador@xubuntu1604:~$ tree Respaldo2
Respaldo2
|-- 201607
|   |-- Documentos
|   |   |-- classified.pdf
|   |   |-- confidential.pdf
|   |   |-- default.pdf
|   |   |-- default-testpage.pdf
|   |   |-- form_english.pdf
|   |   |-- form_russian.pdf
|   |   |-- secret.pdf
|   |   |-- standard.pdf
|   |   |-- topsecret_copia_respaldo.pdf
|   |   |-- topsecret.pdf
|   |   |-- unclassified.pdf
|   |-- 2 directories, 11 files
administrador@xubuntu1604:~$ cp -R Respaldo Respaldo2
```

4.3.8 El comando `mv`

Este comando mueve o renombra archivos o directorios, al igual que el comando `cp`, necesita los parámetros *origen* y *destino*. Si ambos parámetros tienen como base el mismo directorio el comando renombra el elemento. Si el destino es diferente el resultado es el movimiento de los archivos o directorios.

Algunos ejemplos renombrando archivos y directorios:

```
administrador@xubuntu1604:~$ ls Documentos/
classified.pdf      form_russian.pdf  topsecret_copia_respaldo.pdf
confidential.pdf   history.log       topsecret.pdf
default.pdf        secret.pdf        unclassified.pdf
default-testpage.pdf standard.pdf
form_english.pdf   term.log
```

```

administrador@xubuntu1604:~$ mv Documentos/topsecret_copia_respaldo.pdf Document
os/topsecret_copia.pdf

administrador@xubuntu1604:~$ ls Documentos/
classified.pdf      form_english.pdf  standard.pdf      unclassified.pdf
confidential.pdf   form_russian.pdf  term.log
default.pdf        history.log       topsecret_copia.pdf
default-testpage.pdf secret.pdf        topsecret.pdf
administrador@xubuntu1604:~$

administrador@xubuntu1604:~$ ls
binarios  Documentos  Imágenes  Plantillas  Respaldo  Vídeos
Descargas Escritorio  Música    Público     Respaldo2

administrador@xubuntu1604:~$ mv Respaldo2 Respaldo_copia

administrador@xubuntu1604:~$ ls
binarios  Documentos  Imágenes  Plantillas  Respaldo  Vídeos
Descargas Escritorio  Música    Público     Respaldo_copia
administrador@xubuntu1604:~$

```

4.3.9 El comando `rm`

Otra tarea importante en la gestión de archivos y directorios es poder eliminar archivos que ya no son necesarios. Para llevar a cabo esa tarea se debe pasar como parámetro el nombre del archivo y, de ser necesario la *ruta* para llegar hasta el elemento.

También puede utilizarse el caracter `*` para hacer referencia a varios archivos.

Algunos ejemplos:

```

administrador@xubuntu1604:~/Documentos$ ls
classified.pdf  default-testpage.pdf  history.log  term.log
confidential.pdf form_english.pdf      secret.pdf  topsecret.pdf
default.pdf    form_russian.pdf      standard.pdf unclassified.pdf

administrador@xubuntu1604:~/Documentos$ rm *.log

administrador@xubuntu1604:~/Documentos$ rm secret.pdf

administrador@xubuntu1604:~/Documentos$ ls
classified.pdf  default-testpage.pdf  standard.pdf
confidential.pdf form_english.pdf      topsecret.pdf
default.pdf    form_russian.pdf      unclassified.pdf
administrador@xubuntu1604:~/Documentos$

```

Muchas veces se quiere prevenir el borrado de archivos importantes o no se está seguro de los archivos que se borrarán al ejecutar el comando para borrar varios archivos.

El parámetro `-i` provee una manera interactiva de eliminación preguntando al usuario que confirme que quiere borrar el archivo.

```

administrador@xubuntu1604:~$ rm -i Documentos/topsecret_copia.pdf
rm: ¿borrar el fichero regular 'Documentos/topsecret_copia.pdf'? (s/n) s
administrador@xubuntu1604:~$

```


Por ejemplo, se quiere borrar todos los archivos PDF de la carpeta *Documentos* excepto los archivos `form_english.pdf` y `form_russian.pdf`. El comando irá preguntando antes de eliminar cada archivo y, dependiendo de la respuesta, actuará en consecuencia y preguntará por el archivo siguiente.

```
administrador@xubuntul604:~/Documentos$ ls
classified.pdf      default-testpage.pdf  standard.pdf
confidential.pdf   form_english.pdf      topsecret.pdf
default.pdf        form_russian.pdf      unclassified.pdf

administrador@xubuntul604:~/Documentos$ rm -i *.pdf
rm: ¿borrar el fichero regular 'classified.pdf'? (s/n) s
rm: ¿borrar el fichero regular 'confidential.pdf'? (s/n) s
rm: ¿borrar el fichero regular 'default.pdf'? (s/n) s
rm: ¿borrar el fichero regular 'default-testpage.pdf'? (s/n) s
rm: ¿borrar el fichero regular 'form_english.pdf'? (s/n) n
rm: ¿borrar el fichero regular 'form_russian.pdf'? (s/n) n
rm: ¿borrar el fichero regular 'standard.pdf'? (s/n) s
rm: ¿borrar el fichero regular 'topsecret.pdf'? (s/n) s
rm: ¿borrar el fichero regular 'unclassified.pdf'? (s/n) s

administrador@xubuntul604:~/Documentos$ ls
form_english.pdf  form_russian.pdf
administrador@xubuntul604:~/Documentos$
```

Cuando se presentó el comando `rmdir` se explicó que los directorios deben estar vacíos para poder eliminarlos. El comando `rm` permite la eliminación de directorios con contenido mediante el uso del parámetro `-R`.

```
administrador@xubuntul604:~$ ls
binarios  Documentos  Imágenes  Plantillas  Respaldo  Vídeos
Descargas Escritorio  Música    Público     Respaldo_copia

administrador@xubuntul604:~$ rm -R Respaldo_copia

administrador@xubuntul604:~$ ls
binarios  Documentos  Imágenes  Plantillas  Respaldo
Descargas Escritorio  Música    Público     Vídeos
administrador@xubuntul604:~$
```

4.3.10 El comando `locate`

Esta es una herramienta que permite encontrar archivos que tengan en su nombre de archivo, incluyendo la *ruta absoluta*, la cadena de caracteres indicada en el patrón de búsqueda que se indica como parámetro.

Es importante tener en cuenta que el comando busca la información en una base de datos que se actualiza diariamente, por lo tanto, si el archivo fue creado durante el día de trabajo, no estará incluido en la base y, en el caso de los archivos eliminados durante el día estarán en la base de datos pero no existirán físicamente en el sistema. Se puede forzar la actualización de la base de datos ejecutando el comando `updatedb` para lo que se necesitan permisos especiales.

Ejemplo de uso para buscar los archivos que tengan la cadena `.pdf` en cualquier parte del nombre (incluyendo la *ruta absoluta*):

```
administrador@xubuntu1604:~$ locate .pdf
/usr/share/cups/data/classified.pdf
/usr/share/cups/data/confidential.pdf
/usr/share/cups/data/default-testpage.pdf
/usr/share/cups/data/default.pdf
/usr/share/cups/data/form_english.pdf
/usr/share/cups/data/form_russian.pdf
/usr/share/cups/data/secret.pdf
/usr/share/cups/data/standard.pdf
/usr/share/cups/data/topsecret.pdf
/usr/share/cups/data/unclassified.pdf
/usr/share/doc/fontconfig/fontconfig-user.pdf.gz
/usr/share/doc/printer-driver-foo2zjs/manual.pdf
/usr/share/doc/qpdf/qpdf-manual.pdf
/usr/share/doc/shared-mime-info/shared-mime-info-spec.pdf
/usr/share/hplip/data/ps/clean_page.pdf.gz
/usr/share/xubuntu-docs/user/C/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/C/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/de/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/de/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/es/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/es/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/fi/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/fi/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/pt/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/pt/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/ru/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/ru/xubuntu-documentation-USletter.pdf
administrador@xubuntu1604:~$
```

Después de actualizar la base de datos con el comando `sudo updatedb` la misma búsqueda incluye los archivos en el directorio **HOME** del usuario administrador.

```
administrador@xubuntu1604:~$ locate .pdf
/home/administrador/Documentos/classified.pdf
/home/administrador/Documentos/confidential.pdf
/home/administrador/Documentos/default-testpage.pdf
/home/administrador/Documentos/default.pdf
/home/administrador/Documentos/form_english.pdf
/home/administrador/Documentos/form_russian.pdf
/home/administrador/Documentos/secret.pdf
/home/administrador/Documentos/standard.pdf
/home/administrador/Documentos/topsecret.pdf
/home/administrador/Documentos/topsecret_copia_respaldo.pdf
/home/administrador/Documentos/unclassified.pdf
/home/administrador/Respaldo/201607/Documentos/classified.pdf
/home/administrador/Respaldo/201607/Documentos/confidential.pdf
/home/administrador/Respaldo/201607/Documentos/default-testpage.pdf
/home/administrador/Respaldo/201607/Documentos/default.pdf
/home/administrador/Respaldo/201607/Documentos/form_english.pdf
/home/administrador/Respaldo/201607/Documentos/form_russian.pdf
/home/administrador/Respaldo/201607/Documentos/secret.pdf
/home/administrador/Respaldo/201607/Documentos/standard.pdf
/home/administrador/Respaldo/201607/Documentos/topsecret.pdf
```

```

/home/administrador/Respaldo/201607/Documentos/topsecret_copia_respaldo.pdf
/home/administrador/Respaldo/201607/Documentos/unclassified.pdf

[...]

/usr/share/doc/qpdf/qpdf-manual.pdf
/usr/share/doc/shared-mime-info/shared-mime-info-spec.pdf
/usr/share/hplip/data/ps/clean_page.pdf.gz
/usr/share/xubuntu-docs/user/C/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/C/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/de/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/de/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/es/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/es/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/fi/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/fi/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/pt/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/pt/xubuntu-documentation-USletter.pdf
/usr/share/xubuntu-docs/user/ru/xubuntu-documentation-A4.pdf
/usr/share/xubuntu-docs/user/ru/xubuntu-documentation-USletter.pdf

```

4.4 Visualización y modificación de archivos de texto

4.4.1 El comando `file`

Como se explicó anteriormente, los nombres de archivos no tienen restricciones y no es necesario que tengan una extensión y a veces es necesario saber el tipo de archivo.

Hay que tener en cuenta que el comando `file` accede al archivo para buscar información por lo que modifica el tiempo de último acceso al archivo. Por defecto el comando `ls` muestra el tiempo de creación, para ver el tiempo de último acceso de usamos el parámetro `-u` con el comando `ls`.

```

administrador@xubuntul604:~/Documentos$ ll -u classified
-rw-r--r-- 1 administrador administrador 979 jul 12 12:18 classified

administrador@xubuntul604:~/Documentos$ file classified
classified.pdf: PDF document, version 1.2

administrador@xubuntul604:~/Documentos$ ll -u classified
-rw-r--r-- 1 administrador administrador 979 jul 12 13:00 classified

```

Para evitar ese comportamiento se utiliza el parámetro `-p` que preserva el estado del archivo.

```

administrador@xubuntul604:~/Documentos$ ll -u form_english.pdf
-rw-r--r-- 1 administrador administrador 276070 jul 12 09:41 form_english.pdf

administrador@xubuntul604:~/Documentos$ file -p form_english.pdf
form_english.pdf: PDF document, version 1.4

administrador@xubuntul604:~/Documentos$ ll -u form_english.pdf
-rw-r--r-- 1 administrador administrador 276070 jul 12 09:41 form_english.pdf

```

4.4.2 El comando `cat`

Una tarea del administrador de sistemas es la de configurar los servicios y aplicaciones del sistema, donde casi todos los archivos de configuración son archivos de texto plano que se pueden modificar desde la línea de comando.

Una forma de ver el contenido de esos archivos es usar el comando `cat` que visualiza el contenido del archivo de texto en la salida estándar:

```
administrador@xubuntu1604:~$ cat /etc/hosts
127.0.0.1    localhost
127.0.1.1    xubuntu1604

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
administrador@xubuntu1604:~$

administrador@xubuntu1604:~/Documentos$ cat lorem.txt
Dolorum dolorem in eius omnis temporibus suscipit. Dolorum et cupiditate commodi
sapiente et possimus ratione. Quasi ratione laborum in vitae in optio architect
o. Saepe dignissimos sed deleniti atque. Et ratione reiciendis rerum qui.

Tempora aperiam rerum deleniti vel. Explicabo voluptas assumenda ex. Harum esse
repellendus sit qui vitae qui error. Est voluptatem rem sed dolore voluptas. Et
assumenda pariat consequatur rerum labore sit quia. Quia voluptatem aut non oc
caecati eum mollitia voluptate.

Voluptas nam minus sint harum eos dolor qui qui. Itaque quo non voluptas amet om
nis numquam molestiae. Est sit voluptate similique. Molestiae non laboriosam nis
i et dolorem impedit. Quae aut aut sunt quia nulla iste et nulla.

Sapiente optio optio saepe minima hic eos. Sapiente sequi sunt quia sunt id anim
i aut. Harum veritatis animi laboriosam nihil quo ut et incidunt. Magni est vel
est nam voluptas repellendus illo esse.

Et quas doloribus iure. Et porro occaecati laborum consequuntur. Aut veritatis o
dio deleniti consequatur.

administrador@xubuntu1604:~/Documentos$
```

En el caso de visualizar un archivo que contiene código fuente es interesante la posibilidad de que las líneas de texto estén numeradas para una mejor interpretación del contenido. Para esto se utiliza el parámetro `-n` que imprime el número de línea.

```
administrador@xubuntu1604:~$ cat -n /usr/bin/lorem
```

```
1      #!/usr/bin/perl -w
2
3      eval 'exec /usr/bin/perl -w -S $0 ${1+"$@"}'
4          if 0; # not running under some shell
5
6      use strict;
```

```

7      use vars qw($opt_v $opt_w $opt_s $opt_p);
8
9      use Getopt::Std;
10     use Text::Lorem;
11
12     getopts("vw:s:p:");
13
14     if ($opt_v) {
15         print usage();
16         exit 0;
17     }
18
19     die usage()
20         if ((defined($opt_w) + defined($opt_s) + defined($opt_p)) > 1);
21
22     my $lorem = Text::Lorem->new;
23     if ($opt_w) {
24         print $lorem->words($opt_w);
25     }
26     elsif ($opt_s) {
27         print $lorem->sentences($opt_s);
28     }
29     elsif ($opt_p) {
30         print $lorem->paragraphs($opt_p);
31     }
32     else {
33         print $lorem->paragraphs(1);
34     }
35
36     sub usage {
37         return <<USAGE;
38     $0 - Generate random Latin looking text using Text::Lorem
39
40     Usage:
41         $0 -w NUMBER_OF_WORDS
42         $0 -s NUMBER_OF_SENTENCES
43         $0 -p NUMBER_OF_PARAGRAPHS
44
45     -w, -s, and -p are mutually exclusive.
46     USAGE
47     }
48
49     __END__
50
51     =head1 NAME
52
53     lorem - Generate random Latin looking text using Text::Lorem
54
55     =head1 SYNOPSIS
56
57     Generate 3 paragraphs of Latin looking text:
58
59         $ lorem -p 3
60
61     Generate 5 Latin looking words:
62
63         $ lorem -w 5
64
65     Generate a Latin looking sentence:

```

```

66
67     $ lorem -s 1
68
69     =head1 DESCRIPTION
70
71     F<lorem> is a simple command-line wrapper around the C<Text::Lorem>
72     module. It provides the same three basic methods: Generate C<words>,
73     generate C<sentences>, and generate C<paragraphs>.
74
75

```

4.4.3 Los comandos `head` y `tail`

Estos comandos son similares a `cat` pero lo que hacen es mostrar una porción del contenido del texto, en el caso de `head` muestra las primeras líneas y en el caso de `tail` las últimas. Por defecto visualizan las diez primeras líneas, pero se puede personalizar la cantidad utilizando el parámetro `-n`.

```
administrador@xubuntul604:~$ head /usr/bin/lorem
```

```

#!/usr/bin/perl -w

eval 'exec /usr/bin/perl -w -S $0 ${1+"$@"}'
    if 0; # not running under some shell

use strict;
use vars qw($opt_v $opt_w $opt_s $opt_p);

use Getopt::Std;
use Text::Lorem;

```

```
administrador@xubuntul604:~$ head -n 6 /usr/bin/lorem
```

```

#!/usr/bin/perl -w

eval 'exec /usr/bin/perl -w -S $0 ${1+"$@"}'
    if 0; # not running under some shell

use strict;

```

```
administrador@xubuntul604:~$ tail /usr/bin/lorem
```

```

    $ lorem -s 1

=head1 DESCRIPTION

F<lorem> is a simple command-line wrapper around the C<Text::Lorem>
module. It provides the same three basic methods: Generate C<words>,
generate C<sentences>, and generate C<paragraphs>.

administrador@xubuntul604:~$

```

```
administrador@xubuntu1604:~$ tail -n 5 /usr/bin/lorem
F<lorem> is a simple command-line wrapper around the C<Text::Lorem>
module. It provides the same three basic methods: Generate C<words>,
generate C<sentences>, and generate C<paragraphs>.

administrador@xubuntu1604:~$
```

4.4.4 El comando `less`

Los comandos vistos para mostrar el contenido de archivos de texto se enfrentan con el problema de los archivos grandes que muchas veces no se alcanzan a visualizar en la pantalla de la salida estándar, no se puede buscar o recorrer el contenido.

La solución es el comando `less` que permite una visualización paginada con muchas opciones de navegación del documento como el desplazamiento con las teclas de cursos y avance y retroceso de página, realizar búsqueda de texto dentro del archivo, o abrir el archivo en el editor de texto plano definido en el sistema.

La sintaxis es simple y el único parámetro que necesita es la ruta y nombre del archivo que se quiere visualizar.

```
administrador@xubuntu1604:~$ less /usr/bin/lorem
```

4.4.5 El editor `vi`

Para crear o modificar un archivo de texto es necesario un editor. En el entorno gráfico hay muchas opciones como **gedit** en *Debian* y *Ubuntu* o **Mousepad** en *Xubuntu*. En el caso de editores en la terminal de *GNU/Linux* también existen opciones, entre ellas **nano**, **vi** y **emacs**.

Si bien el editor por defecto en la mayoría de las distribuciones es **nano**, se utilizará **vi** para los ejemplos que se darán en la materia.

Se sugiere la instalación del paquete `vim-nox`.

5 Bibliografía

La bibliografía que se indica en esta sección corresponde a material que provee una visión interesante sobre los temas propuestos en esta unidad. Es documentación más completa e incluso más extensiva en el desarrollo de algunos temas.

The Debian Administrator's Handbook, Raphaël Hertzog and Roland Mas, (<https://debian-handbook.info/>)

Administración de sistemas GNU/Linux, Máster universitario en Software Libre, Universitat Oberta de Catalunya

Básicamente GNU/Linux, Antonio Perpiñan, Fundación Código Libre Dominicano (<http://www.codigolibre.org>)

- 1 https://en.wikipedia.org/wiki/Master_boot_record
- 2 https://en.wikipedia.org/wiki/GUID_Partition_Table
- 3 <http://www.uefi.org/>
- 4 https://en.wikipedia.org/wiki/Globally_unique_identifier
- 5 <https://www.gnu.org/software/grub/>
- 6 https://en.wikipedia.org/wiki/List_of_disk_partitioning_software
- 7 <http://www.rodsbooks.com/gdisk/>
- 8 <https://www.gnu.org/software/parted/>
- 9 <http://gparted.org>
- 10 <http://gparted.org/livecd.php>
- 11 <http://gparted.org/display-doc.php?name=gparted-live-manual>
- 12 https://en.wikipedia.org/wiki/File_system
- 13 https://es.wikipedia.org/wiki/Sistema_de_archivos
- 14 https://es.wikipedia.org/wiki/Filesystem_Hierarchy_Standard
- 15 https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard