



CST

Cloud Services & Training

MÉXICO



FUNDAMENTOS LINUX

- Este curso esta dirigido a todo aquel con poca experiencia en Linux y que estén interesados en aprender sobre esta plataforma

UN POCO DE HISTORIA

En 1991, con 23 años, un estudiante de informática de la Universidad de Helsinki (Finlandia) llamado **Linus Torvalds** se propone como entretenimiento hacer un sistema operativo que se comporte exactamente igual al sistema operativo UNIX, pero que funcione sobre cualquier ordenador compatible PC. Posteriormente Linus tuvo que poner como requisito mínimo que el ordenador tuviese un procesador i386, ya que los ordenadores con CPU más antiguas no facilitaban el desarrollo de un sistema operativo compatible con UNIX.

El 5 de octubre de 1991, Linus anuncia la primera versión “oficial” de Linux, la **0.02**. Esta versión ya podía ejecutar dos herramientas básicas de GNU: el intérprete de órdenes (*bash*) y el compilador de C (*gcc*). Linux no tenía aún nada sobre soporte a usuarios, distribuciones, documentación ni nada parecido

¿ QUE ES GNU ?

- Proyecto iniciado en 1984 por Richard Stallman para obtener un sistema operativo completo, totalmente libre, compatible con el sistema operativo Unix.
- GNU es una colección de programas informáticos y un sistema operativo de tipo Unix, desarrollado por y para el Proyecto GNU, y auspiciado por la Free Software Foundation. Está formado en su totalidad por software libre, mayoritariamente bajo términos de *copyleft*. *GNU* es el acrónimo recursivo de "**GNU's Not Unix**" (*en español*: GNU no es Unix), nombre elegido debido a que GNU sigue un diseño tipo Unix y se mantiene compatible con éste, pero se distingue de Unix por ser software libre y por no contener código de Unix.
- Un sistema operativo incluye ambientes de trabajo (*shells*) compiladores, editores, formateadores de texto, correo, etc. Con el tiempo el proyecto GNU tenía una colección impresionante de estos programas. Solo faltaba el núcleo.

¿ PARA QUE SIRVE FSF (FREE SOFTWARE FOUNDATION) ?

- Fundada en 1985 por Richard Stallman para promover el derecho a usar, estudiar, copiar modificar y redistribuir programas.
- La FSF se concentra en el desarrollo de software libre nuevo, y en hacerlo coherente de modo que se elimine la necesidad de usar software propietario.
- Además de desarrollar software, la FSF protege, promueve y preserva software libre.
- Los fondos de la FSF provienen de la distribución de CD's y manuales de software GNU, además de donaciones deducibles de impuestos (en EU).

¿ QUÉ ES COPYLEFT Y GPL ?

- El **copyleft** es un método general para liberar un programa u otro tipo de trabajo (en el sentido de libertad, no de gratuidad), que requiere que todas las versiones modificadas y extendidas sean también libres. El copyleft también ayuda a los programadores que quieran contribuir con mejoras al software libre a obtener permiso para hacerlo.
- La **Licencia Pública General “GNU GPL”**, es una licencia de derecho de autor ampliamente usada en el mundo del software libre y código abierto, y garantiza a los usuarios finales (personas, organizaciones, compañías) la libertad de usar, estudiar, compartir (copiar) y modificar el software. Su propósito es doble: declarar que el software cubierto por esta licencia es libre, y protegerlo (mediante una práctica conocida como copyleft) de intentos de apropiación que restrinjan esas libertades a nuevos usuarios cada vez que la obra es distribuida, modificada o ampliada. Esta licencia fue creada originalmente por **Richard Stallman** fundador de la Free Software Foundation (FSF) para el proyecto GNU.

CARACTERISTICAS DE LINUX

- **Estable** meses (inclusive años) de funcionamiento ininterrumpido.
- **Seguro** prácticamente sin virus, un proceso no puede acceder a áreas de memoria ajenas.
- **Multitarea**.
- **Multiusuario**.
- **Compatible** cumple el estándar POSIX como otros muchos UNIX, lo que facilita la migración entre plataformas.
- **Multi-arquitectura** disponible para Intel y compatibles, PowerPC, Macintosh, DEC Alpha, Sun Sparc, ARM
- **Rápido** maneja eficientemente los recursos: memoria, disco duro, CPU, etc.
- **Capacidad en red excepcional**; fue desarrollado desde un principio para la conexión en red.
- **Elegante** es un modelo de programación: pequeño, extensible, modular.

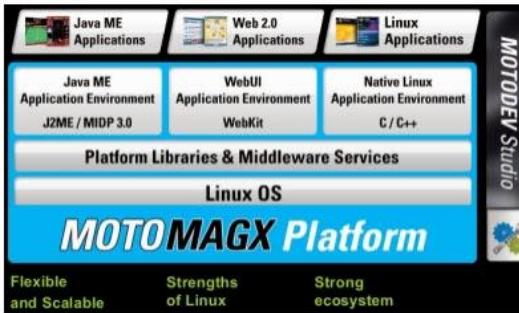
DISTRIBUCIONES DE LINUX

Una distribución es una recopilación de software ya compilado y empaquetado para facilitar su instalación y configuración. El gestor de paquetes permite ver una breve descripción, registra qué programas están instalados, verifica las dependencias y conflictos (al instalar o al desinstalar) para que todo funcione siempre. Incluso actualiza a las nuevas versiones a través de la red de manera transparente.



¿ QUE DISTRIBUCIÓN ELEGIR ?

- Distribuciones para Público en General.
 - Ubuntu
 - Fedora
- Distribuciones para móviles.
 - OpenMoko
 - Android
 - MotoMagx



¿ QUE DISTRIBUCIÓN ELEGIR ?

- Distribuciones para Profesionales.
 - Ubuntu Studio (para artistas y creadores multimedia)
 - Kali Linux (para hackers y profesionales de la seguridad informática)
 - CAELinux (para arquitectos e ingenieros)
- Distribuciones Especializadas.
 - Centos
 - Red Hat
 - Suse
 - Gentoo

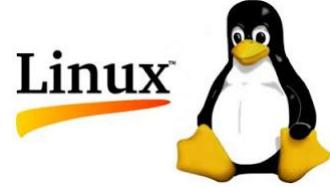
TECLADO

Algunas teclas imprescindibles en modo texto:

- **ctrl+alt+Fn** pasa al terminal virtual *n*. *X* suele estar en el 7.
- **ctrl+alt+(num+)** cambia la resolución de *X*.
- **ctrl+alt+(retroceso)** aborta la sesión *X* y todos los programas en ella. Puede salvarnos de un congelamiento de *X* sin perder información.
- **ctrl+C** interrumpe un programa (en terminal).
- **ctrl+S** congela un programa. Se descongela con **ctrl+Q**.
- **ctrl+Z** suspende un programa. Se reanuda su ejecución con **fg (foreground)** o con **bg (background)**.
- **(flecha arriba)** recuperas los comandos anteriormente escritos en la línea de comandos.
- **tab** puede ayudarte a completar un comando o el nombre de un archivo.
- **shift+RePág** permite hacer un *scroll* de la pantalla del terminal.
- **ctrl+R** hace una búsqueda hacia atrás entre los comandos escritos en el terminal.



INICIO DE SESIÓN Y PRIMEROS COMANDOS



INTRODUCCIÓN

En esta sección el usuario conocerá los comandos básicos y como emplearlos en modo de terminal (shell), así como el uso de estos y el entendimiento de los mismos

CONSOLAS Y TERMINALES LINUX

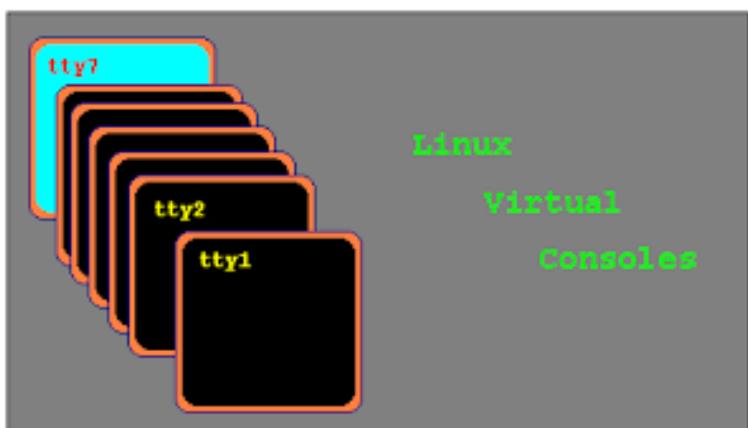
Los sistemas operativos basados en Unix disponen de un intérprete de comandos u órdenes (conocido como terminal, consola o shell) que hace de interfaz entre el usuario y el propio sistema operativo. Es decir, mediante la terminal o consola podemos acceder al sistema operativo sin utilizar la interfaz gráfica y realizar todo tipo de tareas en modo texto. *La terminal o consola es una herramienta poderosa capaz de realizar tareas de forma más rápida que mediante la interfaz gráfica del sistema operativo.*

Algunos ejemplos de terminales:

- Xterm
- Gnome-Terminal
- Tmux
- Tilda
- Nautilus Embedded Terminal
- Eterm

CONSOLAS VIRTUALES

Linux soporta la idea de consola virtual lo que permite iniciar varias sesiones con la misma o diferente identificación. Con esto, la consola (el teclado y el monitor) puede trabajar como varias terminales simultáneamente. Soporta hasta 6 consolas virtuales entre las que se puede alternar oprimiendo la tecla ALT y una de las teclas de función F1-F6.





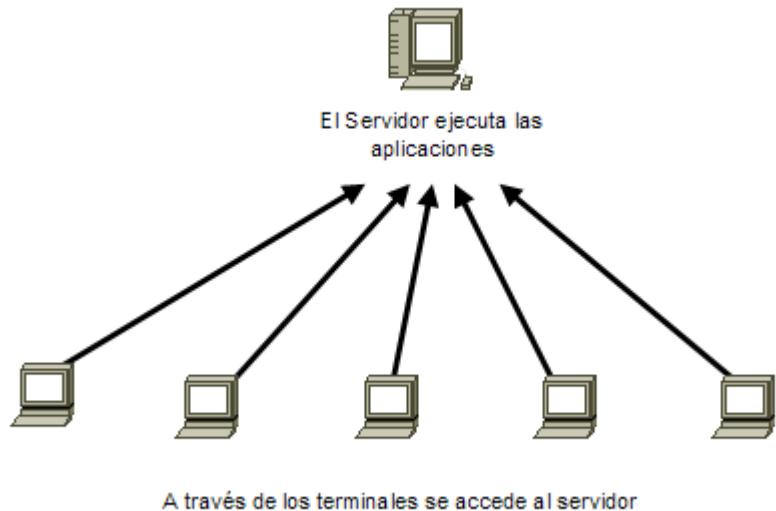
EMULADORES DE TERMINALES

Un **emulador de terminal**, **emulador de consola** o **shell** es un **programa informático** que simula el funcionamiento de un terminal de computadora en cualquier dispositivo de visualización. Normalmente suele estar accesible pulsando **CRTL + ALT + T** simultáneamente.

Los emuladores de terminal incorporan características tales como control de procesos, redirección de entrada/salida, listado y lectura de ficheros, protección, comunicaciones y un lenguaje de órdenes para escribir secuencias de instrucciones por lotes o (scripts o guiones). Uno de los lenguajes o intérpretes más conocidos, es el Bourne Shell, el cual fue el intérprete usado en las primeras versiones de [Unix](#) y se convirtió en un estándar de facto.

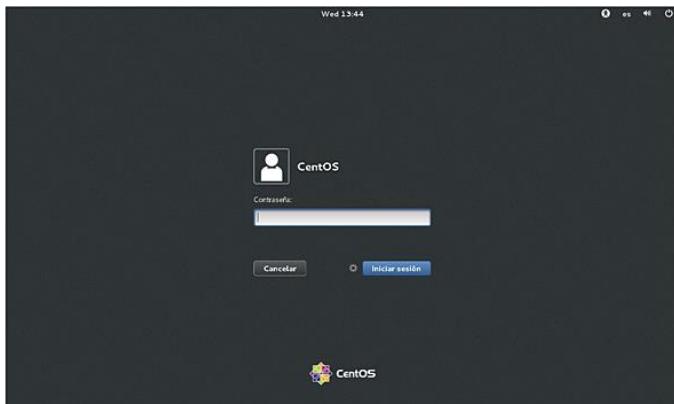
TERMINALES REMOTOS

Un servidor de terminales, es un servidor que ejecuta un software que permite a los usuarios acceder al mismo remotamente desde otros PCs (que hacen de terminales) y manejarlo como si estuvieran sentados frente al servidor.



INICIO DE SESIÓN Y AUTENTICACIÓN

El sistema Linux es multiusuario, la autenticación sirve para comprobar que el usuario situado delante de la consola es quien pretende ser. Tiene que facilitar su identidad (*nombre de inicio de sesión o login*) y una prueba de esta (*contraseña*) para conectarse. Tanto si se trata del modo texto como del modo gráfico, la contraseña escrita por el usuario no se muestra en pantalla

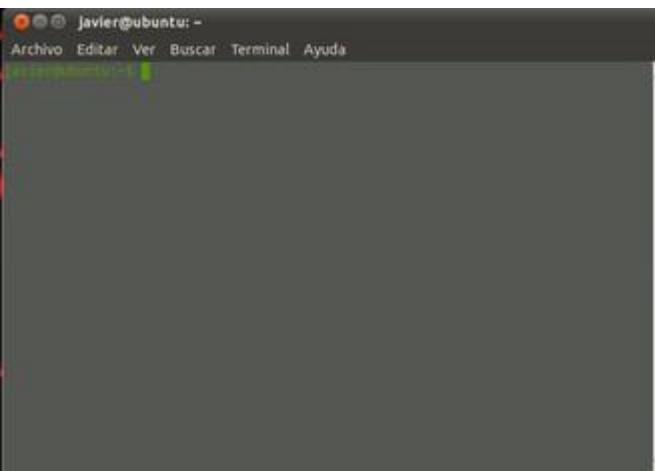


EMULADORES DE TERMINALES

El prompt, generalmente, contiene el carácter \$ para los usuarios sin privilegios y # para el administrador, además de proporcionar más información. Por ejemplo: jou@home-comp:/media/disk\$ donde el usuario en uso es jou, el nombre del equipo es home-comp, y el directorio de trabajo es /media/disk

En sistemas Unix/Linux, el **prompt** tiene el aspecto de:

- *usuario@máquina:~\$*
- *root@máquina:~#*



SINTAXIS DE LOS COMANDOS



Comando/Sintaxis	Descripción	Ejemplos
<code>cat fich1 [..,fichN]</code>	Concatena y muestra un archivos	<code>cat /etc/passwd</code>
	archivos	<code>cat dict1 dict2 dict</code>
<code>cd [dir]</code>	Cambia de directorio	<code>cd /tmp</code>
<code>chmod permisos fich</code>	Cambia los permisos de un archivo	<code>chmod +x miscript</code>
<code>chown usuario:grupo fich</code>	Cambia el dueño un archivo	<code>chown nobody miscript</code>
<code>cp fich1..,fichN dir</code>	Copia archivos	<code>cp foo foo.backup</code>
<code>diff [-e]arch1 arch2</code>	Encuentra diferencia entre archivos	<code>diff foo.c newfoo.c</code>
<code>du [-sabr] fich</code>	Reporta el tamaño del directorio	<code>du -s /home/</code>
<code>file arch</code>	Muestra el tipo de un archivo	<code>file arc_desconocido</code>
<code>find dir test acción</code>	Encuentra archivos.	<code>find . -name ``.bak'' -print</code>
<code>grep [-cilnv] expr archivos</code>	Busca patrones en archivos	<code>grep mike /etc/passwd</code>
<code>head -count fich</code>	Muestra el inicio de un archivo	<code>head prog1.c</code>
<code>mkdir dir</code>	Crea un directorio.	<code>mkdir temp</code>
<code>mv fich1 ..,fichN dir</code>	Mueve un archivo(s) a un directorio	<code>mv a.out prog1</code>
<code>mv fich1 fich2</code>	Renombra un archivo.	<code>mv .c prog_dir</code>
<code>less / more fich(s)</code>	Visualiza página a página un archivo.	<code>more muy_largo.c</code>
	less acepta comandos vi.	<code>less muy_largo.c</code>
<code>ln [-s] fich acceso</code>	Crea un acceso directo a un archivo	<code>ln -s /users/mike/.profile .</code>

USO DEL RATÓN EN LINUX

BOTÓN DEL RATÓN	SINTAXIS
Botón Izquierdo	Se usa para: <ul style="list-style-type: none">Selección de textosElegir elementosArrastrar elementos
Botón Central	Se usa para: <ul style="list-style-type: none">Pegar textoMover elementos
Botón Derecho	Abre un menú del objeto seleccionado, si existe

ATAJOS DEL TECLADO



COMBINACIÓN	ACCIÓN
Ctrl + alt + l	Bloqueo de Sesión
Ctrl + alt + flechas dirección arriba y abajo	Desplazamiento entre los diferentes escritorios
Ctrl + alt + shift + flechas de dirección arriba y abajo	Posicionamiento de escritorios
Ctrl + alt + t	Abre una terminal.
Alt + tab	Sirve para moverse entre las aplicaciones que se estén ejecutando.
Command o tecla de Windows + a	Lista todas las aplicaciones
Ctrl + Command o tecla de Windows + flechas de dirección arriba y abajo	Maximiza o minimiza la ventana.
Alt + f1	Abre el menú de búsqueda



PRIMEROS COMANDOS



IDENTIDAD DE LOS USUARIOS

- **WHO**, Muestra información de los usuarios conectados al sistema.
- **WHOAMI**, muestra el nombre del usuario (userid, username) que ejecuta el comando.
- **FINGER**, muestra el nombre de acceso del usuario, el nombre real, el nombre del terminal y el estado de escritura (como un "*" después del nombre de terminal si el permiso de escritura está denegado), el tiempo parado, la hora de acceso, la localización de la oficina y el teléfono de la oficina.

```
finger -s hiox

Login      Name      Tty      Idle  LoginTime  Office OfficePhone
HIOX      HIOX INDIA  *:0        Sep 14 09:07
HIOX      HIOX INDIA  *pts/0     9 Sep 14 09:08
HIOX      HIOX INDIA  *pts/1    1:29 Sep 14 09:12
```

Ejemplo de comando **Finger**

CAMBIO DE CONTRASEÑA PASSWD

El comando passwd se utiliza para cambiar la contraseña del usuario que se pasa como argumento. Se describe la sintaxis y alguna de las opciones de este comando.

Su sintaxis es de la siguiente forma:

`passwd [opciones] [USUARIO]`

OPCIÓN	DESCRIPCIÓN	OPCIÓN	DESCRIPCIÓN
-a	Informa del estado de las contraseñas de todas las cuentas	-d	Borra la contraseña para la cuenta indicada.
-e	Fuerza a que la contraseña de la cuenta caduque	-h	Muestra mensaje de ayuda y termina
-k	Cambia la contraseña solo si ha caducado	-l	Bloquea la contraseña de la cuenta indicada.
-q	Modo silencioso	-S	Informa del estado de la contraseña la cuenta indicada

CONTEO WC

WC (word count) es un comando utilizado en el sistema operativo Unix que permite realizar diferentes conteos desde la entrada estándar, ya sea de palabras, caracteres o saltos de líneas. El programa lee la entrada estándar o una lista concatenada y genera una o más de las estadísticas siguientes: conteo de líneas, conteo de palabras, y conteo de bytes. Si se le pasa como parámetro una lista de archivos, muestra estadísticas de cada archivo individual y luego las estadísticas generales.

Opciones del comando WC

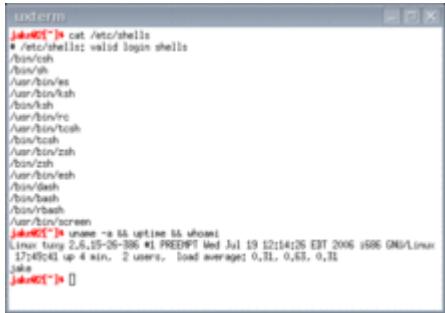
```
wc -l <archivo> número de líneas
wc -c <archivo> número de bytes
wc -m <archivo> imprime el número de caracteres
wc -L <archivo> imprime la longitud de la línea más larga
wc -w <archivo> imprime el número de palabras
```

Ejemplo de uso comando WC

```
$ wc ideas.txt excerpt.txt
      40      149     947 ideas.txt
    2294   16638   97724 excerpt.txt
    2334   16787   98671 total
```

VISUALIZACIÓN (CLEAR, ECHO)

- **CLEAR**, es un comando usado en LINUX, para borrar o limpiar la pantalla.



- **ECHO**, es un comando para la impresión de un texto en pantalla. Es utilizado en las terminales de los sistemas operativos como Unix, GNU/Linux, o MS-DOS; dentro de pequeños programas llamados scripts; y en ciertos lenguajes de programación tales como PHP

TIEMPO (DATE, CAL)

COMANDO	DESCRIPCIÓN	COMANDO	DESCRIPCION
DATE	El comando date es un comando existente en sistemas Unix y tipo unix que muestra la hora y la fecha del sistema	CAL	El comando cal se utiliza para mostrar el calendario

Mostrar la fecha en UTC:

```
Pepes-MacBook-Air:~ pepe$ date -u
Tue Jan 27 20:22:08 UTC 2015
```

Mostrar los segundos transcurridos entre unix epoch:

```
Pepes-MacBook-Air:~ pepe$ date +%
1422390213
```

OPCIONES:

-1	Muestra un sólo mes como salida.
-3	Muestra el mes previo/actual/siguiente como salida.
-s	Muestra el domingo como primer día de la semana.
-m	Muestra el lunes como primer día de la semana.
-j	Muestra fechas julianas (días ordenados, numerados desde el 1 de Enero).
-y	Muestra un calendario para el año actual.

CERRAR SESIÓN

A continuación se muestran unas de las opciones vía línea de comandos para cerrar sesión en Linux:

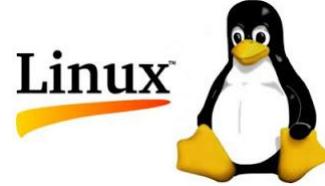
- **Shutdown -r now**
- **Reboot**
- **Init 6**

También podemos programar el reinicio de la siguiente forma:

- **Shutdown -r +10** >>> Reiniciará en 10 minutos.
- **Shutdown -r 14:30** >>> Reiniciará el equipo a las 14:30



DOCUMENTACIÓN



INTRODUCCIÓN

GNU/Linux se distingue por la calidad y cantidad de documentación que incluye, desde manuales, “COMOs”, guías, libros, el usuario tiene todo a su disposición para informarse de cada detalle del sistema. Existe un manual por casi cada comando del sistema. La forma de consultar esta documentación (en una terminal de texto) es mediante el comando ***man*** seguido del comando del que se quiere obtener información. Ejemplo: ***man mount***.

El comando ***info*** es similar, pero en general es más completo y tiene un formato hipertexto (más sencillo que el html).

Los ***Howtos*** (“COMOs”) son una especie de recetas donde se explican paso a paso diversas tareas del sistema.

SECCIONES MANUAL

El manual normalmente está dividido en ocho secciones numeradas, organizadas como sigue (en BSD, Unix y Linux):

SECCIÓN	DESCRIPCIÓN
1	Comandos Generales
2	Llamadas al Sistema
3	Biblioteca C de Funciones
4	Ficheros Especiales (normalmente dispositivos, que se pueden encontrar en /dev) y drivers
5	Formato de ficheros y convenciones
6	Juegos y Salvapantallas
7	Miscelánea
8	Comandos de administración del sistema y demonios.

/ETC/MAN.CONF

Es un archivo usado para configurar la ruta de búsqueda manual, configuraciones regionales, y conjunto de utilidades para el comando man. Durante la inicialización, el comando *man* lee los archivos de configuración ubicados en */usr/local/etc/man.d/*.conf* y */etc/man.conf*

Los archivos contenidos en */usr/local/etc/man.d/*.conf* están destinados a ser utilizado por el sistema de **puertos** para extender el conjunto de manuales para admitir caminos regionales y locales. *etc/man.conf* está destinado a ser utilizado por el administrador local para establecer la política adicional.

DOCUMENTACION COMANDO INFO

El comando *info* nos sirve cuando el texto de ayuda de *man* es insuficiente, o directamente no existe una entrada en su base de datos sobre lo que buscamos. El programa *info* tiene una interfaz navegable: muchas de las páginas tienen enlaces a otras páginas relacionadas, muy similar a como tienen las páginas web. Los enlaces se indican mediante un asterisco (*) seguido por el título del enlace, seguido por un doble dos puntos (::). Para seguir los enlaces, hay que situar el cursor sobre ellos y pulsar la tecla <INTRO>. Para ayudar la navegación por los documentos, también están las siguientes teclas:

- **b:** para ir un paso hacia atrás, a la última página visitada
- **u:** para ir al nivel superior en el árbol de páginas
- **n:** para ir a la siguiente página
- **p:** para ir a la página previa

OPCIÓN –HELP DE LOS COMANDOS LINUX

Uno de los grandes aliados para conseguir información sobre el uso de un comando de la forma más rápida posible. Es más rápido de usar que **man** o **info** porque está muy sintetizado. Podemos obtener una breve descripción de un comando y la sintaxis de las opciones que permite, así sabremos rápidamente como utilizarlo.

Ejemplo de uso

man -help

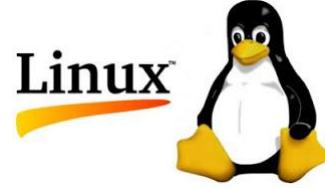
o bien

man -h

```
alejandro@alejandro-Ubuntu:~$ man --help
Uso: man [OPCIÓN...] [SECCIÓN] PÁGINA...
      -C, --config-file=ARCHIVO  use este archivo de configuración de usuario
      -d, --debug                emitir mensajes de depuración
      -D, --default              restablecer todas las opciones a sus valores
                                 predeterminados
      --warnings[=AVISOS]        activa avisos de groff

Modos principales de operación:
      -f, --whatis             equivalente a whatis
      -k, --apropos             equivalente a apropos
      -K, --global-apropos     buscar el texto en todas las páginas
      -l, --local-file          interpreta los argumentos de PAGE como archivos
                                 locales
      -w, --where, --path, --location
                                 imprimir la localización física de las páginas
                                 de manual
      -W, --where-cat, --location-cat
                                 imprimir la localización física del archivo cat
      -c, --catman              usado por catman para reformatear páginas de cat
                                 desactualizadas
      -R, --recode=CODIFICACIÓN página fuente de salida codificada en ENCODING

Buscando páginas de manual:
      -L, --locale=LOCAL        define el local para esta particular búsqueda de
                                 manual
      -m, --systems=SISTEMA    use páginas de manual de otros sistemas
      -M, --manpath=RUTA        establecer la ruta de búsqueda para las páginas
                                 de manual a PATH
      -S, -s, --sections=LISTA  usar lista de sección separada por dos puntos
      -e, --extension=EXTENSIÓN limitar la búsqueda a extensiones de tipo
```



DOCUMENTACION HOW TO

HOWTO o **how-to** (del inglés "**cómo hacer**") es un documento informal, generalmente corto, que describe **cómo** cumplir con una cierta tarea. Esta palabra es propia de la jerga informática o electrónica. En español es también habitual el uso de **como**.

Son generalmente creados para ayudar a personas en un problema y suelen dejar de lado detalles para expertos, por lo que son en general un resumen del tema tratado. También es un término muy utilizado en foros y dedicado a personas que prefieren abrir un hilo nuevo para preguntar algo en vez de molestar en buscarlo por internet



/USR/Sshare/DOC

El sistema de archivos **/usr** es con frecuencia grande, debido a que todos los programas están instalados allí. Normalmente, todos los archivos en **/usr** provienen de la distribución Linux que hayamos instalado; los programas instalados localmente y algunas otras cosas se encuentran bajo **/usr/local**. De esta manera es posible actualizar el sistema desde una nueva versión de la distribución, o incluso desde una distribución completamente nueva, sin tener que instalar todos los programas nuevamente.

/usr/share/man, /usr/share/info, /usr/share/doc

Páginas de manual, documentos de información GNU, y archivos de documentación de los programas instalados



SITIOS DE INTERNET

GNU/Linux dispone de infinidad navegadores y servidores web, de correo, noticias, etc. Cualquier protocolo y utilidad que busques, seguro que está disponible en GNU/Linux. . . un sistema operativo nacido con Internet.

ÁRBOL DE LINUX



La estructura de los directorios de Linux, así como su contenido y funciones, viene definida en el denominado ***Filesystem Hierarchy Standard o FHS*** por sus siglas en inglés, que en otras palabras viene a ser el estándar de jerarquía para los sistemas de archivos en sistemas Linux y otros derivados de UNIX. El **FHS** nació en 1994 y es actualmente mantenida por la Linux Foundation.

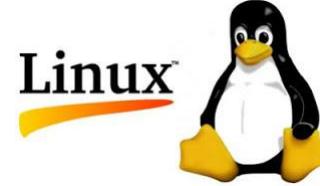
	/
bin	Binarios de usuario
boot	Ejecutables y archivos requeridos para el arranque
dev	Archivos de información de todos los volúmenes
etc	Archivos de configuración del sistema y de aplicaciones
home	Directorio personal con las carpetas de usuario
lib	Bibliotecas necesarias para la ejecución de binarios
media	Directorio de montaje de volúmenes extraíbles
opt	Ficheros de aplicaciones externas que no se integran en /usr
proc	Ficheros de información de procesos
root	Directorio personal de superusuario
sbin	Binarios de sistema
srv	Archivos relativos a servidores web, FTP, etc.
sys	Archivos virtuales con información de eventos del sistema
tmp	Directorio de ficheros temporales
usr	Archivos de programas y aplicaciones instaladas
var	Archivos de variables, logs, emails de los usuarios del sistema, etc.

MONTAR Y ACCEDER A LOS MEDIOS EXTRAIBLES

- *mount* es un comando de sistemas basados en Unix que se utiliza para [montar dispositivos y particiones](#) para su uso por el [sistema operativo](#). Montar es hacer que el sistema operativo proyecte el contenido de ese dispositivo o partición en un enlace lógico (un directorio). Cuando se desocupa se rompe el enlace y se sigue trabajando con los mismos archivos básicos.
- *umount*, este comando permite desmontar un sistema de archivos montado previamente.
- *mtools*, es una colección de herramientas de dominio público que permite a los sistemas Unix manipular archivos de MS-DOS: leer, escribir y moverse por archivos en un sistema de archivos de MS-DOS (generalmente un disquete).



MANIPULACIÓN DE ARCHIVOS



INTRODUCCIÓN

Linux dispone de un numeroso conjunto de utilidades para manipular archivos (individualmente o en grupos) tecleando comandos en una terminal. A través de esta modalidad de trabajo es posible alcanzar un alto nivel de flexibilidad y versatilidad para realizar operaciones de cualquier grado de complejidad.

- Un nombre de archivo puede tener entre **1 y 255** caracteres.
- Se puede utilizar cualquier carácter excepto la barra inclinada **/** y no es recomendable emplear los caracteres con significado especial en Linux, que son los siguientes: **= \ ~ ' " ` * ; - ? [] () ! & ~ < >**. Para emplear ficheros con estos caracteres o espacios hay que introducir el nombre del fichero entre comillas.
- Se pueden utilizar números exclusivamente si así se desea. Las letras mayúsculas y minúsculas se consideran diferentes, y por lo tanto no es lo mismo carta.txt que Carta.txt o carta.Txt.

TIPOS DE ARCHIVOS

A diferencia de Windows que identifica sus tipos de archivo por las extensiones que estos tengan como pueden ser: jpeg, txt, png, etc.., Linux sólo distingue tres tipos de archivos:

- **Archivos o ficheros ordinarios**, son los mencionados anteriormente.
- **Directorios (o carpetas)**, es un archivo especial que agrupa otros ficheros de una forma estructurada.
- **Archivos especiales**, son la base sobre la que se asienta Linux, puesto que representan los dispositivos conectados a un ordenador, como puede ser una impresora. De esta forma introducir información en ese archivo equivale a enviar información a la impresora. Para el usuario estos dispositivos tienen el mismo aspecto y uso que los archivos ordinarios.

RUTAS

Para acceder a un fichero o carpeta utilizamos lo que se llama una ruta, una ruta es algo así como la dirección de una casa.

Hay dos tipos de rutas:

- Relativas
- Absolutas

La diferencia entre una ruta relativa y una absoluta radica en que la relativa depende del directorio en el que te encuentras actualmente, mientras que la absoluta no.

EXPLORACIÓN DEL ÁRBOL

- El comando **pwd** (de las siglas en inglés *print working directory*, cuya traducción sería *imprimir directorio de trabajo*) se utiliza para imprimir el nombre del directorio actual en una sesión de comandos bajo un sistema operativo Unix o derivado.
- El comando **cd** (de las siglas en inglés *change directory*) permite cambiar el directorio de trabajo en curso al directorio que se le especifique. **cd** sin parámetros cambia al directorio de trabajo predeterminado del usuario (establecido en la variable **HOME**). **cd..** cambia al directorio un nivel más bajo que el actual (si estaba en */home/pepe/* ahora estaré en */home*). **cd /home/pepe/Escritorio** cambia al directorio de trabajo ***/home/pepe/Escritorio***
- El comando **ls** (del inglés *list*, cuya traducción es *lista*, *listado* o *listar*) es un comando del sistema operativo Unix y derivados que muestra un listado con los archivos y directorios de un determinado directorio. Los resultados se muestran ordenados alfabéticamente.

EXPLORACIÓN DEL ÁRBOL

El comando **file** es una utilidad que realiza una serie de pruebas (test) para determinar el tipo y formato de un archivo. Se detallan los tipos de pruebas pruebas,

- **Sistema de archivos:** se intenta determinar si el archivo a examinar es un archivo del sistema por medio de la función (system call) *stat*. Gracias a esta prueba se puede determinar si es un dispositivo, enlace simbólico, una tubería, etc.
- **Número mágicos:** Se intenta determinar el tipo, analizando determinados bytes ubicados en específicas posiciones dentro del archivo. Estos bytes se les denomina números mágicos, y suelen estar al comienzo de la cabecera. La información para realizar dicho análisis figura en el archivo */usr/share/misc/magic.mgc*.
- **Prueba de sintaxis:** esta última prueba consiste en determinar que tipo de sintaxis posee un archivo de texto. Esta prueba solo se realiza sobre los archivos que se haya determinado que sean texto plano. Básicamente consiste en buscar la presencia de determinadas palabras claves que permiten identificar la sintaxis usada dentro de dicho archivo.

EXPLORACIÓN DEL ÁRBOL

El comando **stat** muestra información sobre ficheros o sobre los sistemas de ficheros donde se encuentran. De un fichero podemos obtener la siguiente información:

- Tamaño en bytes.
- Número de bloques de datos reservados (bloques de 512 bytes).
- Tamaño en bytes de los bloques de E/S (mínima información con la que se trabaja en el disco).
- Tipo de fichero.
- Dispositivo en el que reside el fichero (Numeración de los dispositivos en el kernel de Linux).
- Nodo-i.
- Número de enlaces
- Permisos
- Dueño
- Grupo

ARCHIVOS

Comando **touch** crea un fichero vacío desde el terminal de linux, si el fichero existe le cambia la fecha y hora de modificación con la actual del sistema o podemos especificar la hora con el parámetro **-date**. La sintaxis del comando **touch** es la siguiente :

touch archivo.txt

Si el archivo no existe lo crea vacío y si existe se modifica la fecha con actual del sistema. También podemos cambiar la fecha del fichero por una fecha que nosotros queramos

touch --date=2018-09-01 archivo.txt

El comando de arriba modificará el archivo archivo.txt con la fecha 01 de Septiembre de 2018.

ARCHIVOS

- El comando ***cp*** es un comando de la familia de los Sistemas Operativos Unix que permite *copiar archivos y directorios*. El comando posee tres modos de operación, indicados según su invocación, lo cual posibilita copiar un archivo en otro, uno o más archivos a un directorio, o bien un directorio en otro.
- El comando ***rm*** es un comando de la familia de sistemas operativos Unix usada para eliminar archivos y directorios del sistema de archivos. *Esta orden debe utilizarse con cautela, ya que puede ser muy destructiva*, debido a que, al momento de ser llamada, por omisión borra los archivos sin pedir confirmación.
- El comando ***mv*** es usado para mover o renombrar archivos o directorios del sistema de archivos. El archivo original es borrado y se crea un nuevo archivo con el mismo contenido, el nombre puede ser diferente o puede ser el mismo. En caso de que se use en la misma ruta con el nombre del archivo y otro nombre, solo cambiara el nombre del archivo.

ORGANIZACIÓN FISICA DE LOS ARCHIVOS EN LINUX

Un ***inodo*** es una estructura de datos que almacena información sobre un fichero de nuestro sistema de archivos. Un ***inodo*** no tiene nombre y se identifica mediante un número entero único. Cada ***inodo*** únicamente puede contener datos de un solo fichero del sistema de archivos. Por lo tanto, si tenemos 4 archivos y 4 directorios estaremos usando 8 ***inodos***.

Algunos de los sistemas de archivos que trabajan con ***inodos*** son:

- ext2/3/4.
- UFS.
- ReiserFS.
- FFS.
- XFS.
- Btrfs.

ORGANIZACIÓN FÍSICA DE LOS ARCHIVOS EN LINUX

Los metadatos almacenados en un inodo son los siguientes:

- **Número de inodo.** El número de inodo es un número entero único que sirve para identificar un inodo.
- **Tamaño del fichero** así como el número de bloques que ocupa el fichero en el disco duro.
- **El dispositivo de almacenamiento** en que está almacenado el fichero. (Device ID)
- **Número de enlaces.** Por lo tanto si hay 2 archivos que apuntan a un mismo inodo tendremos 2 enlaces. Si tenemos un directorio que contiene 15 archivos tendremos 15 enlaces.
- El identificador de usuario (**UID** o User ID). Por lo tanto, los inodos especifican el propietario de un fichero.
- El identificador de grupo (**GID** o Group ID). De este modo, un inodo contiene el grupo a que pertenece un fichero.
- **Marcas de tiempo** como por ejemplo la fecha en que se ha creado el archivo, la fecha del último acceso, etc.
- **Tabla de direccionamiento** donde se detallan los bloques del disco duro en que está almacenado el fichero.

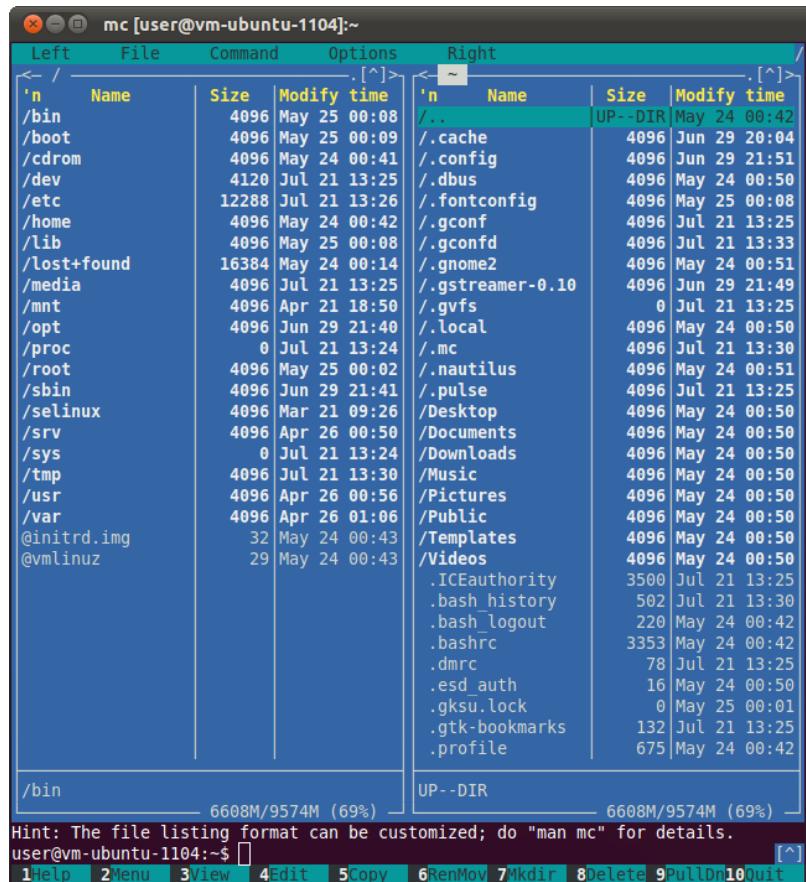


ADMINISTRADORES DE ARCHIVOS

Un **administrador de archivos**, **gestor de archivos** o **explorador de archivos** (del inglés **file manager**) es un programa informático que proporciona una interfaz de usuario para administrar archivos y directorios. Las operaciones más comunes realizadas en archivos o grupos de archivos incluyen crear, abrir (p. ej., ver, reproducir, editar o imprimir), cambiar nombre, copiar, mover, eliminar y buscar archivos; así como modificar sus atributos, propiedades y permisos de acceso. Las carpetas y los ficheros pueden visualizarse en un árbol jerárquico basado en su estructura de directorios.

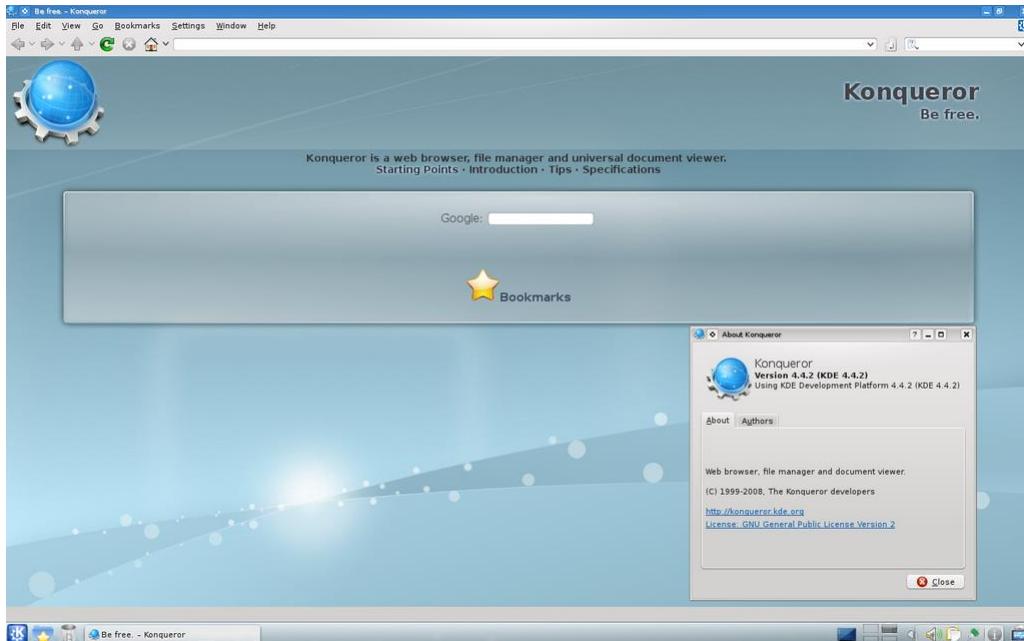
ADMINISTRADORES DE ARCHIVOS

- *Midnight Commander* es una aplicación que funciona en modo texto. La pantalla principal consiste en dos paneles en los cuales se muestra el sistema de ficheros. Se usa de un modo similar a otras aplicaciones que corren en el shell o interfaz de comandos de Unix. Las teclas de cursor permiten desplazarse a través de los ficheros, la tecla insertar se usa para seleccionar ficheros y las Teclas de función realizan tareas tales como borrar, renombrar, editar, copiar ficheros, etc. Las versiones más recientes de *Midnight Commander* incluyen soporte para el ratón para facilitar el manejo de la aplicación.



ADMINISTRADORES DE ARCHIVOS

- **Konqueror** es un navegador web, administrador de archivos y visor de archivos. Forma parte oficial del proyecto KDE. Es software libre y de código abierto, y al igual que el resto de los componentes de KDE, está liberado bajo la licencia GPL. El nombre *Konqueror* es un juego de palabras con el nombre de otros navegadores: primero vino el Navigator (navegador), después el Explorer(explorador), y finalmente el *Konqueror* (conquistador). Además, sigue la convención de KDE de que los nombres de los programas contengan la letra K.

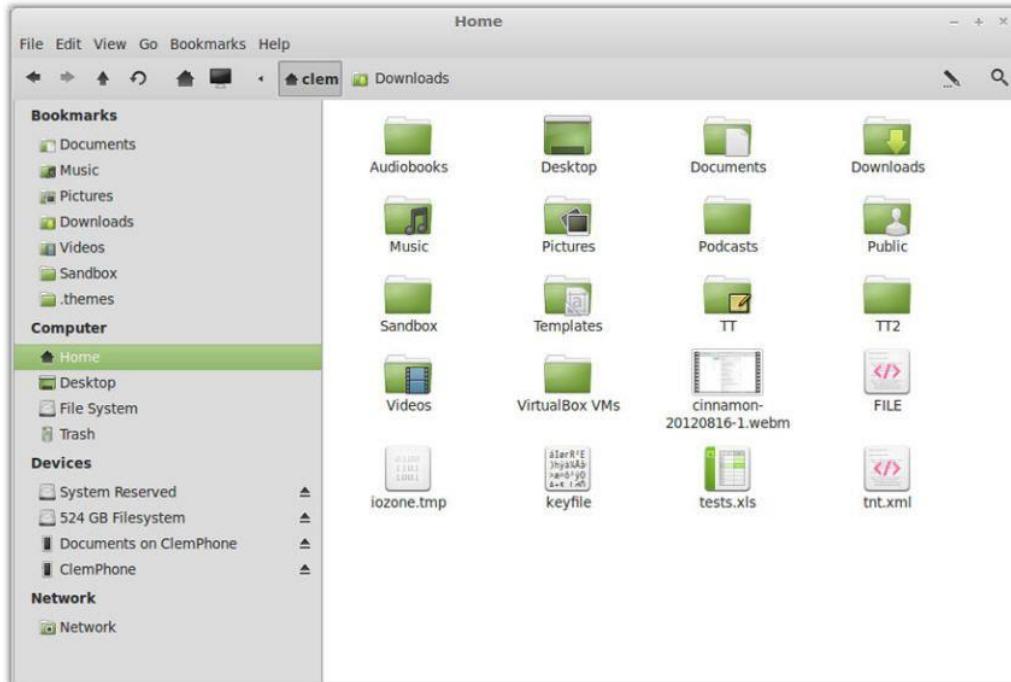


ADMINISTRADORES DE ARCHIVOS



Nautilus es el gestor de archivos por defecto en Gnome. Se encarga de las funciones relacionadas con el manejo de ficheros, directorios y dispositivos, pero ***Nautilus*** es mucho más, entre sus funciones adicionales encontramos:

- Grabador de CD/DVD
- Gestión de tipografías
- Cliente FTP
- Manejo del escritorio
- Gestión de dispositivos removibles (USB, CD, DVD...) y unidades de red (samba, nfs, ssh, ftp...)
- Previsualización de archivos multimedia
- Scripts y extensiones programables
- Transferencia por bluetooth



CONSULTAR ARCHIVOS

El comando de linux **cat** concatena archivos y los muestra en la salida estándar. La sintaxis es la siguiente:

cat [OPCIONES] [ARCHIVO]...

OPCIÓN	DESCRIPCIÓN
-A	Mostrar todo
-b	Omitir los números de la línea para los espacios en blanco en la salida.
-e	Un carácter \$ se mostrará al final de cada línea anterior a una nueva línea.
-E	Muestra un \$ (símbolo de dólar) al final de cada línea.
-n	Numera todas las líneas en la salida.
-s	Si la salida tiene múltiples líneas vacías las sustituye con una única línea vacía.

CONSULTAR ARCHIVOS

more es un comando para ver (pero no modificar) el contenido de un archivo o comando y visualizarlo por páginas. Está disponible en varios sistemas operativos como Unix y derivados DOS, OS/2, y Windows. Este comando sólo permite la navegación hacia adelante y al pulsar una tecla, avanza hacia la siguiente página. La sintaxis es la siguiente:

more [OPCIONES] [ARCHIVO]...

OPCIÓN	DESCRIPCIÓN
-num	Entero que especifica el numero de líneas de un archivo
-d	Uso con teclas específicas “espacios continuar”, ‘q’ quitar ‘h’ para instrucciones
-l	Esta opción evita que al encontrar un carácter finalice el comando
-f	Cuenta las líneas de manera lógica, en vez de por líneas.
-p	No se desplaza, sino que limpia la pantalla y muestra el texto.
-s	Compacta líneas en blanco en una
-u	Suprime el subrayado.

CONSULTAR ARCHIVOS

El comando ***less*** es un visualizador de archivos de texto que funciona en interpretes de comando. A diferencia de otros programas similares, ***less*** permite una completa navegación por el contenido del archivo, utilizando un mínimo de recursos del sistema. La sintaxis es la siguiente:

less [OPCIONES] [ARCHIVO]...

OPCIÓN	DESCRIPCIÓN
-g	Solo se resalta la coincidencia actual de cualquier texto buscado
-I	Las búsquedas serán insensibles a mayúsculas o minúsculas.
-M	Muestra datos de navegación.
-N	Mostrar numero de cada línea (útil en archivos de código fuente)
-p	Muestra los resultados a partir de la primera aparición de la palabra o patrón escrito.
-S	Deshabilita auto-ajuste de líneas (las líneas largas serán visibles por navegación lateral)

CONSULTAR ARCHIVOS

El comando ***od*** (*Octal Dump*) nos permite ver byte a byte el contenido de un archivo. Podemos visualizarlo en varios formatos, como por ejemplo:

- Números Octales
- Números Hexadecimales
- Caracteres Visibles y No Visibles

od [OPCIONES] [ARCHIVO]...

OPCIÓN	DESCRIPCIÓN
-b	Visualiza el archivo en octal
-c	Visualiza el archivo en ASCII
-x	Visualiza el archivo en caracteres hexadecimal

CONSULTAR ARCHIVOS

El comando de linux **cat** concatena archivos y los muestra en la salida estándar. La sintaxis es la siguiente:

cat [OPCIONES] [ARCHIVO]...

OPCIÓN	DESCRIPCIÓN
-A	Mostrar todo
-b	Omitir los números de la línea para los espacios en blanco en la salida.
-e	Un carácter \$ se mostrará al final de cada línea anterior a una nueva línea.
-E	Muestra un \$ (símbolo de dólar) al final de cada línea.
-n	Numera todas las líneas en la salida.
-s	Si la salida tiene múltiples líneas vacías las sustituye con una única línea vacía.



EDICIÓN DE ARCHIVOS DE TEXTO CON “VI”

EDITOR “VI”

El editor ***vi*** es un editor de texto que maneja en memoria el texto entero de un archivo. Es el editor clásico de UNIX (se encuentra en todas las versiones). Puede usarse en cualquier tipo de terminal con un mínimo de teclas, lo cual lo hace difícil de usar al enfrentarse por primera vez al mismo.

Existen tres modos o estados de ***vi***:

- ***Modo comando***: este es el modo en el que se encuentra el editor cada vez que se inicia. Las teclas ejecutan acciones (comandos) que permiten mover el cursor, ejecutar comandos de edición de texto, salir de vi, guardar cambios, etc.
- ***Modo inserción o texto***: este es el modo que se usa para insertar el texto. Existen varios comandos que se pueden utilizar para ingresar a este modo.
- ***Modo línea o ex***: se escriben comandos en la última línea al final de la pantalla.

EDITOR “VI”

Movimiento del cursor:

Comando (teclas)	Acción
Flechas	Mover en la dirección de la flecha
h	Mover hacia la izquierda
l	Mover hacia la derecha
k	Mover hacia arriba
j	Mover hacia abajo
1G	Lleva el cursor hasta el comienzo del archivo
G	Lleva el cursor hasta el final del archivo

Cambio de modo comando a texto:

Comando	Acción
i	Inserta texto a la izquierda del cursor
a	Inserta texto a la derecha del cursor
A	Inserta texto al final de la línea donde se encuentra el cursor
I	Inserta texto al comienzo de la línea donde se encuentra el cursor
o	Abre una línea debajo de la actual
O	Abre una línea encima de la actual

EDITOR “VI”

Borrar texto:

Comando	Acción
x	Borra el carácter bajo el cursor
dd	Borra la línea donde se encuentra el cursor
ndd	Borra las próximas n líneas
D	Borra desde donde se encuentra el cursor hasta el final de la línea
dw	Borra desde donde se encuentra el cursor hasta el final de una palabra

Buscar texto:

Comando	Acción
/texto	Busca hacia adelante la cadena de caracteres “texto”
?texto	Busca hacia atrás la cadena de caracteres “texto”

Salir de vi, salvar, no salvar cambios, etc.:

Comando	Acción
:q	Salir si no hubo cambios
:q!	Salir sin guardar cambios
:w	Guardar cambios
:w archivol	Guardar cambios en archivol
:wq	Guardar cambios y salir

OTRAS HERRAMIENTAS DE EDICION



Algunos editores hexadecimales:

- *Ghex*
- *Khexedit*
- *Okteta*
- *Wxhexeditor*
- *Ifhex*



PERMISOS DE ACCESO A LOS ARCHIVOS



CUENTA DE USUARIO

Una cuenta de usuario se compone de **nombre de usuario (*login*) y de contraseña (*password*)**. Las cuentas de usuario son creadas por el administrador que en Linux es un usuario especial llamado **root**. Los usuarios deberán pertenecer al menos a un grupo de usuarios ya que obligatoriamente deben tener asignado un grupo principal o grupo primario.

GRUPO DE USUARIO

Para poder administrar los permisos de los usuarios de una forma más flexible, el sistema Linux permite la organización de usuarios en **grupos** y establecer permisos a los grupos. Ejemplo, si en un centro educativo el grupo "**profesores**" tiene acceso a ciertas carpetas, cuando demos de alta un profesor nuevo, tan solo tendremos que añadirle al grupo "**profesores**" para que pueda acceder a todas esas carpetas. Es lo que se denomina administración de permisos por grupos.

JERARQUIA DE USUARIO

Identificamos 3 tipos de usuarios en el sistema Linux, los cuales son:

- **Usuario root**, También llamado superusuario o administrador, su ID user es **0**, esta es la única cuenta de usuario con privilegios sobre todo el sistema, *tiene acceso total a todos los archivos y directorios con independencia de propietarios y permisos. Este usuario controla la administración de cuentas de usuarios. Puede ejecutar tareas de mantenimiento del sistema.*
- **Usuarios especiales**, bin, daemon, adm, lp, sync, shutdown, mail, operator, squid, apache, se les llama también cuentas del sistema, estos usuarios no tienen todos los privilegios del usuario **root**, pero dependiendo de la cuenta asumen distintos privilegios de **root**. No tienen contraseñas pues son cuentas que no están diseñadas para iniciar sesiones con ellas

JERARQUIA DE USUARIO

- **Usuarios normales**, Se usan para usuarios individuales, cada usuario dispone de un directorio de trabajo, ubicado generalmente en **/home**. Cada usuario puede personalizar su entorno de trabajo, tiene solo privilegios completos en su directorio de trabajo “**HOME**”. Por seguridad, es siempre mejor trabajar como un usuario normal en vez del usuario root, y cuando se requiera hacer uso de comandos solo de **root**, utilizar el comando **su**

COMANDOS ÚTILES



- ➔ **ls**Lista de carpetas
- ➔ **ls -al**Lista de carpetas con archivos ocultos
- ➔ **cd nombre-de-carpetalr** a una carpeta concreta
- ➔ **cd**Volver a la carpeta Home
- ➔ **cd -**Volver a la carpeta anterior
- ➔ **pwd**Mostrar la carpeta donde te encuentras
- ➔ **mkdir nombre-de-carpetacrear** una carpeta nueva
- ➔ **rm -r nombre-de-carpetaborrar** una carpeta
- ➔ **rm -rf nombre-de-carpetafuerza** el borrado de una carpeta
- ➔ **rm nombre-de-archivo**Borrar un archivo
- ➔ **rm -f nombre-de-archivo**Fuerza el borrado de un archivo
- ➔ **cp nombre-de-archivo nombre-de-copia**Copia un archivo
- ➔ **find nombre-de-archivo**Busca un archivo
- ➔ **wget enlace**Descarga un archivo

PERMISOS DE UNIX

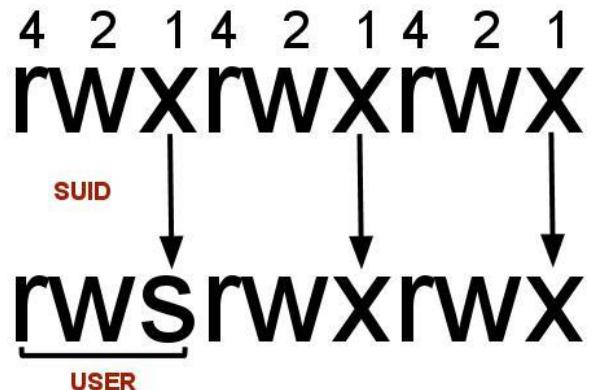
Los permisos de acceso de cada archivo y directorio del sistema son mostrados por un conjunto de 10 caracteres, los cuales proporcionan información acerca del tipo de elemento, junto con permisos para el usuario y grupo propietario para leer, escribir y ejecutar,

Ejemplos:

Permisos	Descripción
<code>drwxr-xr-x</code>	Directorio con permiso 755
<code>crw-rw-r--</code>	Archivo de carácter especial con permiso 664.
<code>srwxrwxr-x</code>	Zócalo con permiso 775.
<code>prw-rw-r--</code>	Tubería (FIFO) con permiso 664.
<code>-rw-r--r--</code>	Archivo regular con permiso 644.

SUID

Setuid, también llamado a veces "suid", y "***setgid***" son permisos de acceso que pueden asignarse a archivos o directorios en un sistema operativo basado en Unix. Se utilizan principalmente para permitir a los usuarios del sistema ejecutar binarios con privilegios elevados temporalmente para realizar una tarea específica





STICKY BIT

El **Sticky bit** se utiliza para permitir que cualquiera pueda escribir y modificar sobre un archivo o directorio, pero que solo su propietario o root pueda eliminarlo. Un ejemplo de uso es el directorio /tmp, que debe tener permisos para ser utilizado por cualquier proceso, pero solo el dueño o root puede eliminar los archivos que crea.

SGID

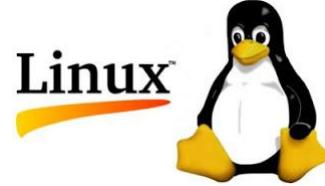
El **SGID** es lo mismo que en el **SUID**, pero a nivel de grupo. Es decir, todo archivo que tenga activo el SGID, al ser ejecutado, tendrá los privilegios del grupo al que pertenece. Esto es muy usado cuando queremos configurar un directorio colaborativo: si aplicamos este bit al directorio, cualquier archivo creado en dicho directorio, tendrá asignado el grupo al que pertenece el directorio.

ADMINISTRAR PERMISOS

- El **comando chgrp** permite cambiar el grupo de usuarios de un archivo o directorio en sistemas tipo UNIX. Cada archivo de Unix tiene un identificador de usuario (UID) y un identificador de grupo (GID), que se corresponden con el usuario y el grupo de quien lo creó.
- **chmod** ("change mode", cambiar modo en idioma inglés) es una llamada al sistema y su **comando** asociado en el sistema operativo UNIX (estandarizados en POSIX y otros estándares) que permite cambiar los permisos de acceso de un fichero o directorio.
- El comando **umask**, es la abreviatura de *user file-creation mode mask*, y sirve para establecer los permisos por defecto que tendrán los nuevos ficheros y directorios que creamos.



ADMINISTRACIÓN DE PROCESOS

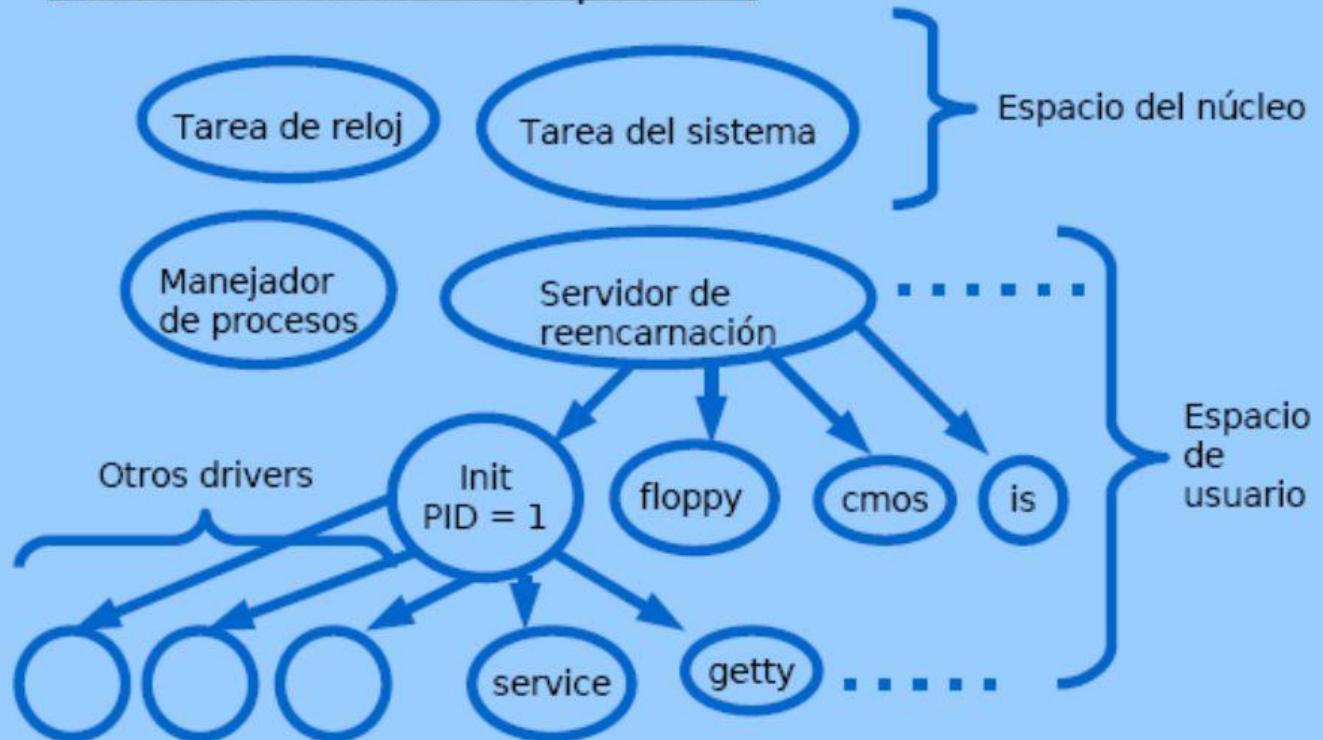


A los procesos frecuentemente se les refiere como tareas. El contexto de un programa que esta en ejecución es lo que se llama un proceso. Este contexto puede ser mas procesos hijos que se hayan generado del principal (proceso padre), los recursos del sistema que este consumiendo, sus atributos de seguridad (tales como su propietario y permisos de archivos asi como roles y demás de SELinux)



Arbol de Procesos

Inicialización del árbol de procesos



VISUALIZACION DE PROCESOS

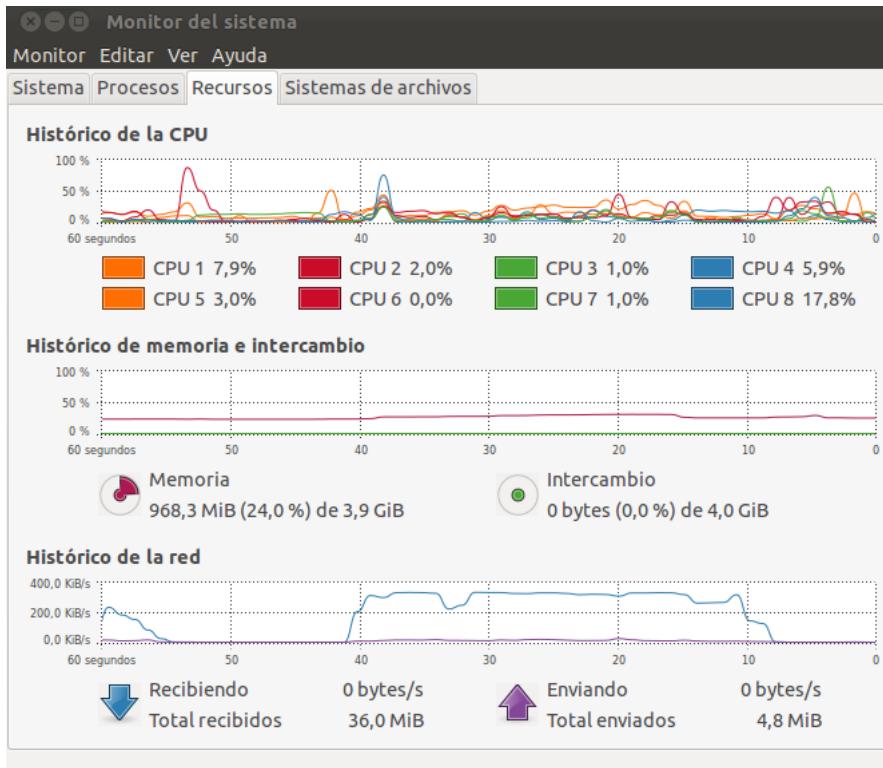
- ***ps*** ("process status") es un **comando** asociado en el sistema operativo UNIX (estandarizado en POSIX y otros) que permite visualizar el estado de un Proceso.
- El ***comando top*** nos da información acerca del uso de la cpu, de la memoria, de los procesos en ejecución en tiempo real.
- ***pstree***, es un pequeño programa que nos permitirá ver el árbol de procesos activos, en nuestros sistemas GNU Linux. Hay que recordar que recientemente hablamos de otro programa similar, aunque en ese caso se trataba de 'tree', una herramienta que nos permitiría ver el árbol de directorios.

SEÑALES Y COMANDO KILL

Kill es un comando unix/linux que permite enviar señales a uno o varios procesos del sistema a través de la shell (bash, ksh, etc). Las señales más utilizadas suele ser la de matar un proceso (9 ó SIGKILL), pararlo (TERM) o reiniciarlo (1 ó HUP) pero hay muchas más que pueden ser útiles en ocasiones. El listado completo de señales disponibles puede visualizarse ejecutando **kill -l**:

```
$ 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
```

ADMINISTRADORES DE PROCESOS GRÁFICOS



Procesos | Recursos | Sistemas de archivos

Nombre del proceso Usuario % CPU ID Memoria Lectura total Escritura total Lectura de disco Escritura en disco

at-spi2-registryd	javier	0	2364	276,0 KiB	N/D	N/D	N/D	N/D
at-spi-bus-launcher	javier	0	2356	364,0 KiB	N/D	N/D	N/D	N/D
dbus-daemon	javier	0	2242	1,4 MiB	N/D	N/D	N/D	N/D
dbus-daemon	javier	0	2261	276,0 KiB	N/D	N/D	N/D	N/D
dconf-service				Alt+Intro, 0 KiB	N/D	N/D	N/D	N/D
deja-dup-monitor				Mapas de memoria, Ctrl+M, 6 MiB	N/D	N/D	N/D	N/D
evolution-addressbook-facto				Archivos abiertos, Ctrl+O, N/D	N/D	N/D	N/D	N/D
evolution-calendar-factory				Cambiar prioridad, ▶, 1 MiB	N/D	N/D	N/D	N/D
evolution-calendar-factory-su				Detener, Ctrl+S, 0 KiB	N/D	N/D	N/D	N/D
evolution-source-registry				Continuar, Ctrl+C, 3 MiB	N/D	N/D	N/D	N/D
gdm-x-session				Finalizar, Ctrl+E, 8 MiB	N/D	N/D	N/D	N/D
gnome-keyring-daemon	javier	0	2239	276,0 KiB	N/D	N/D	N/D	N/D
gnome-keyring-daemon	javier	0	2378	N/D	N/D	N/D	N/D	N/D
gnome-session-binary	javier	0	2246	1,3 MiB	N/D	N/D	N/D	N/D
gnome-shell	javier	0	2393	552,4 MiB	N/D	N/D	N/D	N/D
gnome-shell-calendar-server	javier	0	2432	812,0 KiB	N/D	N/D	N/D	N/D
gnome-software	javier	0	2599	151,3 MiB	N/D	N/D	N/D	N/D
gnome-system-monitor	javier	1	9098	15,1 MiB	31,9 MiB	20,0 KiB	N/D	N/D
goa-daemon	javier	0	2449	320,0 KiB	N/D	N/D	N/D	N/D
goa-identity-service	javier	0	2461	240,0 KiB	N/D	N/D	N/D	N/D
gsd-a11y-settings	javier	0	2524	256,0 KiB	N/D	N/D	N/D	N/D
gsd-clipboard	javier	0	2530	3,5 MiB	N/D	N/D	N/D	N/D

Finalizar proceso



TRABAJO CON PROCESOS

Linux tiene una multitud de **servicios/servidores** y estos pueden arrancar de tres maneras diferentes. Una es junto al inicio del sistema, otra al entrar en un runlevel (nivel de ejecución) determinado, o también cuando los usuarios los inicien manualmente.

Gran parte de los servicios en Linux se encuentran en **/etc/init.d**, donde están los scripts que los inician/detienen. Por lo tanto, los comandos que se ejecutarán para realizar ambas acciones estarán basados en esa ubicación.

/etc/init.d/start (para iniciar un servicio)

/etc/init.d/ stop (para detenerlo)

ESTADOS DE LOS PROCESOS

- **S** ("sleeping") el proceso esta en espera
- **R** ("running") el proceso esta en ejecución
- **T** ("stop") el proceso se encuentra parado
- **D** proceso que se encuentra bloqueado a la espera de un recurso
- **Z**, ("zombie") es un proceso que se encuentra en estado zombie, es decir, es un proceso que ha finalizado pero su proceso padre sigue en ejecución y no se "ha dado cuenta" de la circunstancia de su proceso hijo



SHELL BASH



INTRODUCCIÓN

Bash (*Bourne-again shell*) es un programa informático, cuya función consiste en interpretar órdenes, y un lenguaje de consola. Es una shell de Unix compatible con POSIX y el intérprete de comandos por defecto en la mayoría de las distribuciones GNU/Linux.

La sintaxis de órdenes de Bash es un superconjunto de instrucciones basadas en la sintaxis del intérprete Bourne. La especificación definitiva de la sintaxis de órdenes de Bash

Definiendo Una Variable

```
#!/bin/bash
#
# Esto es un ejemplo en Bash del clasico "Hola Mundo"
#
MENSAJE="Hola Mundo"
echo $MENSAJE
```

Usando Variables De Entorno

```
#!/bin/bash
echo "El usuario '$USERNAME' ha ejecutado el script $0, en el ordenador '$HOSTNAME'. "
```

Usando Caracteres Especiales En Variables

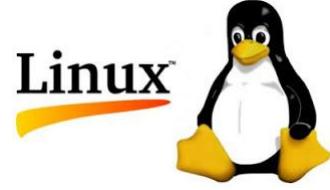
Existen una serie de caracteres que tienen un significado especial en Bash, por ejemplo \$ y ". Si queremos usar literalmente estos caracteres en el valor de una variable tendremos que usar el simbolo '\' delante de ellos.

```
#!/bin/bash

MENSAJE="\\"Hola Mundo ...\\"
echo "El valor de la variable \$MENSAJE es $MENSAJE"
```

Escapes con contrabarra

Backslash Escape	Se expande a...
\b	Un carácter de retroceso
\e	Un carácter de escape
\f	Un carácter de alimentación de línea (<i>form feed</i>)
\n	Un carácter de nueva línea
\r	Un carácter de retorno de carro
\t	Un tabulador horizontal
\v	Un tabulador vertical
\\\	Un carácter contrabarra
\'	Un carácter de comilla simple
\nnn	El carácter de 8 bits cuyo valor es el número octal nnn (de uno a tres dígitos)
\xHH	El carácter de 8 bits cuyo valor es el número hexadecimal HH (uno o dos dígitos hexadecimales)
\cx	Un carácter control-X



REDIRECCIONES

Una redirección consiste en **trasladar la información de un tipo a otro**, por ejemplo de la salida estándar a la entrada estándar o del error estándar a la salida estándar. Esto lo logramos usando el símbolo `>`.

Por ejemplo, para redireccionar la salida de un comando y volcarla a un archivo bastaría con ejecutar:

```
$ ls -la ~ > archivo.txt
```

REDIRECCIONES

- **descriptor de archivo** o **descriptor de fichero** son usados generalmente en sistemas operativos POSIX. En POSIX, un descriptor de archivo es un entero, específicamente del tipo **int** de **C**. Hay 3 tipos de descriptores de archivo estándar que presumiblemente tiene cada proceso:

Valor entero	Nombre
0	Entrada estándar (stdin)
1	Salida estándar (stdout)
2	Error estándar (stderr)

En Linux, el teclado es una **canal de entrada** y la pantalla por dónde se visualiza la información es un **canal de salida**. Así que estos **canales de salida** se pueden **redireccionar** hacia ficheros. Los errores que se puedan producir en la ejecución de comandos son un **canal especial**.

TUBERIAS

Una tubería en Linux no es más que una forma práctica de **redireccionar la salida estándar de un programa** hacia la entrada estándar de otro. Esto se logra usando el símbolo / (**pipe**).

Por ejemplo, podríamos ver los procesos que están corriendo en el sistema usando **ps** y le redireccionamos la salida a **sort** para que los ordene por PID:

```
$ ps -a | sort
```

También podemos redireccionar la salida estándar del comando **cat** y pasarla como entrada estándar del comando **wc** para contar las líneas y palabras de un archivo:

```
$ cat archivo.txt | wc
```



ALIAS



Existe una forma muy sencilla de personalizar y hacer más fáciles algunos comandos que normalmente usamos en “**la consola**”, mediante el uso de **Alias**.

Un **Alias** como su nombre lo indica, nos servirá para reemplazar una palabra o serie de palabras con otra más corta y sencilla. Pongamos un ejemplo práctico, digamos que deseamos ver los **logs** del sistema, usando una aplicación llamada **Colorizer** la cual se encarga de colorear el resultado en consola.

```
$ sudo tailf -n 5 /var/log/syslog | ccze
```

Creando un Alias.

Crear un alias es realmente sencillo. La sintaxis sería:

```
alias palabra_corta='comando o palabras a reemplazar'
```

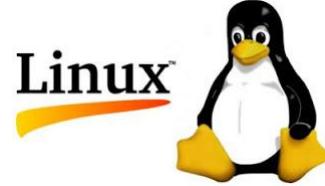
si tomamos el ejemplo anterior sería:

```
alias syslog='sudo tailf -n 5 /var/log/syslog | ccze'
```

EJECUCION DE COMANDOS

COMANDO INTERNO ESPECIAL	DESCRIPCIÓN
:	Comando nulo. Se suele utilizar en estructuras de control que requieren un comando para ser sintácticamente correctas.
<i>Eval</i>	Permite crear comandos a partir de sus argumentos
<i>Exec</i>	Ejecuta comandos (sustituyendo al Shell actual) y abre, cierra y copia descriptores de fichero
<i>Exit</i>	Provoca que el Shell termine
<i> Readonly</i>	Permite ser que una variable sea de sólo lectura (no admite asignaciones)
<i>Set</i>	Establece opciones del proceso Shell actual y modifica los parámetros posicionales.
<i>Shift</i>	Elimina el numero indicado de parámetros posicionales empezando desde el 1, desplazando el resto de parámetros a posiciones inferiores

EJECUCION DE COMANDOS



COMANDO INTERNO ESPECIAL	DESCRIPCIÓN
Times	Muestra los tiempos de procesamiento del Shell y sus procesos hijos.
Trap	Permite atrapar o ignorar señales del sistema.

- **Type** – Muestra información sobre el tipo de comando.
- **Which** – Localiza un comando.
- **Help** – Muestra la página de referencia para la orden interna del shell.
- **Man** – Muestra una guía de referencia del comando en pantalla.
- **Whereis**, Este comando se utiliza para localizar el archivo binario, el código fuente y la página de manual de un determinado comando. Su sintaxis es como sigue:

whereis [opciones] archivo...

SUSTITUCION DE COMANDOS

La *sustitución de comandos* consiste en ejecutar uno o varios comandos y escribir su salida de forma que sea tomada como argumento por otro comando. Esta sustitución es diferente del encadenamiento de comandos que se hace con el símbolo | (pipe), pues no arroja un flujo que es leído a través de standard input sino que escribe la información textualmente en la línea de comandos. La sustitución de comandos en bash hace uso del operador **\$()** (en otros shells normalmente se usan las comillas invertidas o backticks ``).