

# **Gestión de la Información en Sistemas Operativos**

**8 noviembre 2019**

# L1 Sistema Operativo

- ❑ Software básico del ordenador.  
Gestiona los recursos hardware del ordenador.



El **usuario final** trabajará felizmente con las aplicaciones.

El **administrador de sistemas** trabajará con el sistema operativo para conseguir que realice su función.



# Sistema Operativo

- ❑ Es un **conjunto de programas, funciones y servicios**, que permite controlar y gestionar los elementos físicos conectados al ordenador, así como obtener el máximo rendimiento de dichos elementos.
- ❑ Ofrece abstracción al **software de aplicación** y usuarios y los aísla del hardware.

# ❑ Núcleo de los Sistemas Operativos

Gestión de los  
procesos

Gestión de la  
memoria

Gestión de  
Entrada y  
Salida

Gestión del  
sistema de  
archivos



# Sistema Operativo Ubuntu / Linux

- ❑ **UNIX** es un sistema operativo propietario AT&T para grandes ordenadores y servidores.
- ❑ **Linux**: Kernel basado en el sistema operativo propietario **Unix** para computadoras Intel. Es el corazón de los actuales SO *open source*.
- ❑ Distribuciones de SO basados en Linux:  
**GNU/Linux**
- ❑ **Ubuntu**: Distribución completa de Linux (Kernel, manejadores, aplicaciones) como SO basado en **Debian**.
- ❑ Linux es compatible con UNIX (como macOS)

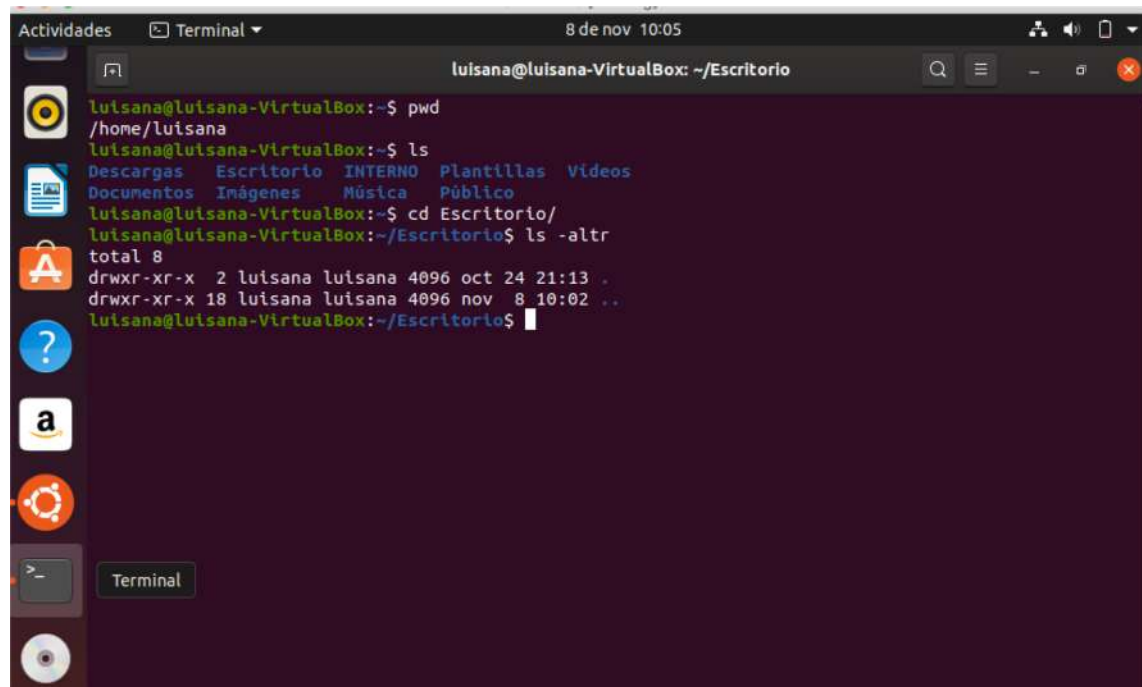
# Interfaz Gráfica de GNU/Linux

- ✓ GUI (*Graphical User Interface*) o DE (*Desktop Environment*)
- ✓ **GNOME** (GNU Network Object Model Environment): entorno de escritorio predeterminado de Ubuntu.
- ✓ Es software libre de código abierto (FOSS).



# Interfaz de Texto de GNU/Linux

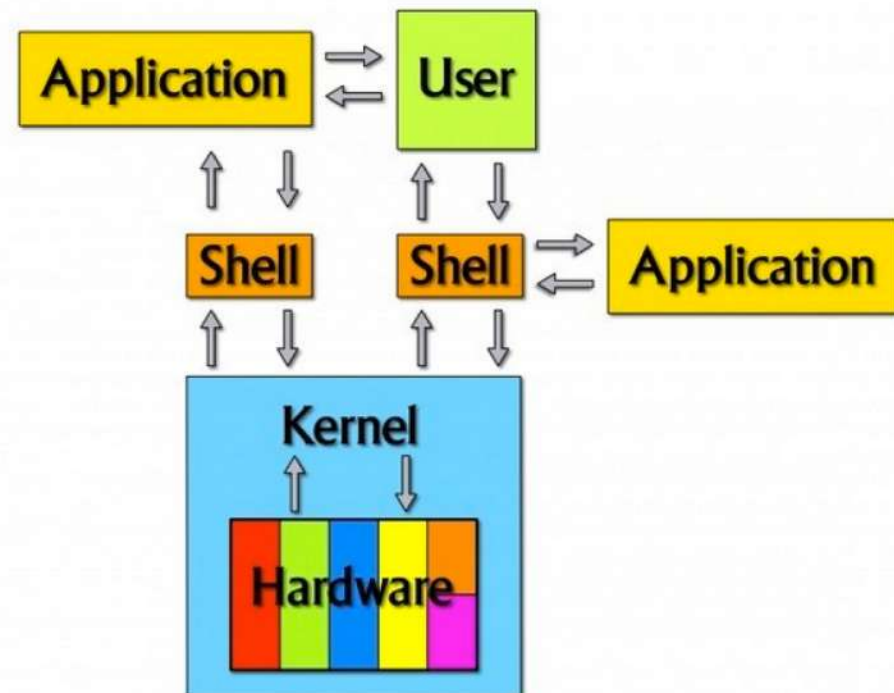
- ✓ Intérprete o terminal de comandos.
- ✓ **GNOME Terminal** es el emulador de terminal o intérprete de comandos textual del sistema operativo Ubuntu.
- ✓ Actúa directamente con **COMANDOS** sobre el Sistema Operativo.



```
luisana@luisana-VirtualBox:~$ pwd
/home/luisana
luisana@luisana-VirtualBox:~$ ls
Descargas  Escritorio  INTERNO  Plantillas  Videos
Documentos  Imágenes  Música  Público
luisana@luisana-VirtualBox:~$ cd Escritorio/
luisana@luisana-VirtualBox:~/Escritorio$ ls -ltr
total 8
drwxr-xr-x  2 luisana luisana 4096 oct 24 21:13 .
drwxr-xr-x 18 luisana luisana 4096 nov  8 10:02 ..
luisana@luisana-VirtualBox:~/Escritorio$
```

# La Shell de GNU/Linux y UNIX

- ✓ Un **SHELL** es un programa que interpreta comandos escritos del sistema operativo.
- ✓ Equivale al COMMAND.COM de MS-DOS
- ✓ Incluye un lenguaje de programación para **procesamiento por lotes de comandos (scripts)**.
- ✓ Distintos tipos de Shell:
  - Sh
  - C-Shell, K-Shell
  - **Bash-Shell (Linux)**





# La Shell de GNU/Linux y UNIX

## ❑ Tipos de Shell

**Shell Bourne (sh).** Creado por S. Bourne, es el más utilizado en la actualidad. Su símbolo del sistema es \$. Es el shell estándar y el que se monta en casi todos los sistemas UNIX/Linux

**Shell Korn (ksh).** Escrito por David Korn, amplía el shell del sistema añadiendo historial de órdenes, edición en línea de órdenes y características ampliadas de programación

**Shell job (jsh).** Incorpora algunas características de control al shell estándar del sistema

**C-Shell (csh).** Procedente del sistema BSD, proporciona funciones tales como control de trabajos, historial de órdenes, etc. Ofrece importantes características para los programadores que trabajan en lenguaje C. Su símbolo del sistema es %

**Bourne Again Shell (bash).** Fue creado para usarlo en el proyecto GNU. BASH, por lo tanto, es un shell o intérprete de comandos GNU que incorpora la mayoría de distribuciones de Linux. Es compatible con el shell sh

✓ Símbolo del sistema (\$ sh , % Csh) o **PROMPT**



# Prompt del Sistema Operativo

Se llama ***prompt*** al carácter o conjunto de caracteres que se muestran en una línea de comandos para indicar que está a la **espera de órdenes**. Varía en función del intérprete de comandos.

En el Bourne Shell y derivados como Bash:

- Carácter \$ para usuarios sin privilegios
- Carácter # para el administrador

usuario@máquina:~\$

root@máquina:~#

En MS-DOS y Windows: C: \>

# Bash de GNU/Linux

```
luisana@luisana-VirtualBox: ~  
luisana@luisana-VirtualBox:~$ ls -altr  
total 100  
drwxrwx--- 1 root    vboxsf  2584 may 20 19:25 INTERNO  
drwxr-xr-x 3 root    root    4096 oct 24 21:05  
-rw-r--r-- 1 luisana luisana  807 oct 24 21:05 .profile  
-rw-r--r-- 1 luisana luisana 3771 oct 24 21:05 .bashrc  
-rw-r--r-- 1 luisana luisana  220 oct 24 21:05 .bash_logout  
drwx----- 3 luisana luisana 4096 oct 24 21:13 local  
drwxr-xr-x 2 luisana luisana 4096 oct 24 21:13 Escritorio  
luisana@luisana-VirtualBox:~$ env  
SHELL=/bin/bash  
SESSION_MANAGER=local/luisana-VirtualBox:@/tmp/.ICE-unix/2013,unix/luisana-VirtualB  
X:/tmp/.ICE-unix/2013  
COLORTERM=truecolor  
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg  
XDG_MENU_PREFIX=gnome-  
GNOME_DESKTOP_SESSION_ID=this-is-deprecated  
GTK_IM_MODULE=ibus  
QT4_IM_MODULE=ibus  
GNOME_SHELL_SESSION_MODE=ubuntu  
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh  
XMODIFIERS=@im=ibus  
DESKTOP_SESSION=ubuntu  
SSH_AGENT_PID=1887  
GTK_MODULES=gail:atk-bridge  
PWD=/home/luisana  
LOGNAME=luisana  
XDG_SESSION_DESKTOP=ubuntu
```



# Bash de GNU/Linux (Introducción)

**.bashrc** es un script (programa, guion o archivo por lotes) que Bash ejecuta siempre que se arranca y antes de que aparezca el prompt del sistema. Está formado por comandos del sistema (los mismos que escribe el usuario). Sirve para customizar/personalizar el entorno del usuario.

**env, echo** son comandos que nos muestra las **VARIABLES** de la SHELL (empiezan por \$).

```
$ echo $HOME | $ env
```

# Gestión del Sistema de Archivos

## ❑ Organización de la información

- ✓ La información se estructura en una serie de ficheros organizados según una estructura jerárquica de directorios en forma de árbol invertido.
- ✓ El árbol padre de esta estructura es el dispositivo raíz o sistema de ficheros **raíz** y se representa por el carácter **/**

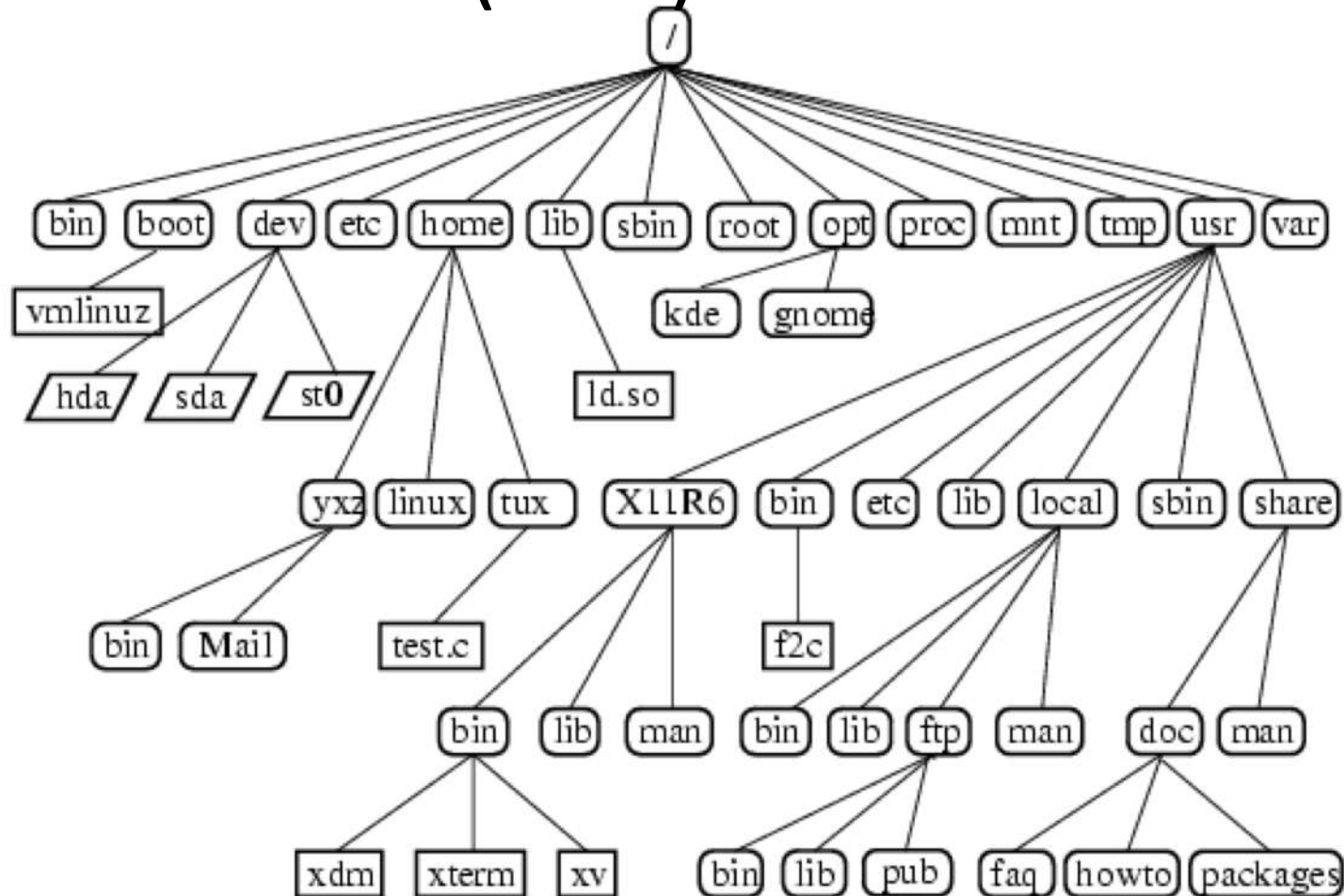
# Gestión del Sistema de Archivos

## ❑ Linux versus MS-DOS

- ✓ En MS-DOS cada dispositivo (disquetera, CD-ROM, particiones del disco duro, etc) tiene una unidad lógica diferente.
- ✓ En GNU/Linux toda la información de los dispositivos de almacenamiento se integran en el árbol raíz.
- ✓ A la hora de especificar las rutas de los ficheros y directorios usaremos el carácter **/** a diferencia de MS-DOS que es \

## Gestión del Sistema de Archivos

### □ Árbol raíz (*root*)





# Estructura de Directorios

Estructura significativa no organizada al azar. Cada directorio tiene un significado y archivos específicos. Gran ordenación. El sistema espera encontrar ciertos archivos de configuración y sistema, en determinados directorios.

Directorios más comunes en Linux con fines específicos:

- **/bin**: programas ejecutables para todos los usuarios
- **/boot**: archivos del gestor de arranque, imágenes del kernel (núcleo del sistema operativo)
- **/dev**: archivos que representan dispositivos físicos
- **/etc**: archivos de configuración del sistema
- **/home**: directorios personales de los usuarios
- **/lib**: bibliotecas y librerías del sistema



# Estructura de Directorios

- **/mnt**: directorio para ubicar los puntos de montaje para los dispositivos de bloque o unidades de disco.
- **/opt**: paquetes de software no esenciales para el sistema.
- **/proc**: sistema de archivos virtual de los procesos.
- **/root**: directorio del administrador del sistema o root.
- **/sbin**: programas esenciales para el sistema, ejecutables de root
- **/tmp: directorio temporal con archivos temporales.**
- **/usr**: programas no esenciales, documentación de aplicaciones, de usuario.
- **/var**: datos que son modificados en tiempo real por los programas: registros de errores, informaciones, logs del sistema y aplicaciones.

# Implementación del Sistema de Archivos

- ✓ ¿Cómo se organizan los archivos en el disco duro o de almacenamiento y cómo se opera con ellos? →  
Lo que hace el Gestor de Archivos
- ✓ El disco de almacenamiento se divide en múltiples bloques (sectores) para contener los archivos. En función del manejo de los bloques/sectores, tenemos diferentes tipos de Sistema de Archivos:
  - Microsoft Windows: FAT (FAT16 y FAT32) y NTFS
  - Unix/Linux: Ext3fs/Ext4fs sistemas de archivos extendidos y HPFS (High Performance File System)

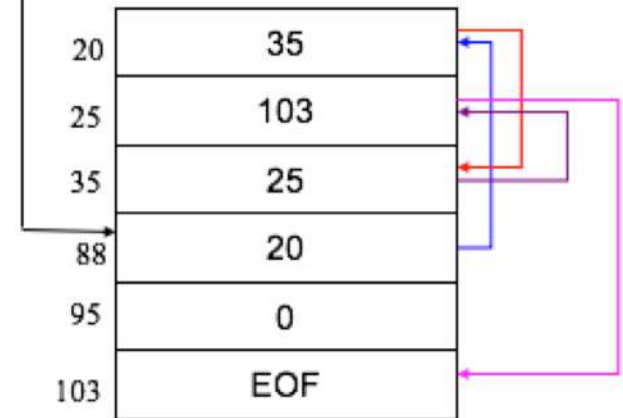
## Implementación del Sistema de Archivos

### ❑ FAT – MICROSOFT WINDOWS

FAT: *File Allocation Table*. Asignación de bloques y sectores del archivo. Utiliza el índice en el directorio y tabla con lista ligada.

○ Entrada directorio

test.txt	.....	88
----------	-------	----



# GNU/Linux

## Implementación del Sistema de Archivos

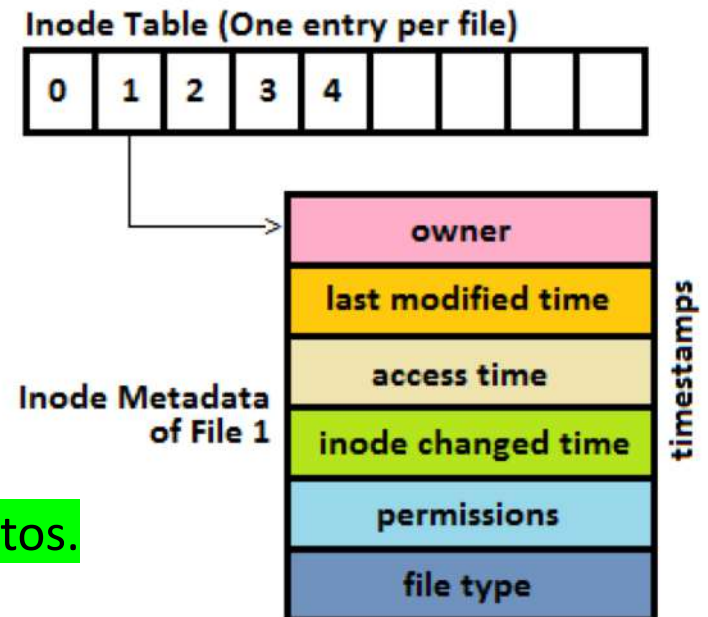
### ❑ LINUX

La información relativa a los ficheros se encuentra en las **Tablas de Nodos de Índices** o **Tabla de Inodes** y contienen la información administrativa de los ficheros.

Los **Inodes** contienen los metadatos del archivo:

- ✓ Identificador de grupo.
- ✓ Identificador de usuario.
- ✓ Fecha y hora de la última modificación.
- ✓ Fecha y hora del último acceso al fichero.
- ✓ Número de enlaces.
- ✓ Tipo de fichero (ordinario, directorio,...).
- ✓ Punteros a los bloques que contienen los datos.

### Inode Entry



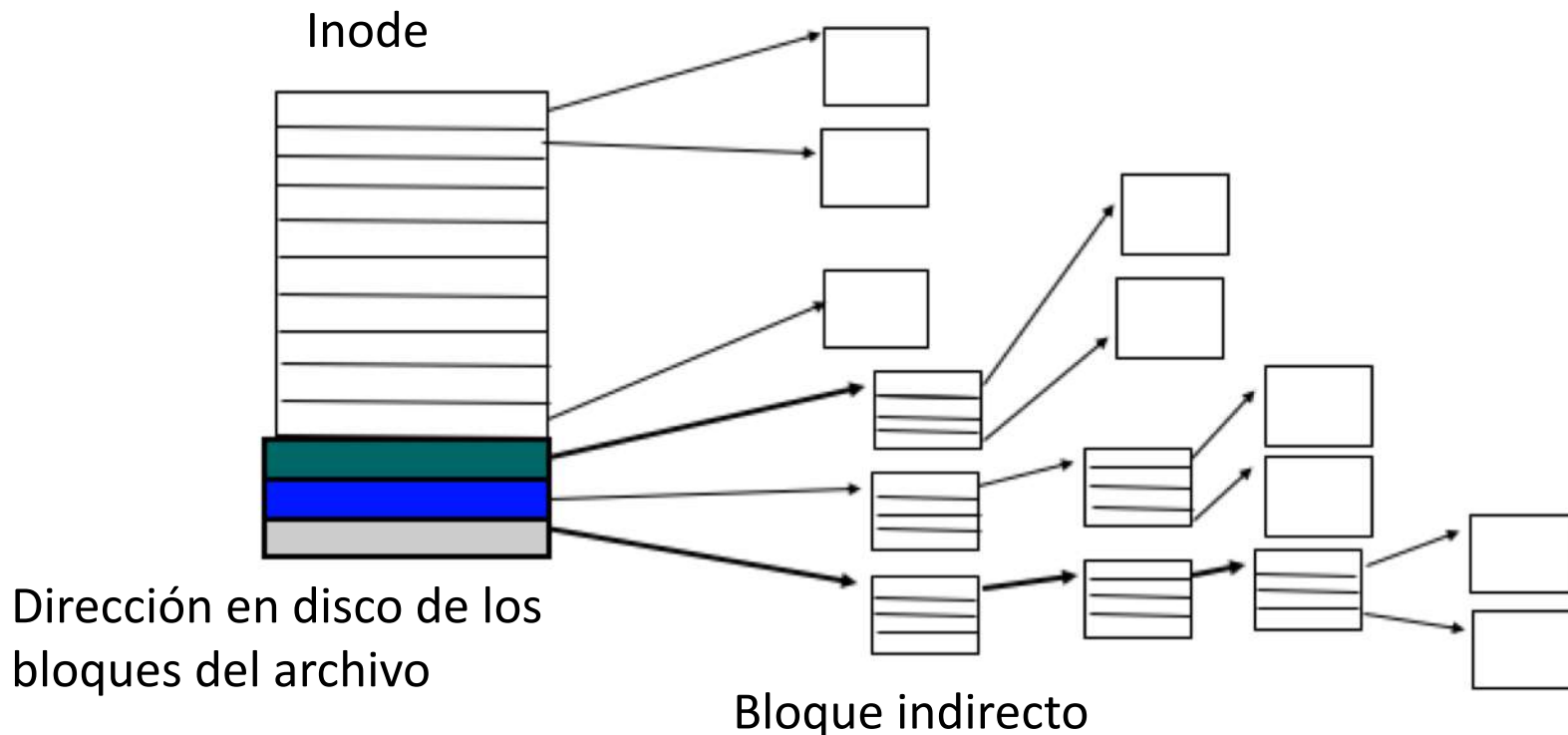


# Implementación del Sistema de Archivos



# LINUX

En el inode está la información al resto de bloques del fichero (puntero a los bloques que contienen los datos).



# Gestión del Sistema de Archivos

## ❑ Tipos de archivos o ficheros

- ✓ **Fichero ordinario.** Contiene datos (ficheros de texto o ASCII) y/o programas (ficheros binarios o ejecutables).
- ✓ **Directorios:** Ficheros que permiten la organización jerárquica en forma de árbol invertido.
- ✓ **Ficheros especiales de dispositivos.** Linux (Unix) trata todos los dispositivos físicos y lógicos como ficheros. Se encuentran en el /dev (/dev/fd0 : primera unidad de floppy disk). Pueden ser de modo bloque o carácter.

# Gestión del Sistema de Archivos

## ❑ Tipos de archivos o ficheros

- ✓ **Enlaces simbólicos:** es un enlace lógico y hace referencia a un fichero físico.
- ✓ **Tubería o pipes:** ficheros especiales que se utilizan para comunicar procesos. En la Shell se representa con el carácter ‘|’ y se usa en filtros.
- ✓ **Sockets:** ficheros especiales para comunicación entre procesos ubicados en diferentes máquinas.
- ✓ **Ficheros ocultos:** archivos que empiezan por .

# Gestión del Sistema de Archivos

## ❑ Referencia de archivos y directorios

- ✓ **Ruta absoluta.** Consiste en especificar el camino entero que se debe seguir desde el directorio raíz / hasta el fichero o directorio.

`/home/user/pepe/Documentos/personales/readme.txt`

- ✓ **Ruta relativa.** Es el camino que se ha de seguir desde el directorio activo hasta llegar al fichero o directorio.

`Documentos/personales/readme.txt`



# Gestión del Sistema de Archivos

## ❑ Directorio actual y directorio padre

✓ Todo directorio contiene dos entradas especiales que se crean al crear el directorio.

- Directorio **actual o activo**: “.” (carácter punto).  
Hace referencia al propio directorio.

./Documentos == Documentos

- Directorio padre o anterior en el árbol jerárquico:  
“..” (dos puntos).

../Documentos

# Comandos directorios y archivos

- ✓ Listar Archivos y Directorios

```
$ ls [opciones] [fichero]  
$ ls | $ ls -altr
```

- ✓ Directorios

```
pwd | cd | mkdir | rmdir | ls
```

- ✓ Archivos

```
cat | cp | mv | rm | find | ln  
grep | more | sort | tr
```

Documento EVA: “Comandos Linux Archivos y Directorios”

```
$ man comando
```

# GNU/Linux

## Metacaracteres

Significado especial para la Shell.

**\*** : sustituye a una cadena de cero o más caracteres

```
$ ls fich*.txt
```

**?** : sustituye a un único carácter

```
$ ls fich?.txt
```

**[ ]** : sustituye a cualquier carácter de los incluidos entre los corchetes, a una secuencia de caracteres o a un intervalo

```
$ ls fich[1234567].txt
```

```
$ ls fich[1-7].txt
```

**~** : sustituye la ruta al directorio personal (variable \$HOME)

```
$ cd ~ | $ cd $HOME | $ cd /home/pepe
```



# Redireccionamiento

Todo proceso en Linux tiene:

- Entrada estándar: es de donde se espera recibir los datos de entrada (la fuente de entrada).

Por defecto: teclado

- Salida estándar a la que se vuelca su salida normal (destino de la salida).

Por defecto: pantalla

- Salida estándar de error donde se vuelcan los mensajes de error que ocurran en el proceso.

Por defecto: pantalla

Podemos cambiar este comportamiento desde la Shell

# Redireccionamiento

Un comando es un proceso para la Shell.

Ejemplos de cambios en el redireccionamiento:

- Redireccionamiento de entrada: <

```
$ sort < empleados.txt
```

- Redireccionamiento de salida: > >>

```
$ echo "Hola" > hola.txt
```

```
$ echo "Hola" >> hola.txt
```

- Redireccionamiento del error: 2>

```
$ ls fich_no_existe 2> /var/log/error.log
```

# Redireccionamiento – Procesos – E/S

En Linux toda la E/S de los procesos se hace a través de ficheros. Y cada proceso tiene asociados 3 ficheros para la E/S (*stdin*, *stdout*, *stderr*).

Y los ficheros en Linux se manejan (manejador de archivos y tablas de *inodes*) con un **descriptor de archivo**: clave a la estructura de datos que contiene los detalles del proceso que se lanza y sus ficheros.

Nombre	Descriptor de fichero	Destino por defecto
entrada estándar ( <i>stdin</i> )	0	teclado
salida estándar ( <i>stdout</i> )	1	pantalla
error estándar ( <i>stderr</i> )	2	pantalla

# Tuberías (pipes) o filtros

- ✓ Otro tipo de redireccionamiento en Linux.
- ✓ Conexión de la salida estándar de un comando o proceso con la entrada estándar de otro. Los comandos se encadenan para que la salida de uno sea la entrada del otro.
- ✓ Se usa el carácter o símbolo |

Ejemplo:

```
$ grep disquetera perifericos.txt | sort > disqueteras.txt
```

El comando **grep** selecciona aquellas líneas del **fichero.txt** que contengan la palabra **disquetera**. La salida predeterminada de **grep** es la pantalla, pero en este caso, como hemos puesto un **pipe** será pasada a la entrada de **sort**, que ordenará las líneas y las volcará al fichero **disqueteras.txt**.

# Enlaces simbólicos

**Enlace lógico** que hace referencia a un fichero físico, archivo o directorio. Los enlaces pueden ser:

- **Duros:** se crea un nombre de archivo en el mismo inodo. No se puede usar con directorios o en archivos de diferentes particiones.
- **Blandos:** apunte indirecto a un fichero o directorio en distintas particiones.

**`ln -s source_file symbolic_link`**

Ejemplo:

```
$ ln -s my_file.txt my_link.txt
```

```
$ ls -l my_link.txt
```

```
lrwxrwxrwx 1 linuxize users 4 Nov 2 23:03 my_link.txt -> my_file.txt
```



# Permisos de los archivos

Usuarios y Grupos en Linux: todo usuario tiene una cuenta con un nombre y clave para acceder al sistema. Y cada usuario debe pertenecer a uno o varios **grupos de usuarios**. Linux provee una gestión de permisos y de protección de datos de un usuario respecto de otros.

Los tipos de permisos sobre archivos en Linux son los siguientes:

- ✓ **Lectura**: Permite fundamentalmente visualizar el contenido del archivo. Podemos visualizarlos con comandos como **ls, cat, cp**
- ✓ **Escritura**: Permite modificar el contenido del archivo. El archivo se puede editar con **vi, gedit**
- ✓ **Ejecución**: Permite ejecutar el archivo como si de un programa ejecutable se tratase. Se asigna a archivos Shell que incluyen procesamiento por lotes de comandos (**scripts**)

# GNU/Linux

## Permisos de los archivos

drwxr-xr-x	3	raul	raul	4096	2005-02-16	14:47	Desktop
drwxr-xr-x	5	raul	raul	4096	2005-02-16	14:47	GNUstep
-rw-r--r--	1	raul	raul	246417	2005-03-03	14:47	foto1.png
-rw-r--r--	1	raul	raul	232505	2005-03-03	16:26	carta2.abw
-rw-r--r--	1	raul	raul	239618	2005-03-03	18:15	informe.abw
drwxr-xr-x	2	raul	raul	4096	2005-02-16	14:47	tmp

Diagram illustrating the components of a file listing command output (ls -l):

- Permiso de Archivo:** Indicated by the first character (d for directory, - for file).
- Permisos de Propietario:** The next three characters (rwx).
- Permisos de Grupo:** The next three characters (rwx).
- Permisos de Otros:** The final three characters (r--).
- Enlaces:** The number in the second column (3, 5, 1, 1, 1, 2).
- Propietario:** The username in the third column (raul).
- Grupo:** The group name in the fourth column (raul).
- Tamaño:** The file size in bytes in the fifth column (4096, 4096, 246417, 232505, 239618, 4096).
- Fecha:** The date in YYYY-MM-DD format in the sixth column (2005-02-16, 2005-02-16, 2005-03-03, 2005-03-03, 2005-03-03, 2005-02-16).
- Hora:** The time in HH:MM format in the seventh column (14:47, 14:47, 14:47, 16:26, 18:15, 14:47).
- Nombre:** The filename in the eighth column (Desktop, GNUstep, foto1.png, carta2.abw, informe.abw, tmp).

- rw x rw x r--** → Donde para dar permiso o no, consideraremos 0 ó 1.
- 111 111 100** → El propietario y su grupo, tienen permisos de lectura, escritura y ejecución. El resto de usuarios, de lectura.
- 7 7 4** → Lo mismo en octal

# GNU/Linux

## Permisos de los archivos

**Comandos** que nos permiten modificar los permisos:

**chmod:** Se utiliza para cambiar los permisos del archivo o directorio. Varias maneras:

**\$ chmod 777 Ejer1.txt**

Asignamos permisos de lectura, escritura y ejecución (control total) a todos los usuarios sobre el fichero Ejer1.txt

**\$ chmod ugo+rwX Ejer1.txt**

Igual. Podemos variar el significado de la orden según utilicemos + (añadimos los permisos que se indican a ese usuario), - (le quitamos los permisos indicados), = (le asignamos los permisos).  
el grupo.

# GNU/Linux

## Permisos de los archivos

**chown:** Cambia el propietario del fichero o directorio. Como parámetro, le pasaremos el nombre del propietario y el nombre del archivo o directorio.

### **\$ chown alumno Cosas**

El propietario del directorio Cosas pase a ser el usuario alumno.

### **\$ chown -R alumno DirectorioCosas**

El propietario de la carpeta cambie de forma recursiva, es decir, que también afectará el cambio al contenido de ésta.

**chgrp:** Es análoga a la anterior, sólo que aquí, en lugar de cambiar el propietario, cambiamos el grupo.