

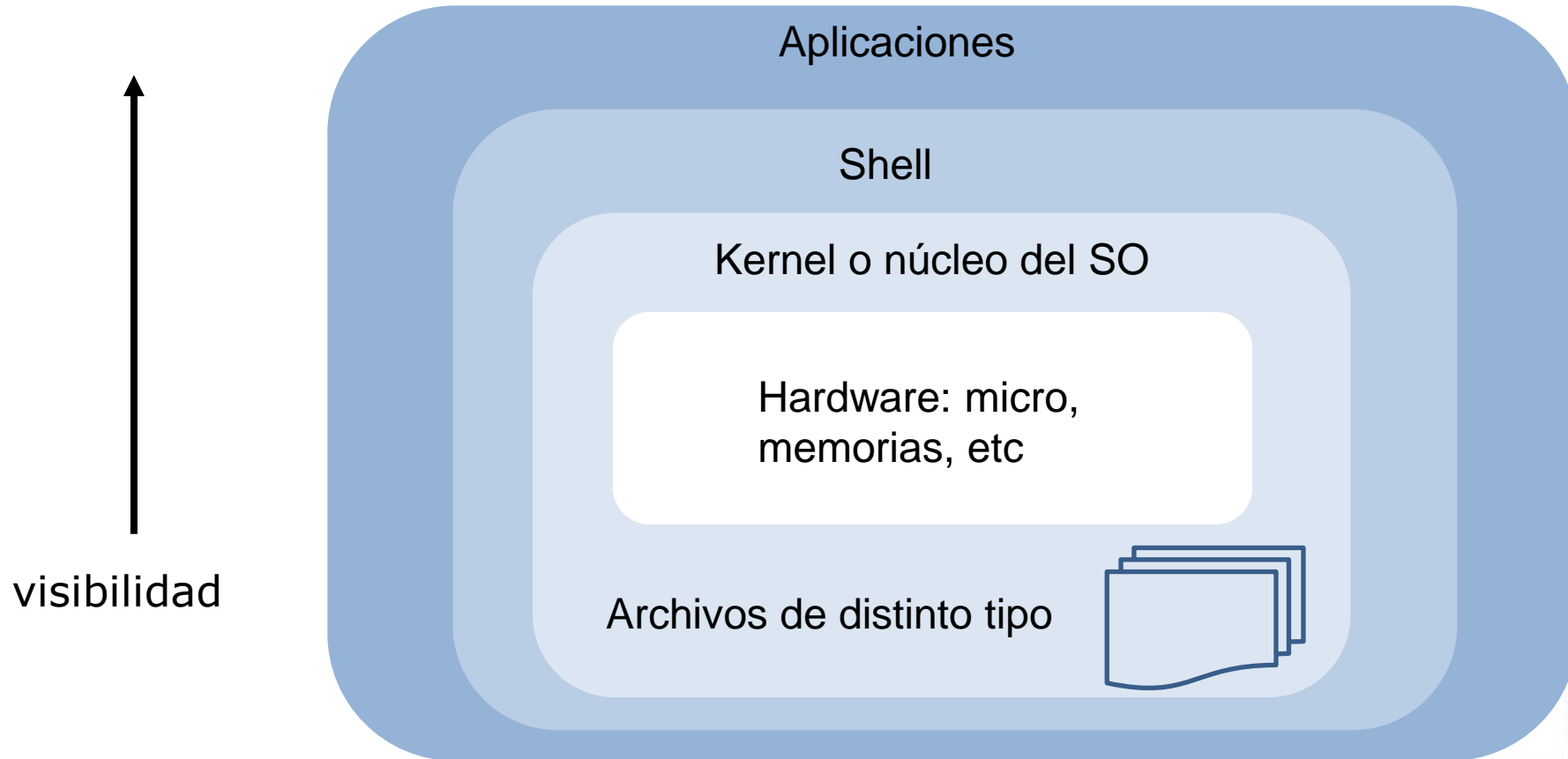
INFORMÁTICA II

LINUX

Segunda parte

¿Qué es Linux?

La estructura general es la siguiente:



Estructura general GNU/Linux

Básicamente formado por 2 componentes

- **Shell:** Es el equivalente al intérprete de comandos de Windows. Sirve para introducir órdenes, ejecutar programas, etc.
- **Núcleo:** Es la parte del sistema que interactúa con el hardware.



¿Qué es Linux?

- ☐ Un sistema operativo de libre distribución.
- ☐ Permite trabajar al usuario en dos modo:
 - monopuesto
 - modo de red
- ☐ Permite iniciar sesión: a) localmente en la máquina.
b) de forma remota.
- ☐ Es una versión de UNIX desarrollada para funcionar en plataformas de ordenador personal.
- ☐ También recibe el nombre de GNU/Linux o sistema operativo libre con licencia pública GNU.



Interfaz de comunicación con el usuario

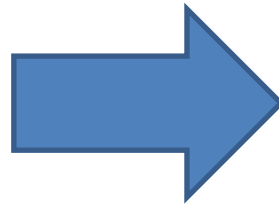
GNU/Linux dispone de dos interfaces de comunicación con el usuario:

- ❑ Una interfaz gráfica o GUI (**Graphics User Interface**).
- ❑ Una Interfaz de Línea de Comandos, **CUI, Command User Interface**.



Interfaz **G**ráfica de Comunicación con el **U**suario

GUI, es un programa en ejecución o proceso lanzado de manera automatizada desde la interfaz de línea de comandos después del arranque del equipo.



Implica que

El control siempre lo tenemos desde la interfaz de línea de comandos, y desde ella podremos controlar todo el entorno gráfico



Elementos de la interfaz gráfica

Elementos básicos del escritorio de Linux
(similares a Windows):

- Iconos
- Ventanas
- Barra de tareas
- Menús de trabajo
 - Aplicaciones
 - Lugares
 - Sistema
- Área de notificación



Elementos de la interfaz gráfica

Linux (a diferencia de Windows) puede usar distintos tipos de interfaz gráfica. Los más importantes son:

- KDE
- Gnome2
- Gnode-Shell

Hay diferencias entre ellos en cuanto a opciones y configuraciones. Usar uno u otro depende del gusto del usuario.



Interfaz de línea de comando: CUI

La administración del sistema a través de la interfaz de línea de comandos en GNU/Linux es más **potente** y **eficiente** que la GUI (depende de CUI). Para acceder a la CUI es necesario acceder a una terminal de comandos.



Gestión de Usuarios y Permisos en GNU/Linux

La administración de cuentas de usuario en GNU/Linux puede realizarse a través de aplicaciones en:

- ❖ Modo gráfico (**GUI, Graphics User Interface**).
- ❖ Modo comando (**CUI, Command User Interface**).



Gestión de Usuarios y Permisos en GNU/Linux

PRECAUCIÓN:

Para GUI, la gestión de cuentas de usuarios son dependientes de la distribución de GNU/Linux instalada, por lo que el nombre de dicha aplicación y su aspecto variará en cada una de ellas (*Ubuntu, OpenSuse, etc.*).

Para CUI (*Interfaz de Usuario de Línea de Comandos*), los comandos GNU/Linux utilizados suelen ser independientes de la distribución.



Sistemas de archivos

Un **sistema de archivos** es el modo con el cual el Sistema operativo organiza los archivos en el disco duro, gestionándolo para que los datos estén de forma estructurada y sin errores.

La estructura de archivos es **jerárquica de directorios o carpetas** en forma de árbol invertido, **el directorio principal (directorio raíz) es el directorio /**, del que cuelga toda la estructura del sistema.

Cada carpeta o directorio puede contener un número arbitrario de archivos, y también puede contener otras carpetas.



Sistemas de archivos

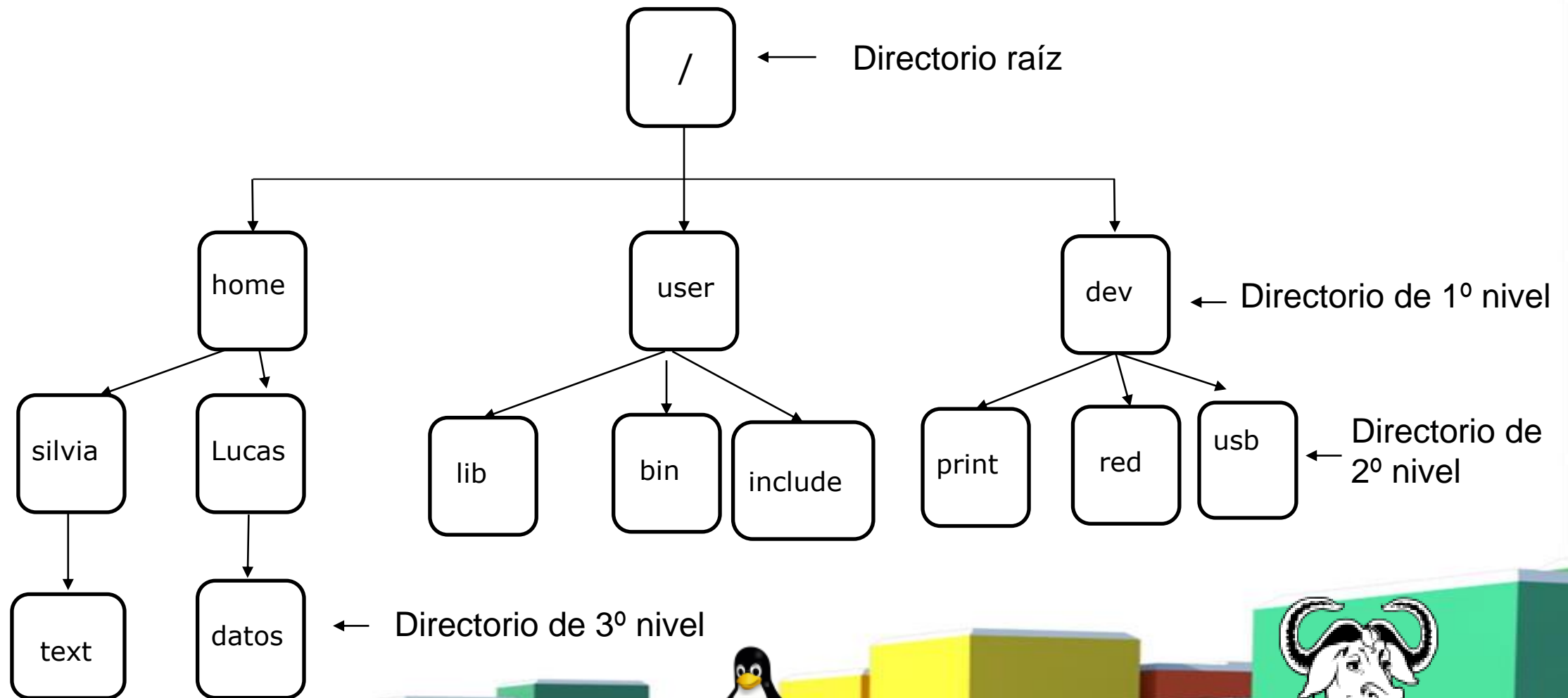
¿Por qué?

Seguro. Mantiene la privacidad de datos y permisos: lectura, escritura y ejecución de usuario y grupos de usuarios.

El sistema de ficheros típico de Linux es el Ext2. Linux es capaz de entender otros tipos de sistemas de ficheros.

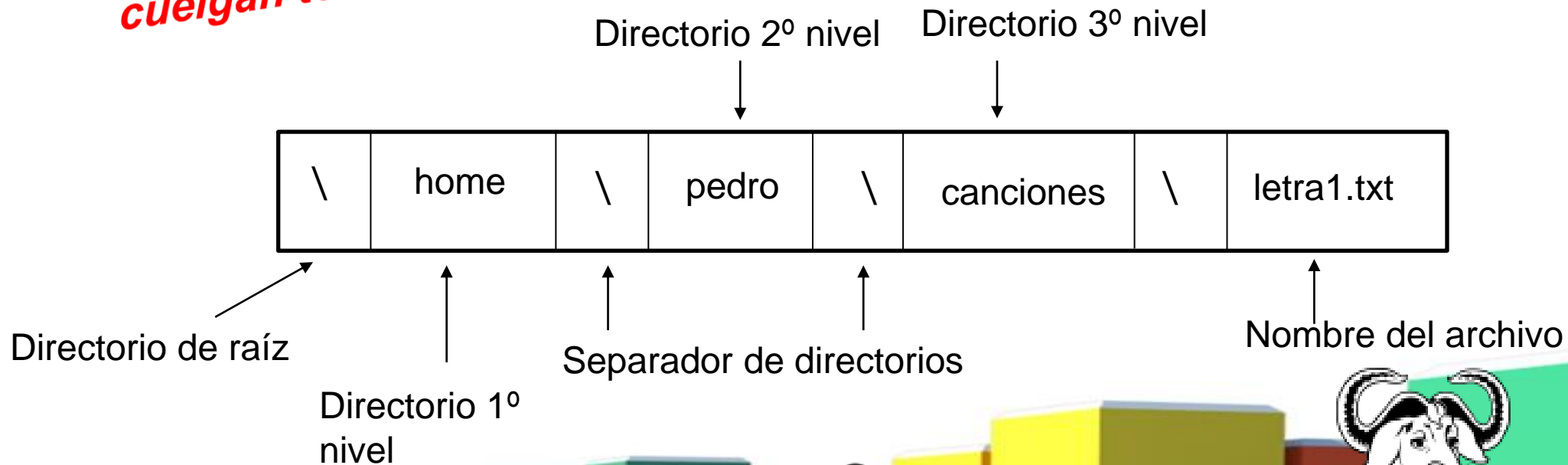


Sistemas de archivos: jerárquica (árbol invertido)



Sistemas de archivos

El sistema de archivos en GNU/Linux se caracteriza por tener una estructura en árbol invertido con una única raíz (root, /) de la cual cuelgan todos los directorios y archivos del sistema.



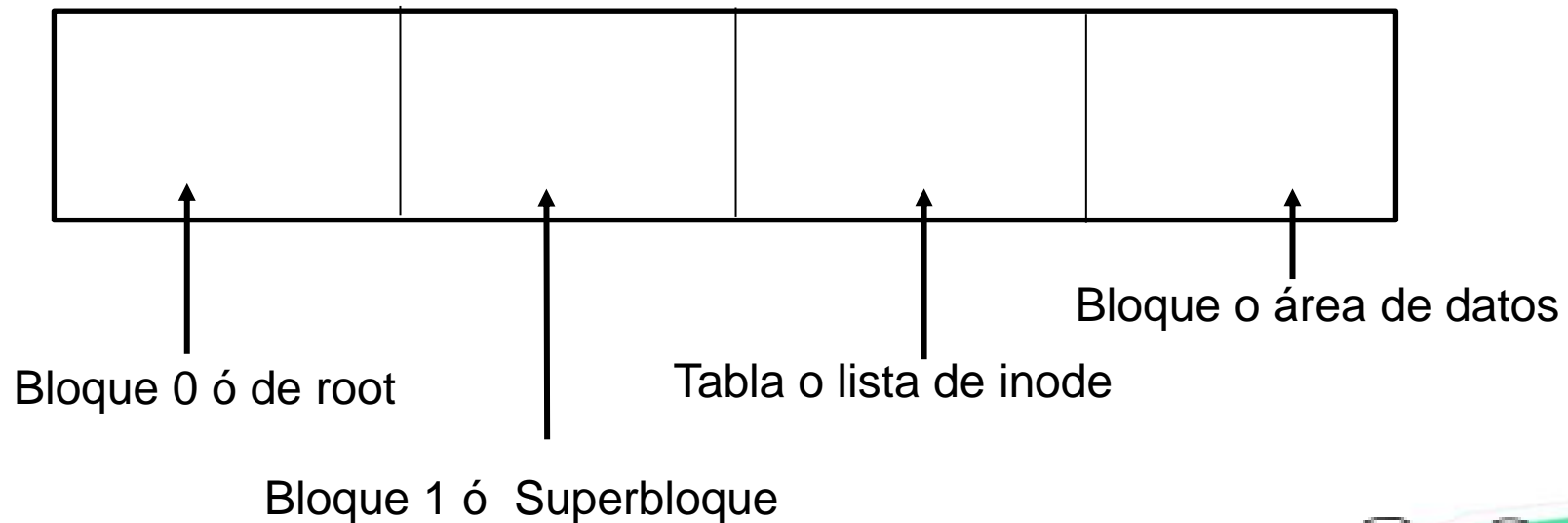
Sistemas de archivos

- Un sistema Linux puede manejar uno o varios discos Duros.
- Cada disco puede contener uno o varios sistemas de archivos.
- Los sistemas de archivo son particiones lógicas del disco.
- El kernel del SO trabaja con el sistema de archivos a nivel lógico y no accede a los discos a nivel físico.
- Sistema de archivos se componen de una secuencia de bloques lógicos, cada uno de tamaño fijo (homogéneo).
- Cada bloque es de 512 bytes o múltiplos.



Sistemas de archivos

La estructura de un archivo en Linux de cualquier clase, se la puede esquematizar como un conjunto de cuatro bloques: :



Sistemas de archivos

- ✓ **Bloque de root o carga o bloque cero** de cada sistema está reservado para almacenar un programa que utiliza el sistema para gestionar el resto de las partes del sistema de ficheros (arranque).
- ✓ **Superbloque o bloque uno** describe el estado de un sistema de ficheros. Ej: tamaño, qué espacio libre queda
- ✓ **Tabla de inodos o números índices.** Este número identifica la ubicación del archivo dentro del área de datos.
- ✓ **Bloque ó área de datos.**



Sistemas de archivos

Inodo contiene información de un archivo o directorio. Guarda la siguiente información:

1. **Identificador de dispositivo** que alberga al sistema de archivos.
2. **Número de inodo** que identifica al archivo dentro del sistema de archivos.
3. **Longitud del archivo** en bytes.
4. **Identificador de usuario del creador** o un propietario del archivo con derechos diferenciados.
5. **Identificador de grupo** de usuarios con derechos diferenciados.



Sistemas de ficheros

Continuación de
I-node

6. **Modo de acceso:** capacidad de leer, escribir, y ejecutar el archivo por parte del propietario, del grupo y de otros usuarios.
7. **Marcas de tiempo** con las fechas de **última modificación** (mtime), **acceso** (atime) y **de alteración** del propio inodo (ctime).



Sistemas de ficheros

Continuación de
I-node

8. **Número de enlaces** (hard links) es el número de nombres (entradas de directorio) asociados con este inodo. Si múltiples nombres están asociados a un mismo inodo entonces todos los nombres son equivalentes entre sí. El número de enlaces es empleado por el sistema operativo para eliminar el archivo del sistema de archivos, tanto el inodo como el contenido, cuando se han borrado todos los enlaces y el contador queda a cero.



Sistemas de ficheros

Bloque o Área de datos ocupa el resto del sistema de archivo.

En esta zona, está almacenado el contenido del archivo a los que hacen referencias los inode.

Un directorio es un archivo que contiene los nombres de los archivos (o directorios) que contiene junto con el número del **inodo** que contiene la información de cada uno de ellos.



Sistemas de archivos

Linux soporta gran variedad de sistemas de archivos, desde sistemas basados en discos, como pueden ser **ext2**, **ext3**, **ReiserFS**, **XFS**, **JFS**, **UFS**, **ISO9660**, **FAT**, **FAT32** o **NTFS**, a sistemas de archivos que sirven para comunicar equipos en la red de diferentes sistemas operativos, como **NFS** (utilizado para compartir recursos entre equipos Linux) o **SMB** (para compartir recursos entre máquinas Linux y Windows).



Sistemas de archivos

Los sistemas de archivos indican el modo en que se gestionan los archivos dentro de las particiones.

Sus características son:

- ✓ previsión de apagones
- ✓ posibilidad de recuperar datos
- ✓ indexación para búsquedas rápidas
- ✓ reducción de la fragmentación para agilizar la lectura de los datos
- ✓ etc.



Sistemas de archivos

Tipos de sistemas de archivos más representativos:

• **ext2:** Hasta hace poco era el sistema estándar de Linux. Fragmentación muy baja, lento manejando archivos de gran tamaño. Las principales ventajas:

- Compatible con sistemas de archivos grandes
- Particiones de disco de hasta 4TB
- Archivos de hasta 2GB de tamaño.
- Nombres de archivos de hasta 255 caracteres.
- Gran estabilidad.
- Actualización.



Sistemas de archivos

ext3: versión mejorada de ext2, con previsión de pérdida de datos por fallos del disco o apagones. Desventaja: imposible recuperar datos borrados. Compatible con el sistema de ficheros ext2. Hoy es el más difundido dentro de la comunidad GNU/Linux. Es el estándar. Sus ventajas frente a ext2 son:

- ✓ Actualización: los dos sistemas comparten el mismo formato, es posible actualizar a ext3, incluso aunque el sistema ext2 esté montado.
- ✓ Fiabilidad y mantenimiento.



Sistemas de archivos

ext4: Es la última versión de la familia de sistemas de archivos ext.

Ventajas:

- ✓ eficiencia (menor uso de CPU, mejoras en la velocidad de lectura y escritura)
- ✓ Ampliación de los límites de tamaño de los archivos, llegando hasta 16TB, y del sistema de archivos, que puede llegar a los 1024PB (PetaBytes).

ReiserFS: Sistema de ficheros de última generación para Linux.

Características:

- ✓ Organiza los archivos para agilizar sus operaciones.
- ✓ El problema de ser tan actual es que muchas herramientas (ej. Recuperación de datos) no lo soportan.



Sistemas de archivos

•**swap**: Sistema de archivos para partición de intercambio de Linux. Todos los sistemas Linux necesitan una partición de este tipo para cargar los programas y no saturar la memoria RAM cuando se excede su capacidad. Linux también ofrece soporte para sistemas de ficheros de Windows, como FAT, FAT32 y NTFS.

Tanto para FAT como para FAT32, Linux tiene soporte completo y estable de escritura y lectura, mientras que para NTFS, y con las últimas versiones del kernel, sólo se puede acceder de manera estable en modo lectura. En modo escritura todavía está en fase experimental y no es estable.



Directorios más importantes

Algunos directorios importantes de Linux en Fedora (otras distribuciones pueden tener nombre diferente):

/ es el directorio raíz. De aquí cuelgan todos los directorios del sistema. Hay varios subdirectorios importantes:

/bin contiene ficheros de comandos ejecutables utilizables por todos los usuarios. Aquí tenemos los programas que pueden lanzar todos los usuarios del sistema.

/sbin es para ejecutables de uso exclusivo por el superusuario. Son los necesarios para arrancar y montar el directorio /usr.



Directorios más importantes

/home directorio donde se encuentran los directorios personales de los usuarios del sistema.

/usr contiene utilidades y programas generales de usuario:

/usr/bin contiene programas de uso general.

/usr/share contiene archivos compartibles, independientes de la arquitectura.

/usr/share/doc contiene cierta documentación del sistema.

/usr/share/man contiene los manuales.

/usr/etc contiene archivos de configuración de uso global.



Directorios más importantes

/usr/include contiene las cabeceras de C y C++.

/usr/lib contiene las bibliotecas de nuestros programas.

/usr/sbin contiene los programas de administración del sistema.

/usr/src contiene los códigos fuente de nuestros programas.

/dev contiene archivos especiales de bloques y caracteres asociados a dispositivos hardware. Aquí encontramos todos los dispositivos físicos del sistema (todo nuestro hardware).



Directorios más importantes

/lib contiene librerías y compiladores del sistema. Contiene las bibliotecas necesarias para que se ejecuten los programas que tenemos en /bin y /sbin únicamente.

/proc contiene los archivos que reciben o envían información al núcleo. No deberíamos modificar el contenido de este directorio.

/etc contiene los ficheros de configuración y utilidades para la administración.

/var contiene ficheros para el administrador. Este directorio contiene información variable, como registros, datos de los servidores, etc.



Directorios más importantes

/boot contiene los archivos de configuración del arranque del sistema, como por ejemplo GRUB.

/media contiene unidades físicas que tenemos montadas: discos duros, unidades de DVD, pen drives, etc.

/opt sirve para admitir ficheros nuevos creados tras la modificación del sistema. Punto de montaje desde el que se instalan paquetes de aplicación adicionales. Se usa para instalar aplicaciones que no vienen en los repositorios, por ej., aquellas que compilamos a mano.

/tmp es donde se almacenan los archivos temporales.



Sistemas de archivos

En disco para Ext2 se organiza así:

Grupo bloques 1					Grupo bloques 2		
Bloque de arranque	Súper bloque	--	Tabla de node I	Bloque de datos	Súper bloque	Tabla de node I	--



Directorios más importantes

Además de los anteriores, hay 2 directorios especiales:

Directorio actual (.): directorio especial que hace referencia al directorio en el que estamos. Si referenciamos al directorio . nos estaremos refiriendo al directorio actual.

Directorio padre (..): directorio especial que hace referencia al directorio padre del directorio en el que estamos. Si referenciamos al directorio .. nos estaremos refiriendo al directorio padre del actual. El único directorio que no tiene directorio padre es el directorio raíz /.

Estos directorios son útiles para hacer referencia a **rutas relativas**.



Archivos y Tipos de archivos

Existen 4 tipos diferentes de archivos (puede cambiar según la bibliografía):

- Archivos Ordinarios
- Directorios de Archivos
- Archivos Especiales de Dispositivos
- Archivos Ocultos



Archivos Ordinarios

Contienen caracteres organizados conjuntamente en una forma elegida por el usuario o una aplicación particular de un programa.

- ❖ Son colecciones de bytes de 8 bits. Pueden ser documentos, ejecutables, programas fuentes, etc
- ❖ El contenido es interpretado por los programas de aplicación.



Archivos Ordinarios

Cada archivo ordinario tiene los siguientes atributos:

- ☐ Nombre del archivo (no necesariamente único)
- ☐ Número único llamado i-node, dentro de file system
- ☐ Tamaño en bytes
- ☐ Fecha de creación
- ☐ Fecha y hora del ultimo acceso
- ☐ Un juego de permisos de acceso.
- ☐ Un dueño y grupo al que pertenece



Archivos Ordinarios

Sobre los permisos de acceso:

Aseguran la privacidad y la seguridad de los archivos. El dueño provee los permisos de lectura, escritura y ejecución; también el control de acceso, el cual, lo puede hacer el propietario, un grupo de usuarios o cualquier otro usuario. **El propietario de un archivo es su creador y como tal, tiene todos los permisos.** Otros usuarios pueden leer los archivos de otro propietario, pero no pueden escribir en ellos.



Directorios de Archivos

- ✓ Contienen el nombre de cada archivo o directorio residente, dentro del directorio dado.
- ✓ A los directorios se les puede asignar los permisos apropiados de acceso para asegurar la privacidad y la seguridad.
- ✓ El propietario de un directorio puede leer, crear o remover archivos dentro de su directorio.
- ✓ Un usuario puede leer archivos dentro del directorio de otro, pero no puede añadir o remover ningún archivo.



Archivos Especiales de Dispositivos

- ✓ Corresponden a los dispositivos físicos como: disco duro, unidades de cintas, impresoras, las terminales y sistemas de memoria, estos dispositivos se encuentran en el directorio /dev.
- ✓ Todos los dispositivos físicos son archivos que contienen toda la información para manejarlos, por lo que las referencias a esos archivos se convierten en comandos de hardware que ejecutan las ordenes especificadas.



Archivos ocultos

- ✓ Los archivos que comienzan con un punto (.) se llaman archivos ocultos, estos son archivos Shell de propósitos especiales.
- ✓ Generalmente se ocultan para no causar confusiones con los otros archivos.
- ✓ Al listar los archivos de un directorio, los archivos que comiencen con un punto (.) no se despliegan.
- ✓ Usualmente en cada directorio aparecen dos archivos ocultos especiales:
 - punto (.) sinónimo del directorio de trabajo actual
 - dos puntos (..), es el padre del directorio de trabajo actual.



Convenciones

- ❑ Archivo, directorio o dispositivo en el sistema operativo tiene un nombre y una ruta específica.
- ❑ **La ruta o path es un camino** de localización de los archivos o directorios en el sistema mediante una cadena de caracteres concreta.
- ❑ **La ruta completa es única dentro de un sistema**
- ❑ Nombre de archivo es único solamente en su directorio y pueden no ser únicos en los subdirectorios.



Rutas en Linux

- **Ruta absoluta:** es la cadena de caracteres completa del recurso, parte del directorio raíz hasta llegar al archivo concreto que se está buscando.
- **Ruta relativa** representa una parte de la ruta absoluta, la ruta relativa contiene la cadena de caracteres que permiten localizar el recurso desde el directorio actual desde el que se está trabajando.



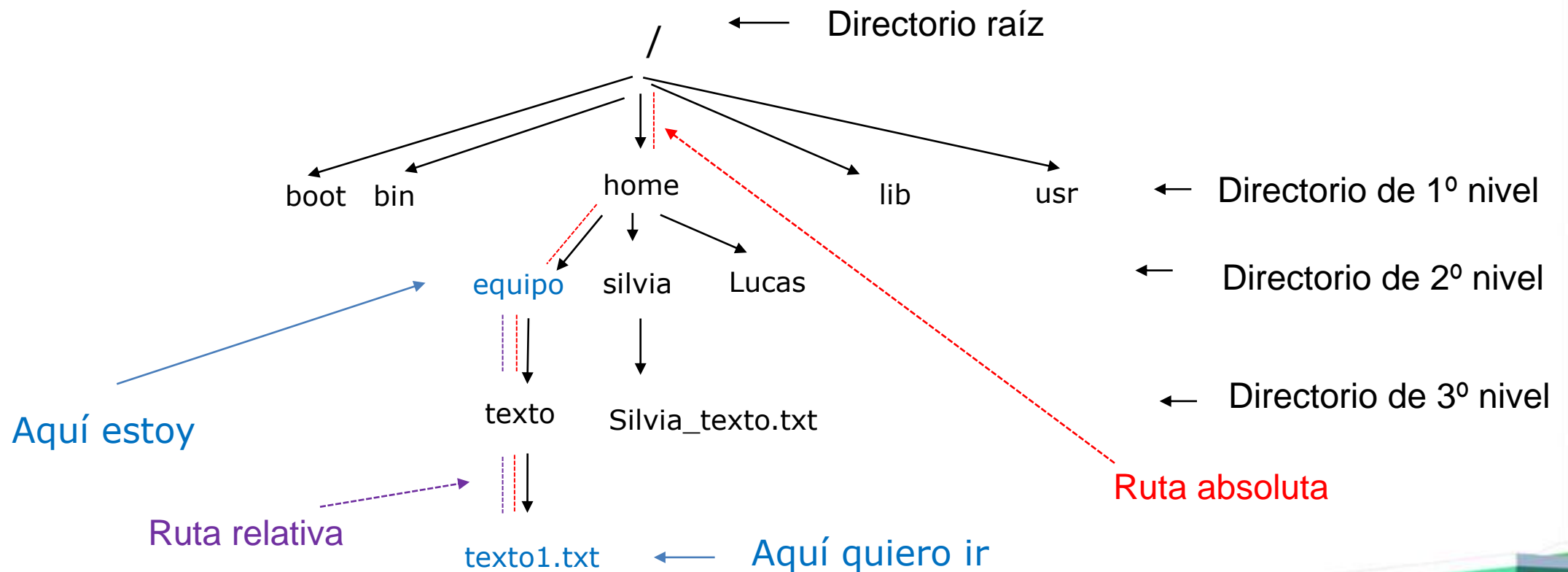
Rutas en Linux

Ruta absoluta:

- ❑ *En la ruta absoluta se indican todos los directorios de la jerarquía desde el root hasta el archivo o directorio final, por ello necesariamente inicia con / .*
- ❑ *El comando pwd muestra una ruta absoluta. Cuando nos movemos de un directorio a otro podemos utilizar una ruta absoluta.*
- ❑ *Las rutas absolutas en general son más extensas que las rutas relativas.*
- ❑ *Ventaja: siempre funcionan, independientemente del lugar del árbol de directorios desde el que se ejecute el comando.*



Sistemas de archivos: jerárquica (árbol invertido)



Ejemplos de archivos y rutas

Ruta (path) absoluta, comienza con el directorio raíz (/):

```
Ls -l /home/equipo/texto/texto1.txt
```



Ruta (path) relativo, comienza en el directorio actual, que no es (/):

```
Ls -l texto/texto1.txt
```



Atributos de un Archivo

Los cuatro atributos básicos de archivo son **nombre**, **tamaño**, **dispositivo y modo**. Nombre es el nombre del archivo, Tamaño es el tamaño del archivo, Dispositivo es un descriptor de un tipo específico de medio de almacenamiento legible (archivo o directorio) y Modo es cómo se accede/utiliza el archivo.



Nombres de Archivos

- Es una secuencia de 1 a 256 caracteres
- Archivo, directorio y dispositivo en el sistema tiene un nombre.
- El nombre del archivo es el único identificador del contenido del directorio.
- Si 2 archivos estén en el mismo directorio, no pueden tener nombres iguales
- Los nombres de archivo pueden ser los mismos si están en diferentes directorios o subdirectorios.



Los nombres de los archivos pueden contener los siguientes caracteres:

- ❖ Letras mayúsculas
- ❖ Letras minúsculas
- ❖ Números
- ❖ Guión bajo “_”
- ❖ Punto
- ❖ Coma



Caracteres de control (funciones especiales) no pueden usarse para nombrar archivos

Comillas dobles “

Signo de interrogación ?

Comilla cerrada ’

Comilla abierta ‘

Asterisco *

Corchetes []

Signo menos -

Prohibido
usar



Directorio raíz (root)

- Como una excepción, el directorio raíz (root) se llama / y se le refiere con este único carácter.
- Ningún otro archivo puede ser llamado de esta manera.
- Los nombres de los archivos pueden contener extensiones que son otro nombre abreviado que va después de un punto.



Shell de Linux

- Es un intérprete de comandos, la cara visible al usuario, interpreta las órdenes que recibe y las transmite al Kernel mediante System Calls.
- Hay varios disponibles:
 - **sh** shell estándar o de Bourne
 - **csh** c-shell
 - **ksh** korn shell
 - **bash** Bourne Again shell



Ejemplos (atributos):

drwxr-xr-x	2	juan	juan	4096	2021-01-16	13:45	GNUstep
-rw-r--r--	1	juan	juan	239618	2021-02-03	17:04	informe-abw
drwxr-xr-x	5	juan	juan	4096	2021-01-23	09:12	temp

Diagram illustrating the components of a file listing (ls command output) with labels and arrows pointing to the corresponding fields:

- Tipo de archivo**: Points to the first column (file type).
- Permiso del propietario**: Points to the first character of the permissions (e.g., 'd' for directory).
- Permiso de grupo**: Points to the second character of the permissions (e.g., 'r' for read).
- Permiso de otros**: Points to the third character of the permissions (e.g., 'x' for execute).
- Enlaces**: Points to the link count (e.g., '2' for two links).
- Propietario**: Points to the owner name (e.g., 'juan').
- otro**: Points to the group name (e.g., 'juan').
- tamaño**: Points to the file size (e.g., '4096').
- fecha**: Points to the modification date (e.g., '2021-01-16').
- hora**: Points to the modification time (e.g., '13:45').
- Nombre**: Points to the file name (e.g., 'GNUstep').

Protección de Archivos

- ❖ En todo sistema multiusuario es necesario establecer niveles de permisos para proteger los archivos y directorios.
- ❖ Los niveles de permisos permiten el acceso de algunos usuarios y bloquean o niegan el acceso a otros.
- ❖ Los archivos son protegidos por permisos de acceso. Existen 3 tipos.
- ❖ Los permisos de acceso se aplican a todos los tipos de archivos (ordinarios, de dispositivos y de directorios).



Permisos de acceso: archivos ordinarios

read : Permite a un usuario ver el contenido del archivo. No se pueden guardar las modificaciones que se hallan hecho editándolo.

write: permite modificar el contenido del archivo o borrarlo.

execute: Si el archivo es un programa, el permiso de ejecución permite correrlo, caso en contrario esto es impedido por el sistema



Permisos de acceso: archivos de directorio

read :Permite ver el contenido del directorio. Esto es: se puede pedir un listado de archivos. Para esto también será necesario poseer el permiso de ejecución.

write: permite crear, modificar el contenido o borrar un archivo de cualquier tipo.

execute: En este caso, si falta no se puede circular por el directorio en cuestión. De esta manera no se puede pedir el listado de archivos, ni utilizarlo como camino de paso hacia otro subdirectorio.

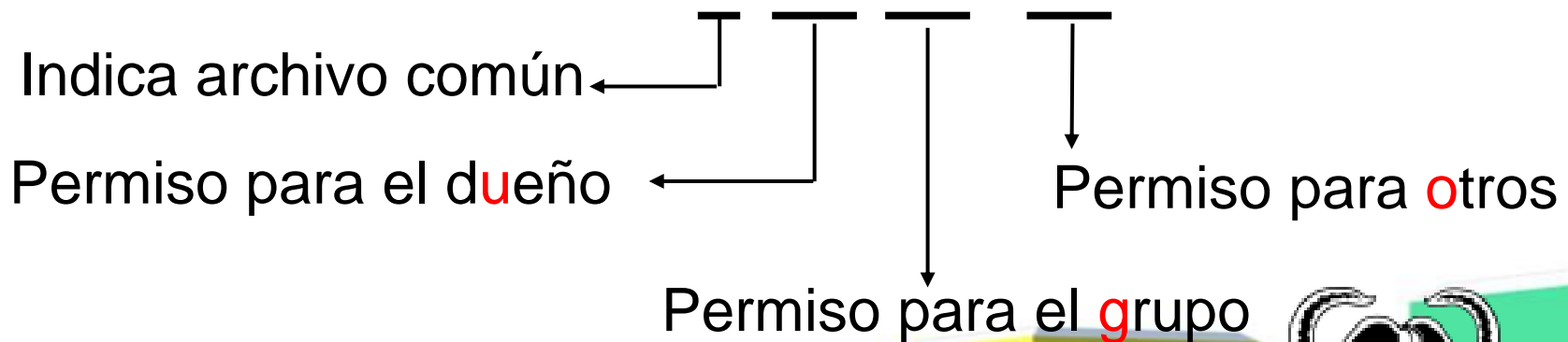


Permisos de acceso

Entidades de un sistema unix:

- U** User El dueño del archivo
- G** Grup Grupo al que pertenece el archivo
- O** Others Los otros (el resto del mundo)

Interpretación del listado: - rwx rwx rwx



Propiedad de los archivos

- ❖ El dueño de un archivo es quien lo creó. La propiedad se puede cambiar con el comando `chown` (change owner).
- ❖ El grupo al que pertenece un archivo se cambia con el comando `chgrp`.
- ❖ En algunos sistemas se reserva el cambio de propiedad de un archivo sólo al administrador y los usuarios comunes pueden cambiarlo de grupo sólo si pertenecen al grupo destino.



Cambio de permiso en archivos

Lo realiza el dueño del archivo o el administrador del sistema mediante el comando `chmod`

Existen dos formas para ello:

- Simbólica
- Numérica



Cambio de permiso en archivos: Forma Numérica

Los tres niveles (rwx) se pueden representar por dígitos octales, que tienen un peso asignado de acuerdo a su posición. Lo mismo se aplica a cada entidad (dueño, grupo, otros), con lo que se forma un número octal de tres dígitos.

rwx 421

4, 2 y 1 son los valores en el sistema de numeración octal que tienen los tres niveles mostrados (rwx), x= 1, w= 2 y r= 4. Si están los tres la suma de ellos da 7.



Realimentación pedagógica



Enunciar:

- ☐ Lo que me pareció difícil
- ☐ Lo que comprendí
- ☐ Lo confuso

BIBLIOGRAFÍA

1) Título: Advanced Linux programming

Autor: mark L. Mitchell, Alex Samuel, Jeffrey Oldham

Editorial: New Riders Publishing. ISBN: 0735710430

Disponible gratuitamente en http:

[/www.advancedlinuxprogramming.com/](http://www.advancedlinuxprogramming.com/)

2) Título: Programación en Linux

Autor: Kurt Wall. Editorial: Prentice Hall

ISBN: 0-7897-2215-1

3) Titulo: Aprendiendo C++ para Linux en 21 Días

Autor: Jesse Liberty y David B. Horvath. Editorial: Prentice Hall. ISBN:

970-26-0012-X Año: 2001G



GRACIAS

