

# Cátedra de Sistemas Operativos

## Unidad 2 – Administración de Archivos

### Parte 1

2024

# Temario – Administración de Archivos

- ARCHIVOS
  - Concepto
  - Nombre de archivos
  - Estructura de archivos
  - Tipos de archivos
  - Métodos de acceso
  - Atributos
  - Operaciones con archivos
- DIRECTORIOS:
  - Jerarquía y rutas de acceso
  - Operaciones con directorios

# Concepto de archivos

- Los **archivos** son unidades lógicas de información creada por los procesos, cuyo tamaño se expresa en Bytes.
- Permiten almacenar gran cantidad de información.
- La información que se almacena en los archivos debe ser **persistente**
- Un archivo se puede compartir entre varios procesos.
- La parte del sistema operativo que trata con los archivos se conoce como **sistema de archivos**.

# Nombre de Archivos

- Se debe asignar un nombre en el momento de crear un archivo.
- La forma de nombrar un archivo depende del sistema operativo.
- Algunos S.O. distinguen mayúsculas de minúsculas.
- Linux ☐ **nombre**
- Windows ☐ **nombre.extensión**

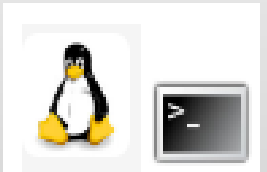
¿Cuál es la función de la “extensión” del nombre de un archivo?



# Nombre de Archivo en Linux

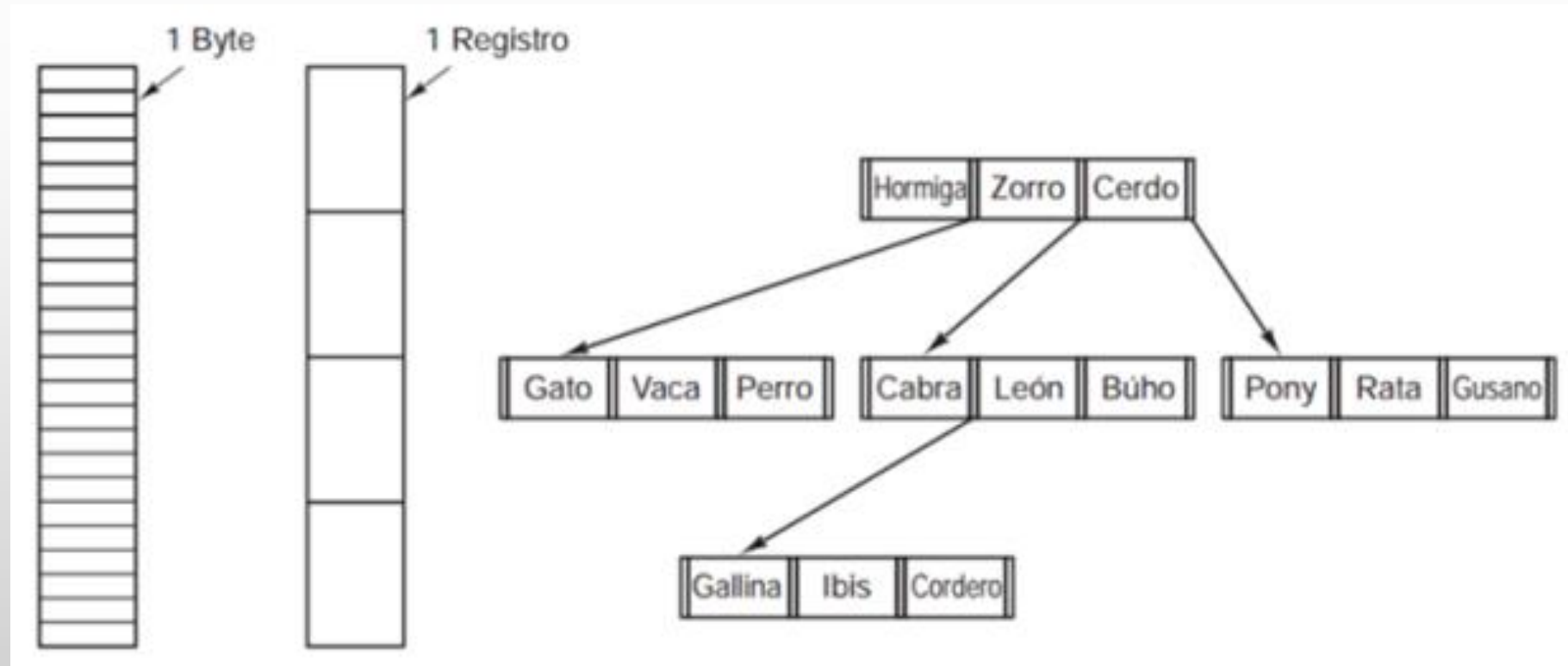


- Puede tener entre 1 y 255 caracteres.
- Se puede utilizar cualquier carácter excepto la barra inclinada / y el carácter nulo.
- No es recomendable emplear los caracteres con significado especial en Linux. En tal caso, el nombre del archivo deberá ir entre comillas.
- Se pueden utilizar números exclusivamente.
- Es sensible a las letras mayúsculas y minúsculas.



# Estructura de archivos

- Secuencia de bytes.
- Secuencia de registros.
- Árbol.



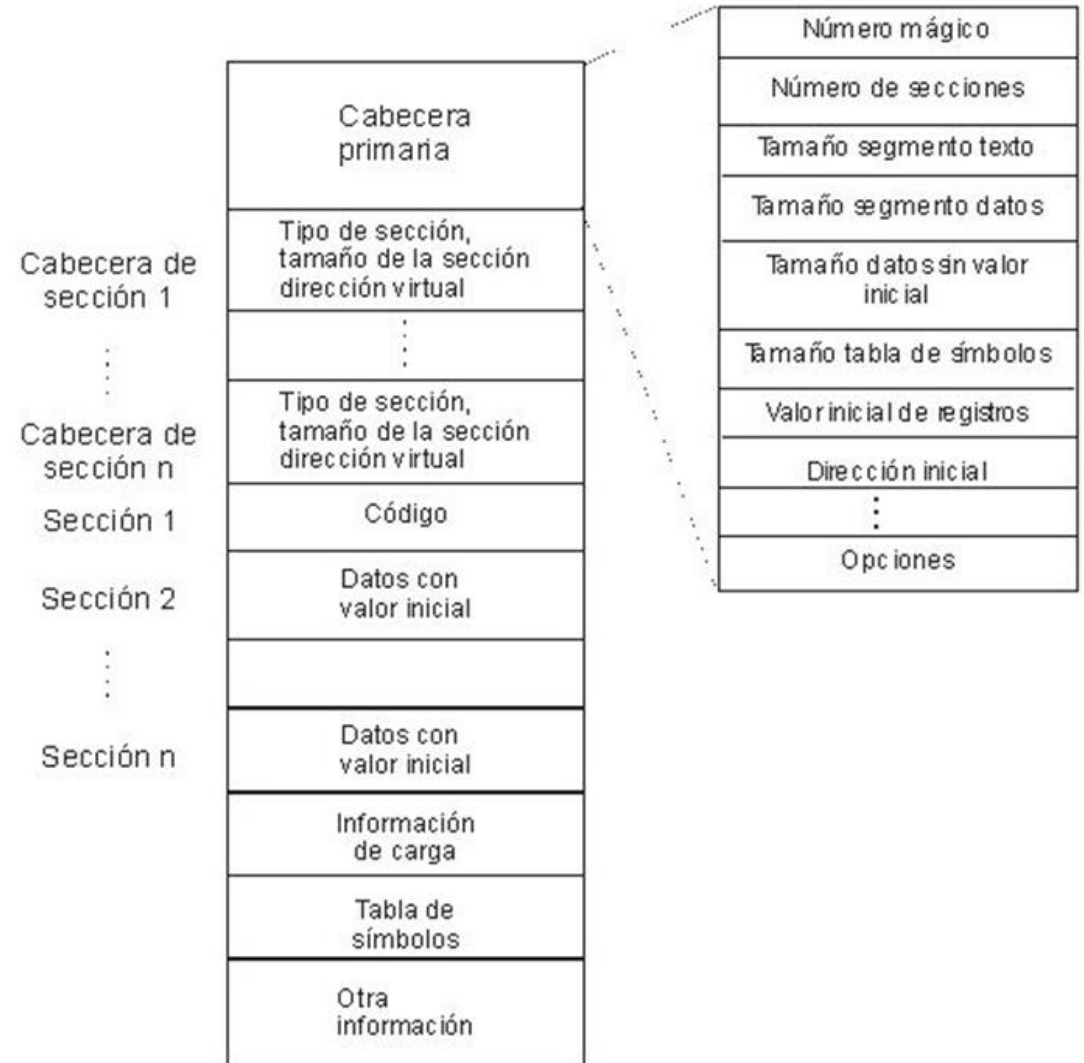
# Tipos de Archivos

1. Archivos regulares o normales.
2. Directorios o carpetas.
3. Archivos especiales de carácter.
4. Archivos especiales de bloques.

# Tipos de Archivos (cont.)

## 1. Archivos regulares:

- Contienen información del usuario.
- Tipos:
  - Archivos ASCII [?] se visualizan con cualquier editor de texto.
  - Archivos binarios [?] dependen de la aplicación que los creó.



Archivo binario ejecutable



# Tipos de Archivos (cont.)

## 2. Directorios o Carpetas:

- Son archivos del sistema utilizados para mantener la estructura del sistema de archivos.
- Contienen archivos y/o directorios.



# Tipos de Archivos (cont.)

## 3. Archivos especiales de carácter:

- Relacionados con la Entrada/Salida.
- Representan dispositivos de E/S en serie.
- Son archivos que modelan impresoras, terminales, interfaces de red, teclados, mouses, etc.



# Tipos de Archivos (cont.)

## 4. Archivos especiales de bloque:

- Relacionados con la Entrada/Salida.
- Representan dispositivos de E/S que manejan bloques de datos.
- Permiten el acceso aleatorio.
- Son archivos que modelan discos, pendrives, etc.



# Tipos de Archivos en Linux



- **Regulares:** Pueden contener cualquier tipo de información (-).
- **Directorios:** Agrupa otros archivos de una forma estructurada (d).
- **Especiales:** Representan los dispositivos (b o c).
- **Enlazados:** Le permiten dar a un único archivo múltiples nombres (l).
- **Tubería:** Utilizados para comunicación entre procesos (p).



```
$ ls -l
total 8
drwxr-xr-x  2 user2      1024 Feb  9 14:22 directory1
-rw-r--r--  1 user2           0 Feb 10 10:20 emptyfile
-rw-r--r--  1 user2    104357 Feb  5 08:20 large-file
drwxr-xr-x  3 user2      1024 Feb 10 11:13 veggies2
```

Permiso	Identifica
-	Archivo.
d	Directorio.
b	Archivo de bloques especiales.
c	Archivo de caracteres especiales.
l	Archivo de vínculo o enlace.
p	Archivo especial de cauce.

# Métodos de Acceso

## **Archivos de acceso secuencial**

- Se accede a los datos en un determinado orden.
- Ejemplo: cintas magnéticas.

## **Archivos de acceso aleatorio**

- Los datos se acceden en cualquier orden.
- Se utiliza la operación SEEK.
- Ejemplo: discos.

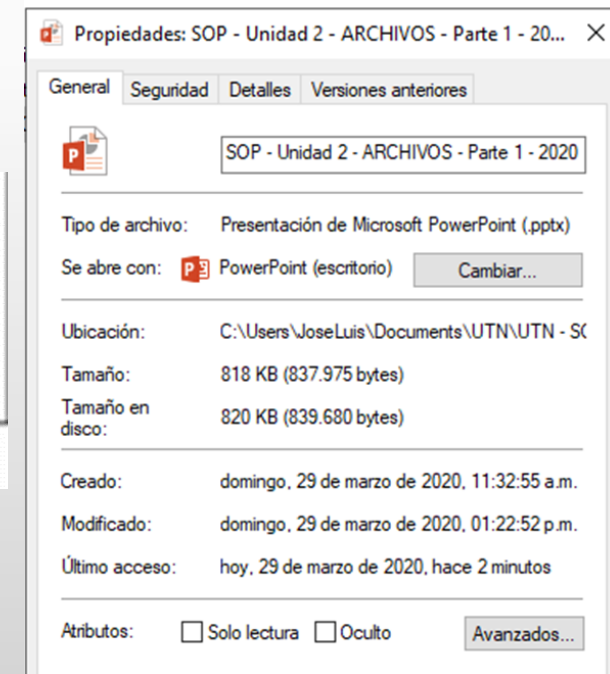
# Atributos de un Archivos

- Un atributo o metadato es información adicional asociada con un archivo (además de su nombre y los datos propiamente dichos).

- Creador
- Propietario
- Protección
- Hora de creación
- Hora del último acceso
- Hora de la última modificación
- Tamaño actual
- Contraseña
- Bandera de solo lectura
- Bandera oculto
- Bandera del sistema
- Bandera ASCII/binario
- Bandera temporal

```
$ ls -l
total 8
drwxr-xr-x  2 user2      1024 Feb  9 14:22 directory1
-rw-r--r--  1 user2         0 Feb 10 10:20 emptyfile
-rw-r--r--  1 user2 104357 Feb  5 08:20 large-file
drwxr-xr-x  3 user2      1024 Feb 10 11:13 veggies2
```

Permiso      Enlaces      Dueño      Tamaño      Fecha      Hora      Nombre arch /dir

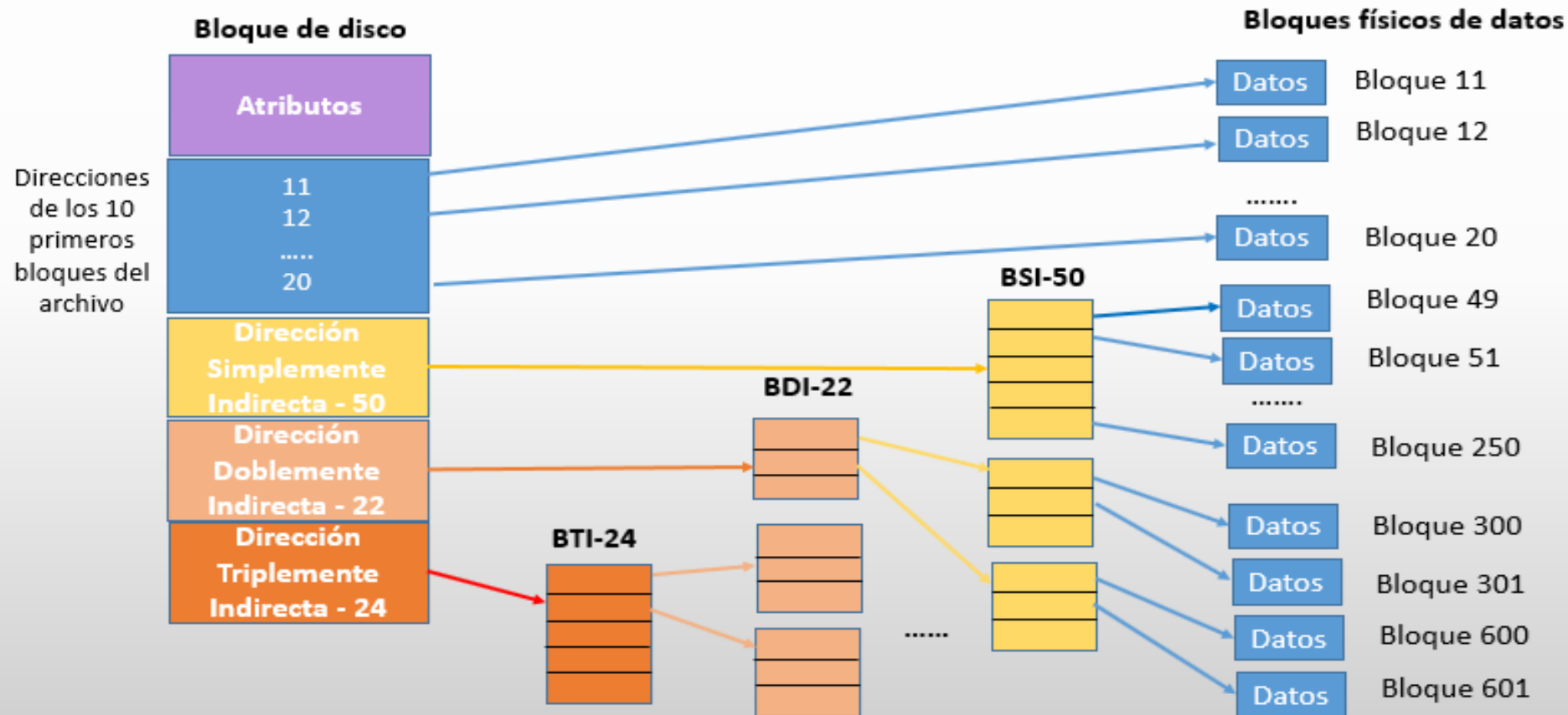


# Nodo-i

- Es una tabla que contiene información referida a un archivo.
- Cada archivo tiene su propio nodo i.
- Contenido del nodo i:
  - Atributos.
  - Direcciones de los bloques que almacenan los datos (son los datos que el usuario guarda en el archivo, por ejemplo: una foto, un apunte, un listado de notas).
  - Direcciones indirectas:
    - Dirección simplemente indirecta.
    - Dirección doblemente indirecta.
    - Dirección triplemente indirecta.

# Atributos de un Archivos en Linux

## Nodo-i o i-nodo (nodo índice)





# Atributos de un Archivos en Linux

**Atributos → Información referida al archivo:**

**Tipo de archivo**

**Permisos de acceso**

**Número de enlaces**

**Dueño del archivo**

**Grupo**

**Tamaño**

**Fecha de creación**

**Fecha de último acceso**

**Fecha de última modificación**

**El nombre del archivo NO se encuentra almacenado en el nodo i**

En Linux un Atributo es toda información referida al archivo y que se encuentra almacenada en el nodo i del archivo.

# Operaciones con Archivos

1. Open
2. Close
3. Create
4. Delete
5. Read
6. Write
7. Append
8. Seek
9. Get attributes
10. Set attributes
11. Rename

# Operaciones con Archivos

1. **Open** `?` permite que el sistema lleve los atributos y la lista de direcciones de bloques de disco del archivo a memoria. Un archivo debe abrirse ANTES de poder leerlo

`fd` `?` file descriptor

`fd = open (nombre_archivo, permisos)`

2. **Close** `?` libera espacio en tablas internas

`close (fd)`

# Operaciones con Archivos (continuación)

3. **Create**  se crea un archivo sin datos y se definen algunos de sus atributos (fecha de creación, propietario, permisos, etc.)

`fd = create (nombre_archivo, permisos)`

Ejemplo `$ touch archivo`

4. **Delete**  elimina el archivo del disco para liberar espacio

`delete (fd)`

Ejemplo `$ rm archivo`

# Operaciones con Archivos (continuación)

**5. Read**  lee los datos del archivo y los lleva a un buffer en RAM

`cantidad_bytes_leídos = read (fd, dirección_buffer, cantidad_datos_a_leer)`

Ejemplo `$ cat archivo`

**6. Write**  escribe el archivo desde un buffer en RAM en el disco

`cantidad_bytes_escritos = write (fd, dirección_buffer, cantidad_datos_a_escribir)`

Ejemplo `$ echo "nuevo contenido" > archivo1`

`$ ls /bin > archivo2`

# Operaciones con Archivos (continuación)

7. **Append** [?] se agregan datos al final del archivo

Ejemplo \$ date >> archivo

8. **Seek** [?] coloca el puntero del archivo en una posición específica

9. **Get attributes** [?] permite obtener los atributos de un archivo

Ejemplo: \$ ls -l

10. **Set attributes** [?] permite modificar los atributos de un archivo

Ejemplo: \$ chmod 777 archivo

11. **Rename** [?] permite modificar el nombre de un archivo

Ejemplo: \$ mv nom\_viejo nome\_nuevo

# Comandos en Linux

- Crear archivos : touch, nano, vi, redireccionamiento, mkdir, cp
- Movernos en la estructura: cd
- Visualizar contenido: ls, more, tail, head, less
  - Con Filtros: grep, sort, cut, find
  - Enlazar: ln
  - Renombrar: mv
  - Eliminar: rmdir, rm



# Qué es un archivo?



Es una secuencia de bytes identificado con un número i-nodo que permite su localización en disco.

Permiso	Identifica
-	Archivo.
d	Directorio.
b	Archivo de bloques especiales.
c	Archivo de caracteres especiales.
l	Archivo de vinculo o enlace.
p	Archivo especial de cauce.



# I-nodo



Un inodo es una estructura de datos que mantiene un registro de todos los archivos y directorios dentro de un sistema de archivos.

Metadatos que contiene el i-nodo

Tipo de archivo

Tamaño de archivo

ID del propietario

ID del grupo

Permisos de lectura, escritura y ejecución

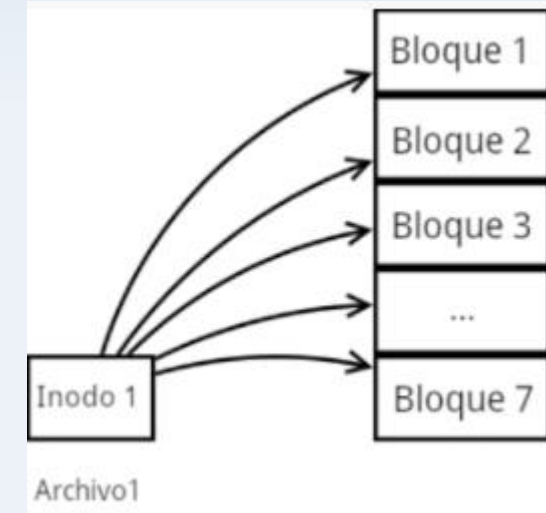
Marcas de tiempo fecha de última modificación (mtime), acceso (atime) y de alteración del propio i-nodo (ctime).

Nro de enlaces

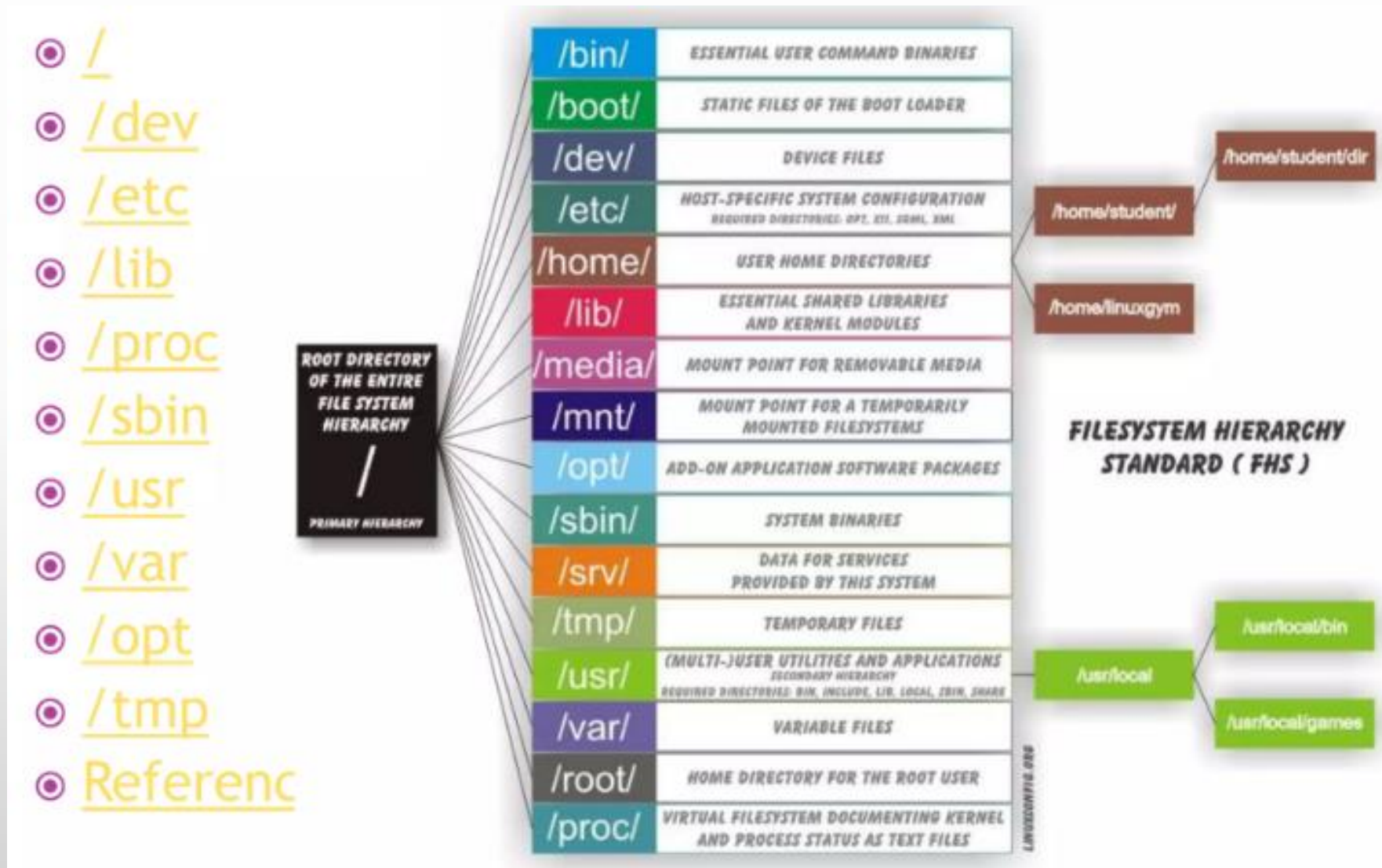
Punteros a bloques Son números de 32 bits que permiten localizar un bloque físico del disco

Asignado al archivo

Usando el comando stat podemos consultar la información que un inodo guarda sobre un archivo o directorio determinado. Si ejecutamos el comando stat seguido del nombre de un archivo, directorio o enlace obtendremos el siguiente resultado:



# Organización de Directorios



# Organización de directorios



La partición raíz / siempre debe contener físicamente las particiones /etc,/bin, /sbin, /lib y /dev, para que el sistema pueda arrancar correctamente.

## ¿Cómo podemos ver un listado de los archivos y directorios?

ls

Algunas Opciones:

- a
- l
- h
- 1
- i
- R

```
juli@juli-VirtualBox:~$ ls
a1      Documentos  Escritorio  parcial.sh  Reporte
a2      enlace     Imágenes   Plantillas  ReporteDiario
Descargas enlace2     Música     Público     script.sh
```

# Comando ls -l

Comando ls -l



```
clarisa@DESKTOP-D79LATI:~$ ls -l ./d1
total 0
-rw-r--r-- 1 clarisa clarisa  4 Apr  1 12:46 a3
-rw-r--r-- 1 clarisa clarisa  8 Apr  1 12:56 a4
drwxr-xr-x 1 clarisa clarisa 4096 Apr  1 12:31 d3
clarisa@DESKTOP-D79LATI:~$
```



Primer campo:

Primer carácter: Tipo de archivo

- Archivo regular   d Directorio   b Archivo especial de bloque c Archivo especial de caracter

Segundo al décimo caracteres: Permisos de acceso al archivo

Niveles de usuarios:

Dueño   Grupo   Otros

Permisos:

r: lectura   w: escritura   x: ejecución

Dueño   Grupo   Otros

rw-r--r--



# Comando ls -l (cont.)

Comando ls -l



```
clarisa@DESKTOP-D79LATI:~$ ls -l ./d1
total 0
-rw-r--r-- 1 clarisa clarisa  4 Apr  1 12:46 a3
-rw-r--r-- 1 clarisa clarisa  8 Apr  1 12:56 a4
drwxr-xr-x 1 clarisa clarisa 4096 Apr  1 12:31 d3
clarisa@DESKTOP-D79LATI:~$
```



**Segundo campo: Cantidad de enlaces**

**Tercer campo: Dueño del archivo**

**Cuarto campo: Grupo al cual pertenece**

**Quinto campo: Tamaño en bytes del archivo**

**Siguientes campos: Fecha y hora de última modificación**

**Último campo: Nombre del archivo**



# Directorio de conexión ~



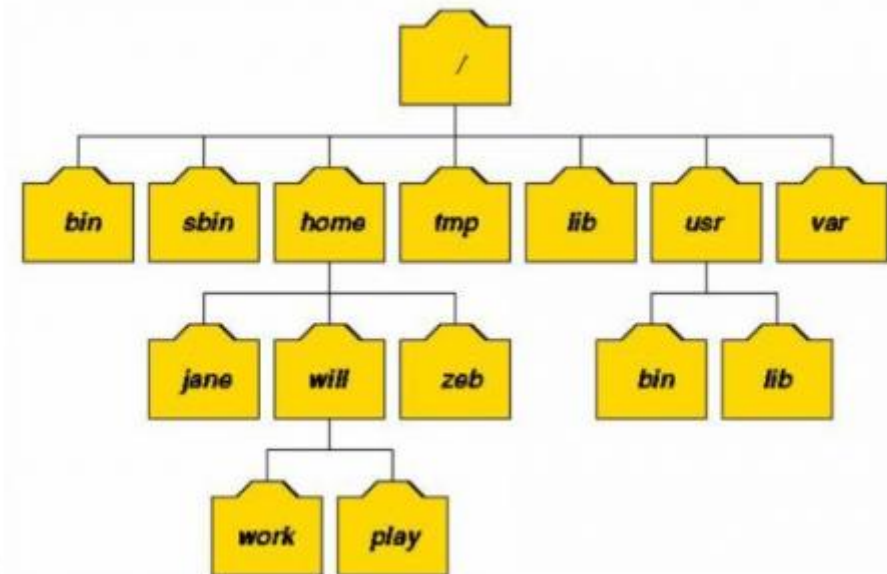
## Home directory

Debajo del directorio Home se crea por defecto un directorio para cada usuario que posee el SO. Cuando un usuario inicia sesión se ubica en este directorio para trabajar, permitiendo moverse por el resto del sistema de archivo según los permisos otorgados.

Existe una variable donde se guarda el directorio de conexión es la variable HOME.

Existe una por cada usuario

```
juli@juli-VirtualBox:~$ echo $HOME  
/home/juli
```



# Directorio de trabajo .



Es el directorio en el que estoy ubicado actualmente trabajando.

## ¿Cómo puedo ver el directorio activo?

Con el comando **pwd** o con la variable **\$PWD**

**pwd** muestra la ruta absoluta al directorio activo (de trabajo).

Comando

```
juli@juli-VirtualBox:~$ pwd
/home/juli
juli@juli-VirtualBox:~$ cd Música/
juli@juli-VirtualBox:~/Música$ pwd
/home/juli/Música
```

Valor de la variable

```
juli@juli-VirtualBox:~$ echo $PWD
/home/juli
juli@juli-VirtualBox:~$ cd Música/
juli@juli-VirtualBox:~/Música$ echo $PWD
/home/juli/Música
juli@juli-VirtualBox:~/Música$
```



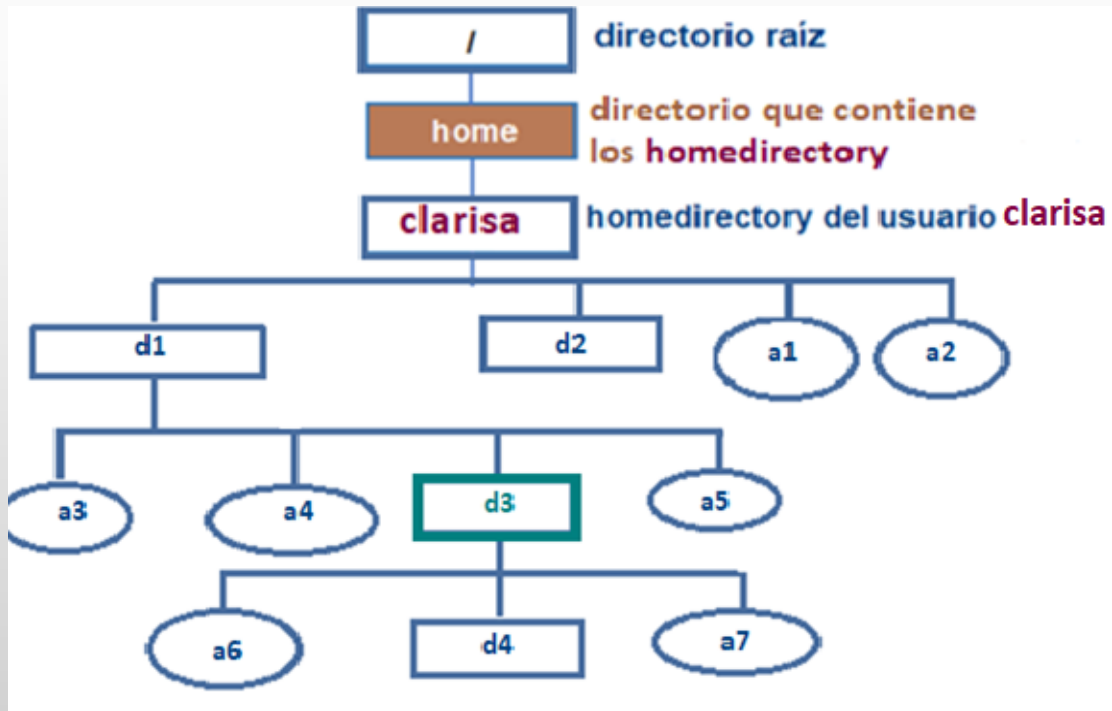


## ¿Cómo moverse del directorio activo a otro directorio?

Con el comando **cd**

**cd** (change directory) este comando permite cambiarse de un directorio a otro

**cd directorioAMoverse**



**cd** /home/clarisa/d1/d3

El **homedirectory** dejó de ser el directorio actual.

Ahora el **directorio actual** es **d3**



# Trayectoria o rutas de acceso



Una ruta de acceso es el camino que se debe recorrer para encontrar la ubicación de un archivo.

## Tipos de Rutas

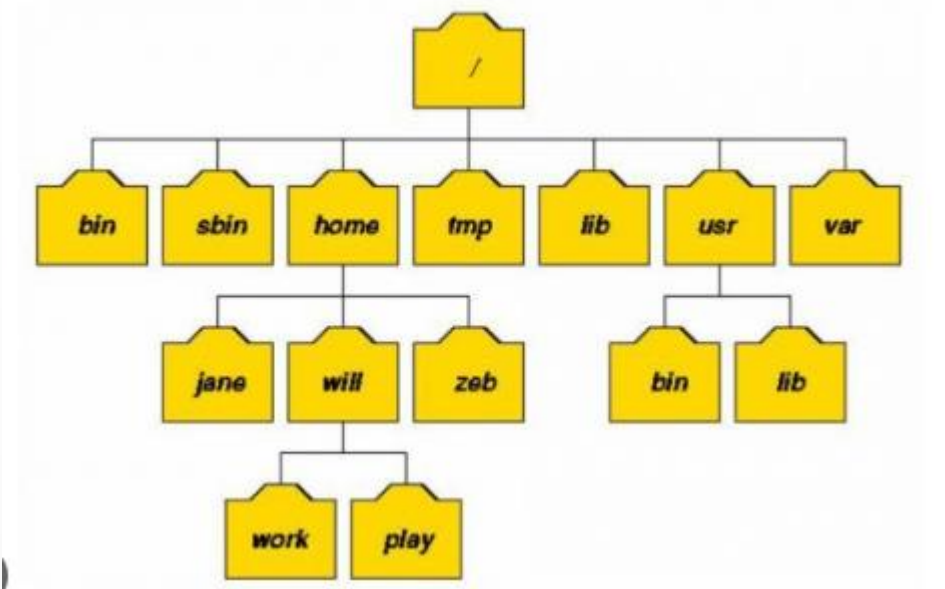
### Rutas Absolutas

Comienza siempre con el Dir. raíz / y continúan todos los directorios que hay que recorrer para llegar al archivo deseado. Entre cada componente de la ruta se debe colocar /  
Ejemplo  
/home/juli/Documentos/notas

### Rutas Relativas

Utiliza el directorio actual de trabajo para comenzar la búsqueda.  
. Hace referencia al directorio actual  
.. Hace referencia al directorio padre  
Entre cada componente de la ruta se debe colocar / Ejemplo ../Documentos/notas

# Ejercicios de rutas de acceso



1. Si su directorio de trabajo es `/usr/lib`, escriba la ruta absoluta para llegar al directorio `play`
2. Continuando en el mismo directorio activo, escriba la ruta relativa para llegar a `work`.
3. Si su directorio de trabajo es `jane`, escriba la ruta relativa y absoluta para llegar `tmp`.

# A repasar

<https://arepasar.com.ar/app/unirse/?id-cuestionario=VHX-SGCC-IYH>

Código de acceso VHX-SGCC-IYH



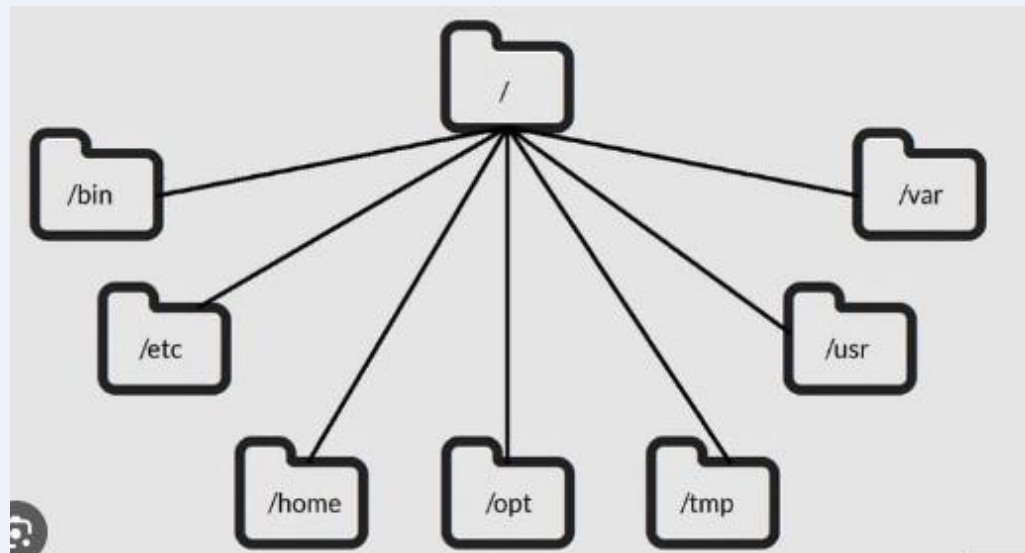
# Sistema de Archivos



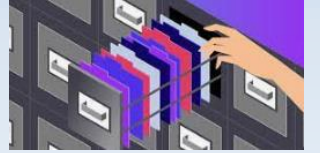
El sistema de archivos es la parte del sistema operativo que se encarga de administrar los archivos y directorios de un dispositivo.

Lo que hace el sistema de archivos es almacenar de forma estructurada toda esta información para facilitar su búsqueda, edición y eliminación.

Cada sistema de archivos pertenece a una clase o tipo que define la característica administrativa. Así como algunas versiones de Windows usan FAT o NTFS, Linux en particular adopta el sistema de archivo Ext4.



# Crear, ver y eliminar archivos



## ¿Cómo crear archivos ordinarios o regulares?

- ✓ Con editores de texto.
  - vi, nano, mcedit, gedit, etc.
- ✓ Con concatenación
  - > o >>
- ✓ Con comando touch
  - touch nuevoArchivo**
- ✓ Con comando **cat** y redireccionamiento
  - cat archivo >> nuevoArchivo**

A screenshot of a terminal window titled 'juli@juli-VirtualBox: ~'. The terminal shows the command 'gedit archivo1' being executed. A window titled '\*archivo1' is open, displaying the text '1 nuevo archivo de sop'.

```
juli@juli-VirtualBox:~$ ls >> archivo2
```

```
juli@juli-VirtualBox:~$ touch archivo3
```

```
juli@juli-VirtualBox:~$ cat archivo2 > archivo4  
juli@juli-VirtualBox:~$ cat archivo3 >> archivo4
```

# Crear, ver y eliminar archivos



¿Cómo ver el contenido de un archivo?

**cat nombreArchivo**  
**more nombreArchivo**  
**less nombreArchivo**

```
juli@juli-VirtualBox:~$ cat archivo2
archivo2
Descargas
Documentos
Escritorio
Imágenes
Música
NotasSOP
Plantillas
Público
Vídeos
```

¿Cómo ver las primeras líneas o las ultima líneas de un archivo?

**head [opciones] nombreArchivo**  
**tail [opciones] nombreArchivo**

¿Cómo eliminar un archivo?

**rm nombreArchivo**

```
juli@juli-VirtualBox:~$ rm archivo1
```



# Más detalle del comando more, tail y head

- ❑ `more -n directorio/nombre de archivo` (de a n líneas por vez)
- ❑ `more +n directorio/nombre de archivo` (a partir de la línea n en forma paginada)
- ❑ `tail -n ruta directorio/nombre de archivo`  
Muestra las últimas n líneas
- ❑ `head -n ruta directorio/nombre de archivo`  
Muestra las primeras n líneas



# Más detalle del comando more, tail y head (cont.)

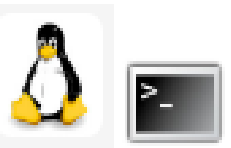


`more -n directorio/nombre de archivo (de a n líneas por vez)`

❑ `more +n directorio/nombre de archivo (a partir de la línea n en forma paginada)`

Ejemplo: Visualizar el contenido del archivo comandos de a 3 líneas por vez y a partir de la línea 5:

`$ more -3 +5 ./comandos` o `more -3 +5 comandos`

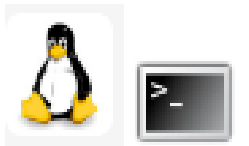




Se obtiene la misma salida cualquiera sea el orden en el cual se ingresan las opciones:

```
clarisa@DESKTOP-D79LATI:~$ more +5 -3 comandos  
aa-enabled  
aa-exec  
add-apt-repository  
--More-- (0%)
```

```
clarisa@DESKTOP-D79LATI:~$ more -3 +5 comandos  
aa-enabled  
aa-exec  
add-apt-repository  
--More-- (0%)
```

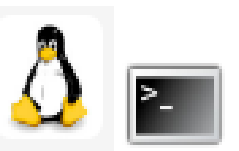


❑ **tail -n** ruta directorio/nombre de archivo

Muestra las **últimas n líneas**

Ejemplo:

```
clarisa@DESKTOP-D79LATI:~$ tail -5 comandos
zgrep
zipdetails
zless
zmore
znew
```



**head -n ruta directorio/nombre de archivo**  
**Muestra las primeras n líneas**

**Ejemplo:**  
**Visualizar las primeras 6 líneas del archivo comandos**

```
clarisa@DESKTOP-D79LATI:~$ head -6 comandos
NF
VGAAuthService
X11
[
aa-enabled
aa-exec
clarisa@DESKTOP-D79LATI:~$
```



# Crear, ver y eliminar directorios



¿Cómo crear directorios?

`mkdir nombreDirectorio`

`mkdir -p directorio/subdirectorio/subdirectorio2`

└─ opcion `-p` crea los directorios padres si no existen

```
juli@juli-VirtualBox:~$ mkdir dir1
juli@juli-VirtualBox:~$ mkdir -p dir2/subDir
juli@juli-VirtualBox:~$ ls dir2/
subDir
```

# Crear, ver y eliminar directorios



## ¿Cómo ver el contenido de un directorio?

**ls [opciones] nombreDirectorio**

Algunas opciones

-i  
-l  
-a  
-R

## ¿Cómo eliminar un directorio?

o Si el directorio está vacío

**rmdir nombreDirectorio**

o Si el directorio NO está vacío

**rm -r nombreDirectorio**

```
juli@juli-VirtualBox:~$ ls -R dir2
dir2:
subDir

dir2/subDir:
archivo
juli@juli-VirtualBox:~$ ls -l
total 52
-rw-rw-r-- 1 juli juli  96 mar 16 08:33 archivo2
-rw-rw-r-- 1 juli juli   0 mar 16 08:34 archivo3
-rw-rw-r-- 1 juli juli  96 mar 16 08:35 archivo4
drwxr-xr-x 2 juli juli 4096 mar 22  2022 Descargas
drwxrwxr-x 2 juli juli 4096 mar 16 08:38 dir1
drwxrwxr-x 3 juli juli 4096 mar 16 08:38 dir2
drwxr-xr-x 2 juli juli 4096 mar 22  2022 Documentos
drwxr-xr-x 3 juli juli 4096 mar 22  2022 Escritorio
drwxr-xr-x 2 juli juli 4096 mar 22  2022 Imágenes
```

# Copiar archivos

El comando para copiar archivos es **cp**

- o Copiar un archivo a otro

**cp archivo copiaArchivo**

- o Copiar archivos a un directorio

**cp archivo-1 archivo-2 directoriDestino**

- o Copiar en forma recursiva un directorio completo:

**cp -r direc-origen direc-destino**

cp crea un archivo nuevo por ende tiene un inodo diferente



# Más información del comando cp



- ✓ Inicialmente el archivo y su copia tienen el mismo contenido pero el contenido del archivo original y el contenido de la copia tienen diferente ubicación debido a que ocupan distintos bloques del disco.
- ✓ Cualquier modificación realizada en el archivo original no afecta a la copia y cualquier modificación realizada en la copia no afecta al archivo original.

❑ `$ cp ruta_direct/nombre_archivo ruta_direct_destino/nombre_copia`

**La copia puede ser creada en el mismo directorio o en otro directorio.**

Si es creada en el mismo directorio, entonces debe tener distinto nombre.

Si es creada en otro directorio, entonces puede tener el mismo nombre.

Si omitimos el nombre de la copia, es creada con el mismo nombre del archivo original.

Si omitimos el directorio destino, por defecto la copia es creada en el directorio actual.

## Ejemplo comando cp

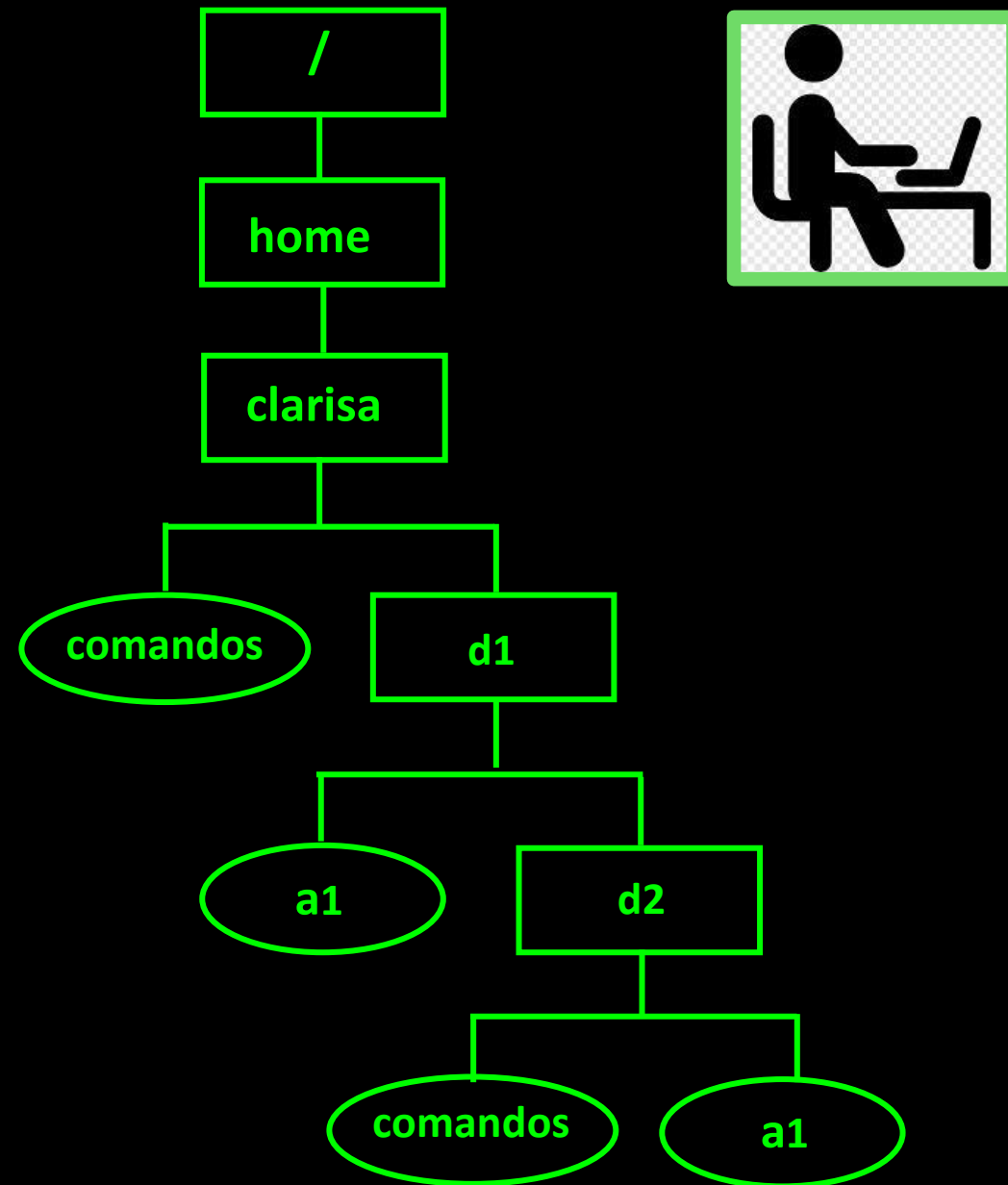
Si creamos las copias de dos o más archivos en un mismo directorio y cada copia debe crearse con el mismo nombre de los archivos originales, entonces la resolución es posible en un mismo paso (en una misma línea de comandos)

Ejemplo: Crear una copia de a1 y comandos en el directorio d2 con el mismo nombre:

```
$cp ./comandos ./d1/a1 ./comandos d1/d2
```

Comprobamos que fueron creadas las copias de a1 y comandos en el directorio d2:

```
$ls ./d1/d2  
a1 comandos
```







# Mover o renombrar archivos

El comando para mover un archivo de un lugar a otro es **mv** tambien sirve para renombrar un archivo de un nombre a otro

- o **Mover** un archivo

**mv archivo rutaAdondeSeQuiereMover**

Ruta absoluta

o

Ruta relativa

- o **Renomabrar** un archivo:

**mv archivo nuevoNombre**

mv mueve o renombra un archivo nuevo por ende tiene el mismo inodo porque sigue siendo el mismo archivo

```
juli@juli-VirtualBox:~$ ls -l
archivo2
juli@juli-VirtualBox:~$ mv archivo2 dir1/
juli@juli-VirtualBox:~$ ls -l
juli@juli-VirtualBox:~$ ls dir1/
archivo2
```

# Ejemplo mv (sintaxis)



## Para trasladar archivos:

### Sintaxis:

`$ mv ruta_directorio/nombre_de_archivo ruta_directorio_destino/nombre` con el cual será trasladado

Si en el segundo argumento omitimos el directorio destino, por defecto será trasladado al directorio actual.

Si en el segundo argumento omitimos el nombre del archivo, será traslado con el mismo nombre.

## Para renombrar archivos:

### Sintaxis:

`$ mv ruta directorio/nombre de archivo misma ruta/nuevo nombre`



# Ejercicios



1. Si su directorio de trabajo es `/usr/lib`, escriba la ruta absoluta para llegar al directorio `play`.
2. Continuando en el mismo directorio activo, escriba la ruta relativa para llegar a `work`.
3. Si su directorio de trabajo es `jane`, escriba la ruta relativa y absoluta para llegar `tmp`.

# Filtros - Comando grep



Localiza en un texto las líneas que contienen determinados caracteres.

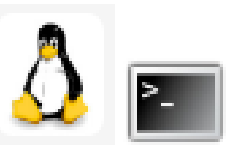
Sintaxis en archivos regulares:

`$grep "carácter" ruta/directorio/nombre del archivo`

Ejemplo:

Listar las líneas que contienen la palabra martes en el texto almacenado en el archivo semana.

`$ grep "martes" d1/semana`



# Filtros - Comando grep (cont.)



También se utilizar para localizar en un texto las líneas que **comienzan** determinados caracteres.

Sintaxis en archivos regulares:

```
$grep '^caracter' ruta/directorio/nombre del archivo
```

Ejemplo 1:

Listar las líneas que **comienzan** con la letra m en el texto almacenado en el archivo semana.

```
$ grep '^m' d1/semana
```



# Filtros - Comando grep (cont.)



`$grep '^caracter' ruta directorio/nombre del archivo`

**Ejemplo 2:**

**Listar las líneas cuyo cuarto carácter sea la letra r en el texto almacenado en el archivo semana.**



# Filtros - Comando grep (cont.)



Resolución: Los tres primeros pueden ser cualquier carácter.

Dentro de grep el punto (.) equivale a un caracter cualquiera:

```
$ grep '^...r' d1/semana
```



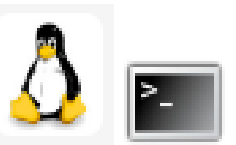
# Filtros - Comando grep (cont.)



Listar los comandos cuyo nombre comienza con la letra l  
No podemos ejecutar: `$ grep '^l' /bin` porque esta sintaxis corresponde a archivos regulares.

```
clarisa@DESKTOP-D79LATI:~$ grep '^-' /bin
grep: /bin: Is a directory
```

¿Podemos ejecutar el comando grep para localizar líneas cuando el texto es el listado de un directorio?  
Respuesta: Si, para que sea posible debemos recurrir a la siguiente técnica:





# Pipes o tuberías



Es una técnica por la cual la salida que emite un comando pasa como entrada del siguiente comando en lugar de ser visualizada por pantalla.

Cada comando se ejecuta sobre la respuesta que genera el comando anterior.

Cada comando toma como argumento la respuesta que genera el comando anterior.

**Esta técnica permite ejecutar ciertos comandos sobre el listado de un directorio**



# Pipes o tuberías



**Sintaxis: comando 1 | comando 2**

**Para ejecutar ciertos comandos sobre el listado de un directorio, generamos el listado y lo enviamos como entrada del siguiente comando, de esta manera el siguiente comando se ejecuta sobre el listado del directorio**

**¿Cuál es la sintaxis?**



# Pipes o tuberías

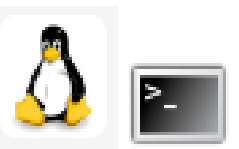


**Sintaxis:** ls opciones directorio | comando

El listado que generamos al ejecutar ls pasa como entrada del siguiente comando

El comando se ejecuta sobre el listado que genera el comando ls.

Ahora si podemos listar los comandos cuyo nombre comienza con la letra l:



# Pipes o tuberías



```
ls /bin | grep '^|'
```



# Comando sort



Visualiza en forma ordenada un texto:

```
$sort -knúmero_de_campo -n rutadirectorio/archivo_regular
```

Visualiza en forma ordenada un texto de acuerdo al campo n

Si el campo es literal ordena de a-z

Si el campo es numérico ordena en forma ascendente

Si agregamos la opción -r el orden se invierte:

Si el campo es alfabético el orden es alfabético inverso (z-a)

Si el campo es numérico el orden inverso es descendente

Si el campo es numérico agregamos la opción -n



# Comando sort - Ejemplos



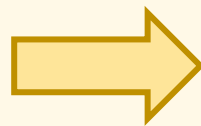
Visualizar el archivo legajo ordenado por cargo:

```
$sort -k3 d1/legajo
```

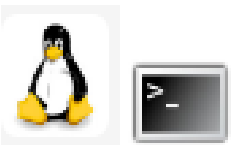
Obtener un listado ordenado por legajo en forma decreciente:

```
$sort -k2 -n -r d1/legajo o $sort -k2 -nr d1/legajo
```

Contenido del  
archivo legajo



```
alvarez,juan 3450 lider 10  
suarez,pedro 2340 operador 2  
cabana,lara 3523 gerente 8
```



# Comando sort - Ejemplos



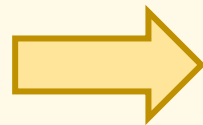
Obtener un listado con los **dos** empleados de mayor **antigüedad**:

```
$sort -k2n d1/legajo | head -2
```

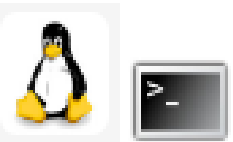
 (Son los dos primeros empleados del listado)

Aquellos empleados de menor son los de mayor antigüedad

Contenido del  
archivo legajo



```
alvarez,juan 3450 lider 10  
suarez,pedro 2340 operador 2  
cabana,lara 3523 gerente 8
```



# Comando sort - Ejemplos



```
clarisa@DESKTOP-D79LATI:~$ sort -k2n d1/legajo|head -2  
suarez,pedro 2340 operador 2  
alvarez,juan 3450 lider 10
```

```
clarisa@DESKTOP-D79LATI:~$ sort -k2nr d1/legajo|tail -2  
alvarez,juan 3450 lider 10  
suarez,pedro 2340 operador 2
```

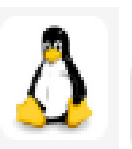






Listar el directorio actual ordenado en forma descendente de acuerdo al tamaño de los archivos

```
clarisa@DESKTOP-D79LATI:~$ ls -l | sort -k5nr
drwxr-xr-x 1 clarisa clarisa 4096 Oct 17 20:26 d1
-rw-r--r-- 1 clarisa clarisa 1188 Oct 9 11:52 daily-report
-rw-r--r-- 1 clarisa clarisa 866 Oct 9 20:49 memory-history
-rw-r--r-- 1 clarisa clarisa 675 Oct 2 22:28 reporte3
-rw-r--r-- 1 clarisa clarisa 598 Oct 2 20:26 reporte
-rw-r--r-- 1 clarisa clarisa 569 Oct 11 10:05 prog10
-rw-r--r-- 1 clarisa clarisa 567 Oct 2 20:41 reporte2
-rw-r--r-- 1 clarisa clarisa 418 Oct 7 17:06 repo
-rw-r--r-- 1 clarisa clarisa 261 Oct 9 12:04 ReporteDiario
-rw-r--r-- 1 clarisa clarisa 164 Oct 11 09:39 par
-rw-r--r-- 1 clarisa clarisa 143 Oct 17 20:32 consigna2
-rw-r--r-- 1 clarisa clarisa 140 Oct 7 17:15 horasmincop
-rw-r--r-- 1 clarisa clarisa 131 Oct 7 10:03 prog222
-rw-r--r-- 1 clarisa clarisa 127 Oct 9 12:59 me
-rw-r--r-- 1 clarisa clarisa 104 Oct 7 15:10 horasmin
-rw-r--r-- 1 clarisa clarisa 75 Oct 10 20:19 consigna3
-rw-r--r-- 1 clarisa clarisa 72 Oct 10 20:37 consigna6
-rw-r--r-- 1 clarisa clarisa 69 Oct 10 20:23 consigna4
-rw-r--r-- 1 clarisa clarisa 67 Oct 17 20:18 comando10
-rw-r--r-- 1 clarisa clarisa 60 Oct 17 20:08 consigna1
-rw-r--r-- 1 clarisa clarisa 42 Oct 10 20:26 consigna5
-rw-r--r-- 1 clarisa clarisa 38 Oct 17 20:06 semana
-rw-r--r-- 1 clarisa clarisa 4 Oct 3 20:38 comandos
-rw-r--r-- 1 clarisa clarisa 3 Oct 10 20:06 nu20
total 12
```



# Comando cut



Genera un listado que contiene determinadas columnas

Sintaxis:

`$cut -d' ' -fn°_campo,n°_campo ruta directorio/nombre del archivo`

Ejemplo:

Obtener un listado que contenga solo cargo y antigüedad de los empleados

`$cut -d' ' -f3,4 d1/legajo`

Obtener un listado que contenga apellido y nombre, y cargo de los dos empleados menor antigüedad:

`$sort -k4 d1/legajo`

```
alvarez,juan 3450 lider 10
suarez,pedro 2340 operador 2
cabana,lara 3523 gerente 8
```



# Comando cut - Ejemplos



```
clarisa@DESKTOP-D79LATI:~$ sort -k4 -nr d1/legajo|tail -2|cut -d' ' -f1,3
cabana,lara gerente
suarez,pedro operador
clarisa@DESKTOP-D79LATI:~$ sort -k4 -n d1/legajo| head -2|cut -d' ' -f1,3
suarez,pedro operador
cabana,lara gerente
clarisa@DESKTOP-D79LATI:~$
```



# Comando find



Busca archivos de acuerdo a un determinado patrón de búsqueda:

**\$find** directorio patrón de búsqueda

directorio: es el directorio a partir del cual se realizará la búsqueda

Patrón de búsqueda:

-type: por tipo de archivo

-type f: archivo regular

-type d: directorio



# Comando find



Busca archivos de acuerdo a un determinado patrón de búsqueda:

**Sintaxis:**

**\$find** directorio patrón de búsqueda

directorio: es el directorio a partir del cual se realizará la búsqueda

**Patrón de búsqueda:**

**-name nombre\_de\_archivo:** por nombre

**-inum nro\_de\_nodo i:** por número de nodo i (busca los enlaces de un archivo)



# Comando find



Busca archivos de acuerdo a un determinado patrón de búsqueda y ejecuta un comando tomando como argumentos los archivos encontrados.

Para ejecutar un comando sobre todos los archivos encontrados  
Debemos utilizar la siguiente sintaxis:

```
$find directorio patrón de búsqueda -exec comando {} \;
```

La opción `-exec` indica que se ejecutará el comando indicado a continuación sobre los archivos encontrados que se ingresen como argumentos



# Comando find



Ejemplo: Eliminar un archivo teniendo en cuenta que posee enlaces. Considerar que los enlaces se encuentran distribuidos a partir del homedirectory:

```
$find $HOME -inum n° de i-nodo -exec comando {} \;
```

Previamente averiguamos el número de i-nodo ejecutando ls -li

Comprobar si el archivo fue eliminado:

```
$find $HOME -inum n° de i-nodo
```





```
clarisa@DESKTOP-D79LATI:~$ cat >consigna1
Crear un enlace de comando10 en d1 con el nombre enlace 20.
Crear un enlace de comando10 en dire con el nombre enla30.
```

```
clarisa@DESKTOP-D79LATI:~$ ln ./comando10 d1/enlace20
clarisa@DESKTOP-D79LATI:~$ ln ./comando10 dire/enla30
clarisa@DESKTOP-D79LATI:~$ ls -l d1/enlace20
-rw-r--r-- 3 clarisa clarisa 67 May 23 20:10 d1/enlace20
```

```
clarisa@DESKTOP-D79LATI:~$ cat>consigna6
Eliminar el archivo comando10 teniendo en cuenta que
posee enlaces.
```







```
clarisa@DESKTOP-D79LATI:~$ cat>consigna6
Eliminar el archivo comando10 teniendo en cuenta que
posee enlaces.
clarisa@DESKTOP-D79LATI:~$ ls -i comando10
111464090777451964 comando10
clarisa@DESKTOP-D79LATI:~$ ls -i d1/enlace20
111464090777451964 d1/enlace20
clarisa@DESKTOP-D79LATI:~$ ls -i dire/enla30
111464090777451964 dire/enla30
clarisa@DESKTOP-D79LATI:~$ find ~ -inum 111464090777451964
/home/clarisa/comando10
/home/clarisa/d1/enlace20
/home/clarisa/dire/enla30
clarisa@DESKTOP-D79LATI:~$ find ~ -inum 111464090777451964 -exec rm {} \;
clarisa@DESKTOP-D79LATI:~$ cat>consigna7
Verificar si el archivo fue eliminado.
clarisa@DESKTOP-D79LATI:~$ find ~ -inum 111464090777451964
clarisa@DESKTOP-D79LATI:~$
```



# Comando find – Ejemplos:



Buscar a partir del homedirectory los archivos regulares.  
Buscar a partir del homedirectory los directorios.

Buscar a partir del homedirectory los archivos (de cualquier tipo) cuyo nombre sea d2.

```
clarisa@DESKTOP-D79LATI:~$ find ~ -name d2
/home/clarisa/d1/d2
/home/clarisa/d2
```

Buscar a partir del homedirectory los archivos regulares  
cuyo nombre sea d2.

```
clarisa@DESKTOP-D79LATI:~$ find $HOME -type f -name d2
```

Buscar a partir del homedirectory los directorios  
cuyo nombre sea d2.

```
clarisa@DESKTOP-D79LATI:~$ find ~ -type d -name d2
/home/clarisa/d1/d2
```

```
clarisa@DESKTOP-D79LATI:~$ cat>semana  
lunes martes  
martes  
miercoles  
viernes
```

Mostrar el contenido de **todos** los **archivos regulares** ubicados **a partir** del **directorío actual** cuyo **nombre** sea semana

```
$find $PWD -type f -name semana -exec cat {} \;
```

O bien: `$find -type f -name semana -exec cat {} \;`

**Por defecto la búsqueda se realiza a partir del directorio actual:**

```
clarisa@DESKTOP-D79LATI:~$ find -type f -name semana -exec cat {} \;  
lunes martes  
martes  
miercoles  
viernes
```



# Comando tr



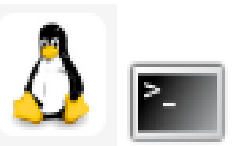
Comando tr.

Comando tr -s " "

Convierte a los espacios que hay entre un campo y el siguiente, en un único espacio.

Ejemplo:

```
clarisa@DESKTOP-D79LATI:~$ echo "Buenos      días"|tr -s " "  
Buenos días  
clarisa@DESKTOP-D79LATI:~$
```





# Comando tr

El comando tr puede utilizarse para filtrar los números que hay en un texto.

Ejemplo:

```
clarisa@DESKTOP-D79LATI:~$ echo "El día del mes es $num"
El día del mes es 10
clarisa@DESKTOP-D79LATI:~$ num=`echo "Hoy es 10 de Septiembre" |tr -cd [:digit:]`
clarisa@DESKTOP-D79LATI:~$ echo "El día del mes es $num"
El día del mes es 10
```



# Comando tr

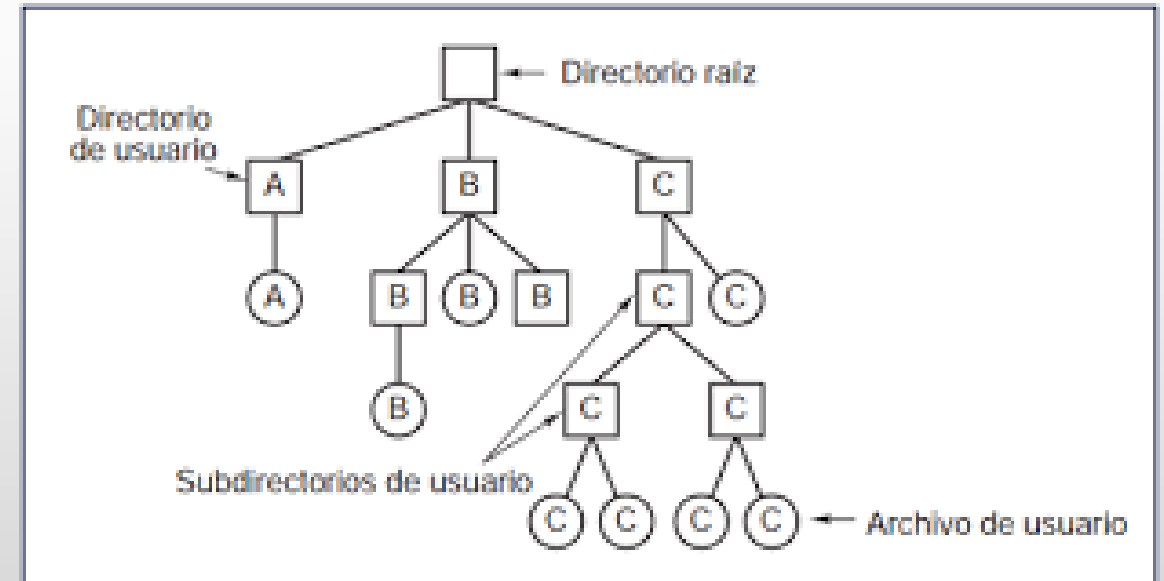
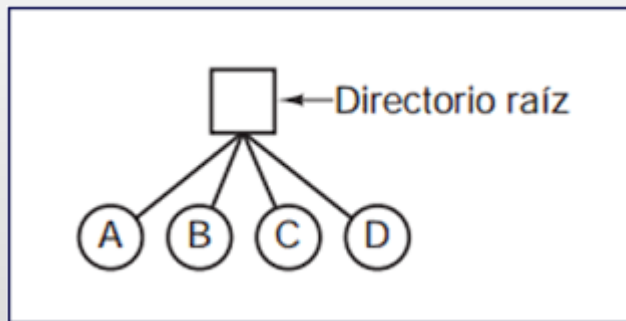
El comando tr puede utilizarse para transformar las minúsculas en mayúsculas

Ejemplo:

```
clarisa@DESKTOP-D79LATI:~$ cat>fecha
hoY es 10 de septiembre
clarisa@DESKTOP-D79LATI:~$ cat fecha|tr "a-z" "A-Z">fechamay
clarisa@DESKTOP-D79LATI:~$ cat fechamay
HOY ES 10 DE SEPTIEMBRE
```

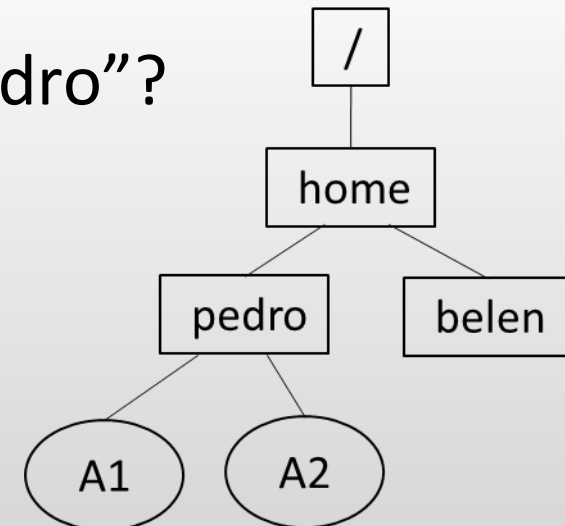
# Directorios

- Un directorio es un archivo especial que puede contener archivos y directorios.
- Permite organizar los archivos del usuario en el sistema de archivos
- Directorios de un solo nivel
- Directorios jerárquicos



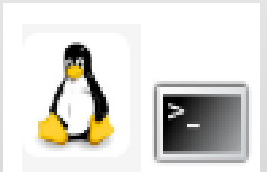
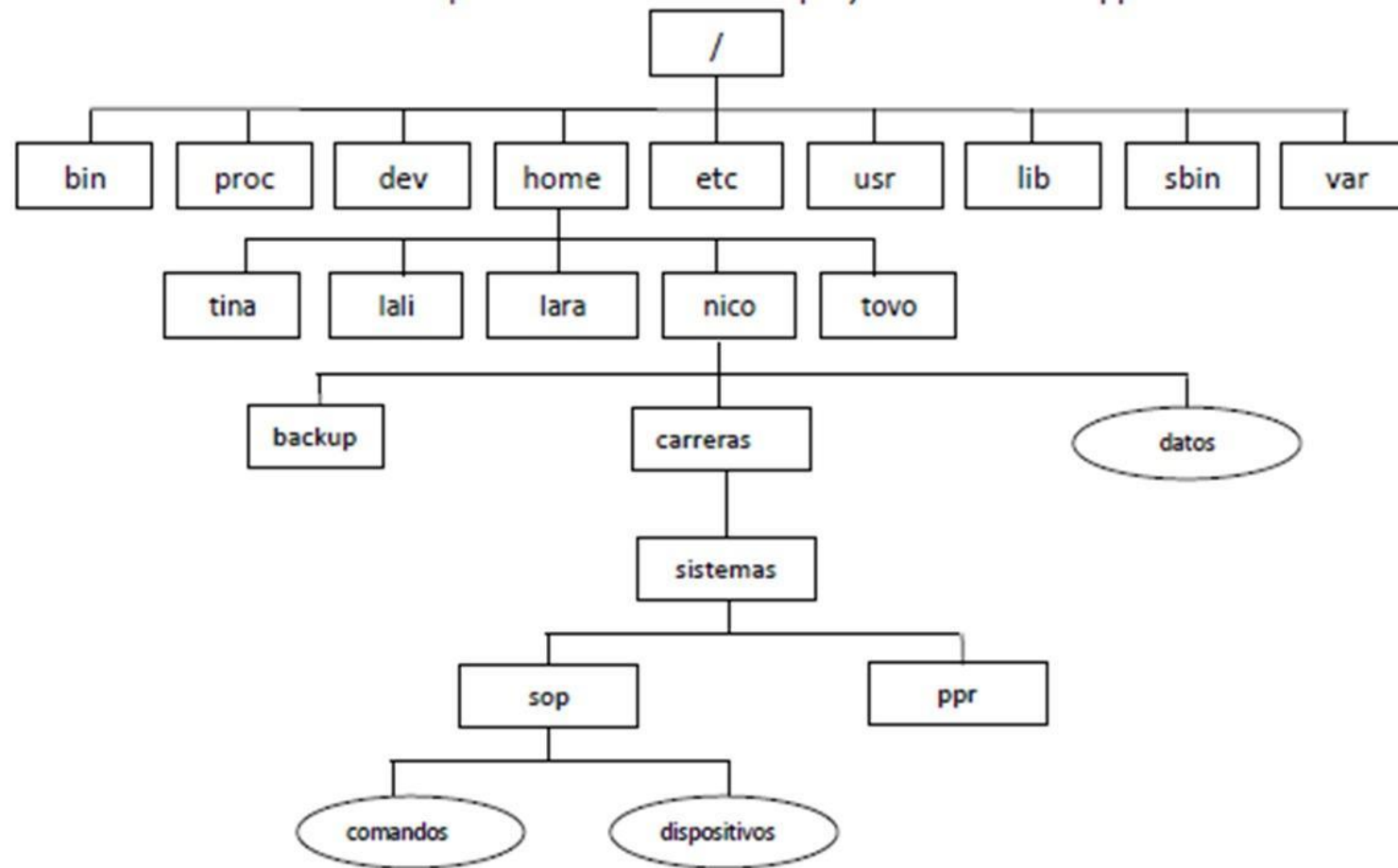
# Directorios

- Cuando se crea un directorio, automáticamente se crean los directorios “.” (punto) y “..” (punto punto)
- El directorio “.” hace referencia al propio directorio
- El directorio “..” hace referencia al directorio padre
- ¿Quién es el directorio padre del directorio “pedro”?
- ¿Quién es el directorio padre de “home”?
- ¿Quién es el directorio padre de “/”?

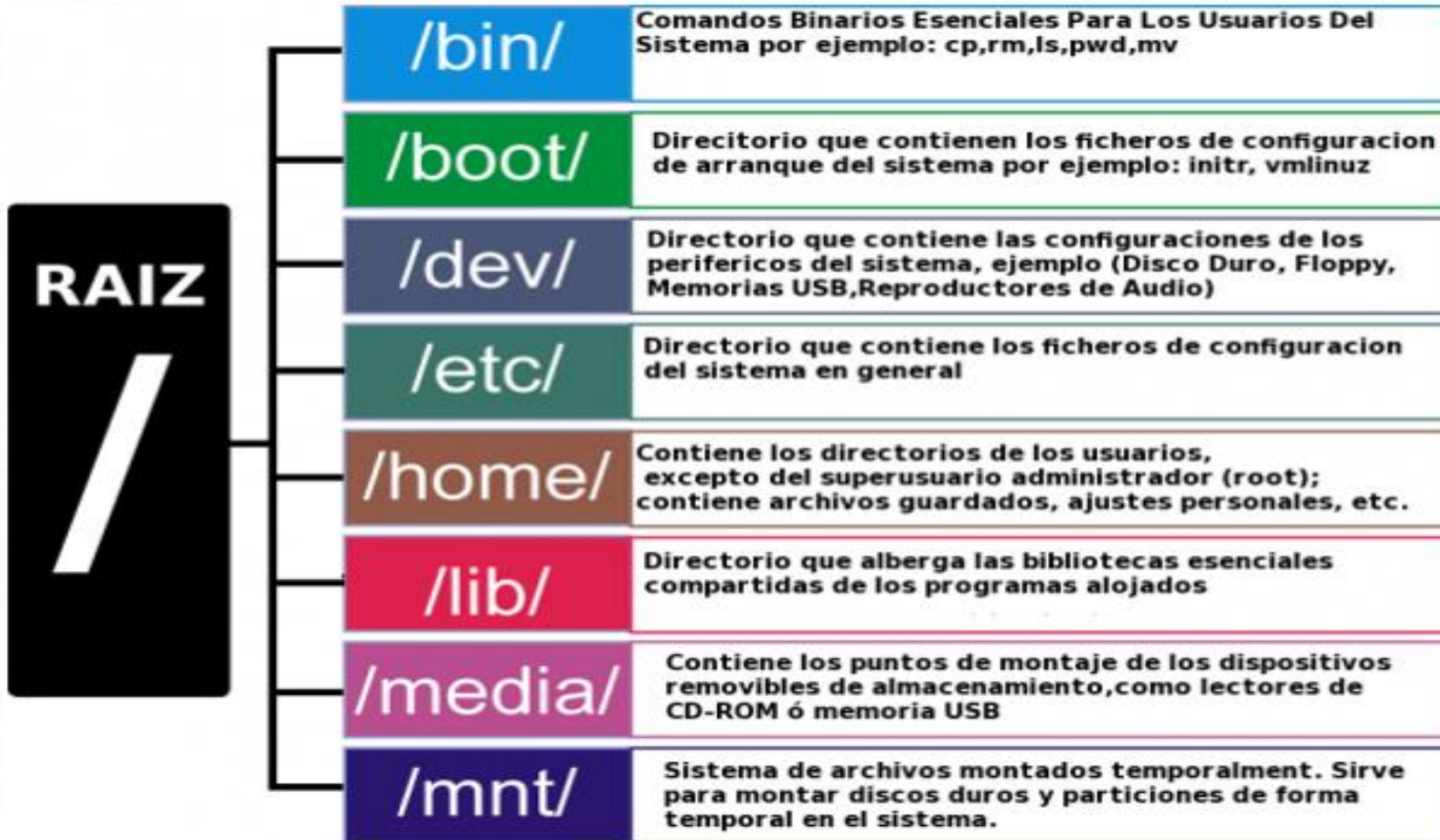




# Estructura de directorios en Linux



# Estructura de directorios en Linux



# Rutas absolutas y relativas

Las **rutas absolutas** empiezan siempre en el directorio raíz “/” y son únicas

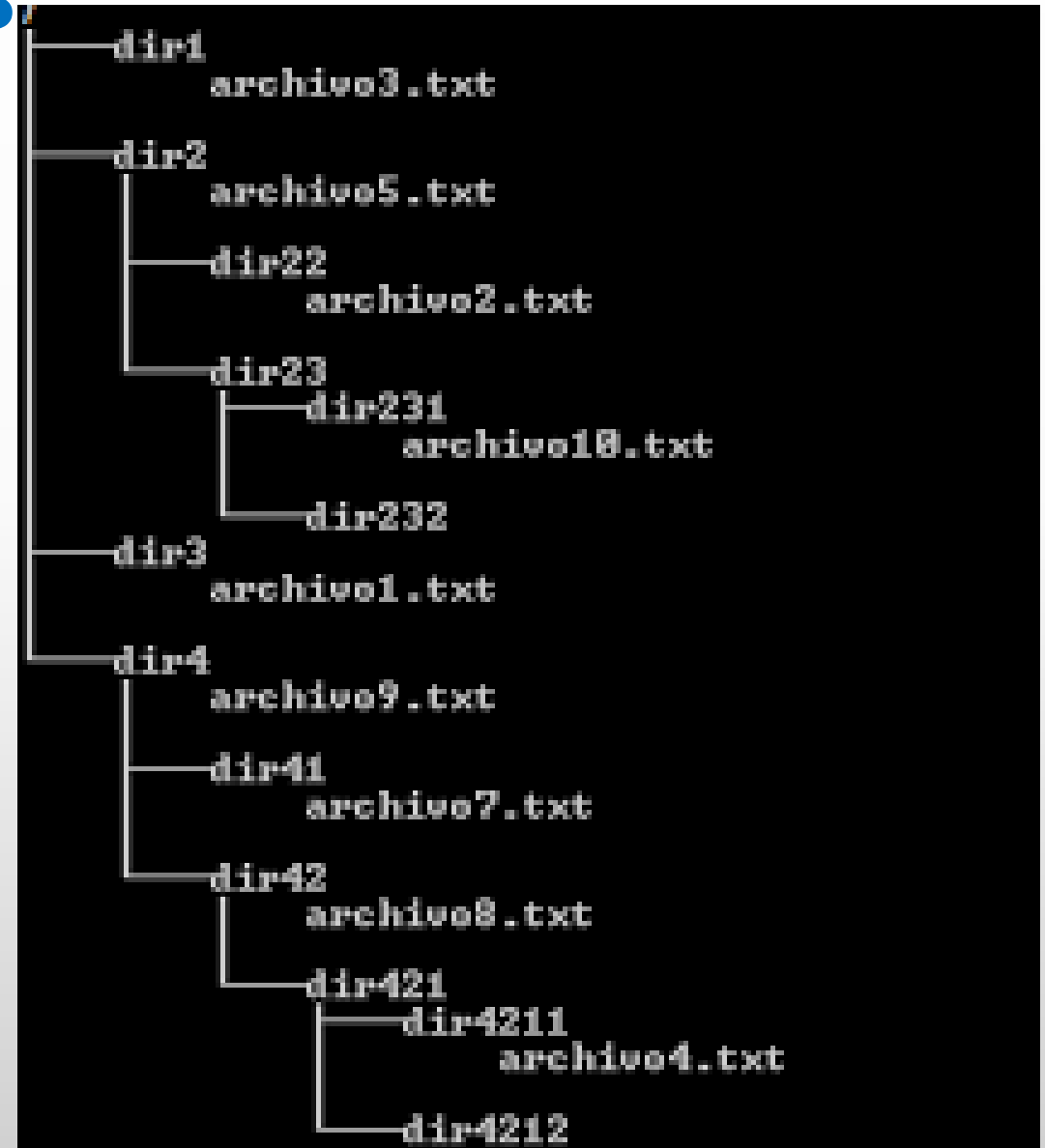
Ejemplo: /dir1/archivo3.txt

/dir2/dir22/archivo2.txt

Las **rutas relativas** dependen del directorio de trabajo o directorio actual.

Si estamos en el directorio **dir2**, para llegar al **archivo1.txt** ubicado en **dir3**, la ruta relativa sería:

../dir3/archivo1.txt



# Directorio de conexión y directorio actual



**Homedirectory o directorio de conexión:**

Directorio en el cual queda posicionado el usuario cuando inicia la sesión de trabajo.

Por defecto queda creado en **/home** cuando el usuario crea un nuevo usuario.

Al inicio de la sesión el homedirectory es el **directorio actual**.

Si posteriormente el usuario se cambia hacia otro directorio entonces el homedirectory deja de ser el **directorio actual**.

# Directorio de conexión y directorio actual (continuación)



El directorio actual cambia a medida que modificamos nuestra posición en el árbol.

Al inicio de la sesión el homedirectory es el directorio actual.

Si posteriormente el usuario se cambia hacia otro directorio entonces el homedirectory deja de ser el directorio actual.

# Operaciones con Directorios

- Create `?` crea un directorio vacío, excepto por los directorios “.” y “..”  
`$ mkdir directorio_nuevo`
- Delete `?` elimina un directorio vacío  
`$ rmdir directorio_vacio`
- Opendir `?` abre un directorio, para poder luego leer su contenido
- Closedir `?` cierra un directorio, liberando espacio en tablas internas
- Readdir `?` permite leer cada entrada de un directorio
- Rename `?` permite modificar el nombre de un directorio
- Link `?` crea una entrada en un directorio. Un archivo puede aparecer en más de un directorio. Incrementa el contador de i-nodos  
`$ ln archivo_original archivo_enlace`
- Unlink `?` elimina una entrada de un directorio. Reduce el contador de i-nodos

# i-nodos (nodo índice)

- Todo archivo o directorio en Linux tiene asociado un número de i-nodo
- El i-nodo es un bloque de disco que contiene los atributos de un archivo y las direcciones de los bloques de datos
- El i-nodo es un número entero
- Los directorios en Linux contienen el nombre del archivo/directorio y su número de i-nodo asociado
- El atributo número de enlaces  $\ell$  identifica desde cuántos directorios se accede a un archivo

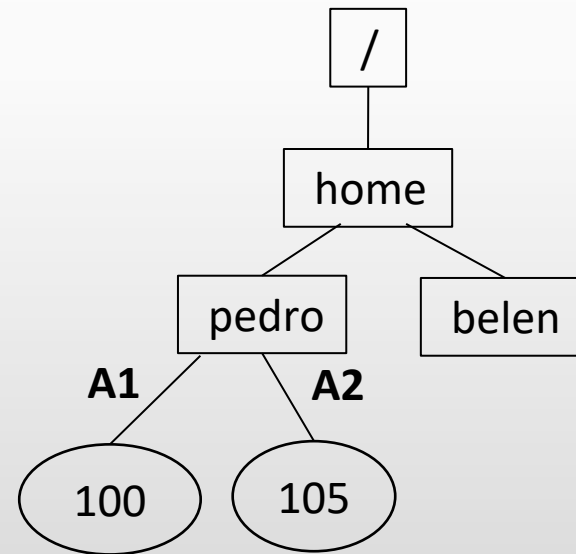
# Operaciones con Directorios (cont.)

- Link  crea una entrada en un directorio. Un archivo puede aparecer en más de un directorio. Incrementa el contador de i-nodos

## Directorio de pedro

Nombre Archivo	Nro. I-nodo
A1	100
A2	105

# In /home/pedro/A2 /home/belen/L-A2





# Operaciones con Directorios (cont.)

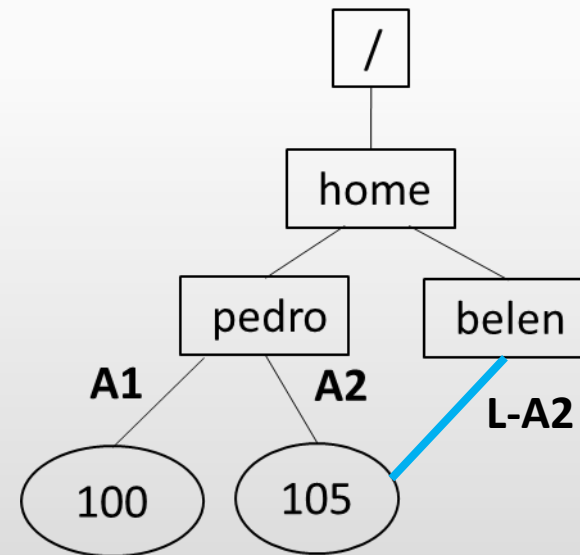
- Link `?` crea una entrada en un directorio. Un archivo puede aparecer en más de un directorio. Incrementa el contador de i-nodos

Directorio de pedro

Nombre Archivo	Nro. I-nodo
A1	100
A2	105

Directorio de belen

Nombre Archivo	Nro. I-nodo
L-A2	105



# In `/home/pedro/A2` `/home/belen/L-A2`

El atributo número de enlaces al i-nodo 105 ahora vale 2

# Comando In



Crea un enlace duro de un archivo (sólo podemos crearlo en una misma partición de disco)

**Al crear un enlace no generamos un nuevo archivo.  
Agregamos un puntero hacia el nodo i de un archivo ya existente.**

Observando el árbol podemos decir que, al crear un enlace agregamos una nueva ruta hacia un archivo ya existente.

También podemos considerar a los enlaces como un mismo archivo con nombres diferentes.

**Desde cualquiera de los enlaces podemos visualizar y modificar el contenido del archivo o sus permisos de acceso.**



# Comando ln -s



Crea un enlace simbólico de un archivo

Al crear un enlace simbólico no agregamos un puntero hacia el nodo i de un archivo ya existente.

Si posteriormente eliminamos el archivo ya existente, el enlace queda apuntando a la nada.



# Operaciones con Directorios (cont.)

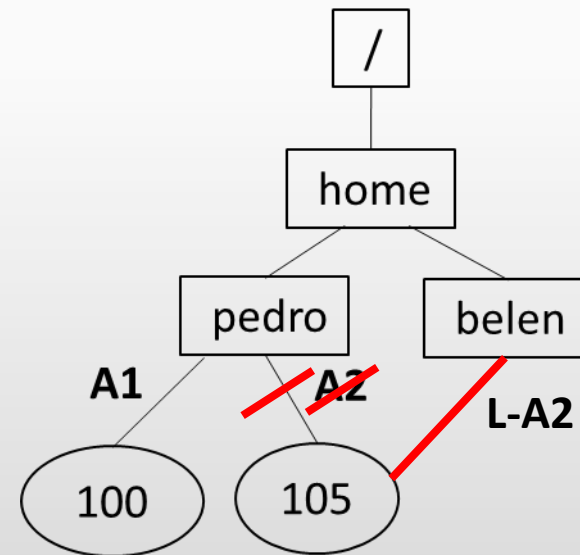
- Unlink `?` elimina una entrada de un directorio. Reduce el contador de i-nodos

Directorio de pedro

Nombre Archivo	Nro. I-nodo
A1	100
A2	<del>105</del>

Directorio de belen

Nombre Archivo	Nro. I-nodo
L-A2	105



```
# rm /home/pedro/A2
```

El atributo número de enlaces al i-nodo 105 ahora vale 1

# Comando In



**¿Qué sucede cuando eliminamos uno de los enlaces?**

Sólo borramos un puntero hacia el nodo  $i$ , es decir que solo borramos un acceso (una ruta) hacia el archivo.

El archivo se elimina cuando borramos todos los enlaces.

**¿De qué manera comprobamos que los enlaces son un mismo archivo?**

Visualizando el número de nodo  $i$ .

¿Dudas o Inquietudes?