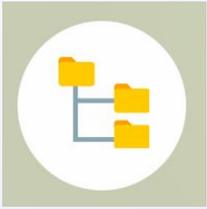
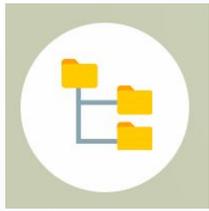


Administración de Archivos

**Laboratorio de
Sistemas
Operativos I**

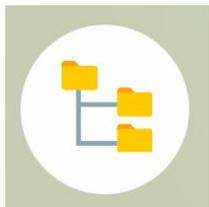


- Un **sistema de archivos** es un conjunto de normas y procedimientos establecidos para almacenar información.
- Un **archivo** es un conjunto independiente de datos (imagen, texto, etc.)
- Todos los datos en una computadora están organizados en archivos



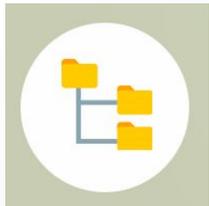
Sistemas de Archivos

- Es un componente del SO que se encarga de administrar el **espacio de almacenamiento**, asignar a los datos el espacio que necesiten, ordenarlos, permitir el acceso a ellos y su recuperación.
- Los dispositivos de almacenamiento de datos (discos duros internos o externos, USB, tarjeta SD) tienen un sistema de archivos para organizar la información guardada..
- Formatear un dispositivo de almacenamiento es crear un sistema de archivos en el dispositivo para guardar los datos.



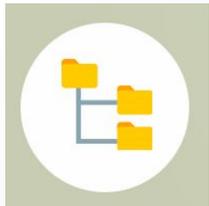
Cada SO cuenta con su propio Sistema de Archivos

Sistema de Archivos	Nativo en	Compatible con	Anteriores
FAT (File Allocation Table)	MS-DOS 1º Windows Extraíbles	Todos	FAT16 (2GB) FAT32 (4GB) exFat
NTFS (New Technology File System)	Windows	Mac (lectura) GNU/Linux	Alternativa a Fat32
APFS (Apple File System)	MacOS	GNU/Linux	HFS+
EXT4 (Extended file System)	GNU/Linux	GNU/Linux	EXT2 EXT3



CARACTERÍSTICAS DEL SISTEMA DE ARCHIVOS EXT4

- Menor uso del CPU. Mayor velocidad de lectura y escritura.
- Disminuye la fragmentación. Mayor seguridad.
- Máximo volumen de datos 1 EB (1 Exabyte)
- Máxima dimensión de archivo 16 TB (16 Terabytes)
- Máximo número de archivos 4 mil millones
- Nivel de profundidad en subdirectorios 64000



CARACTERÍSTICAS DE LOS ARCHIVOS EN GNU/LINUX

- Longitud de nombre de archivo: 255 caracteres (/usr/include/linux/limits.h)
- Cualquier caracter, excepto **NULL** y /. Para caracteres especiales usar “ ”
- Los archivos ocultos comienzan con un . (ejemplo .profile)
- Case sensitive: sensible a mayúsculas y minúsculas.

Nota2.txt



nota2.txt

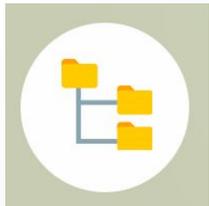


NOTA2.txt



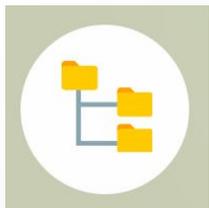
nota2.TXT

- Las extensiones de archivos no son obligatorias. El sistema reconoce el contenido del archivo por tipo de programa que puede manipularlo.
- La extensión no es un elemento diferenciado sino una convención dada a los nombres de archivo para reconocer su contenido. Se las utiliza por compatibilidad.



TIPOS DE ARCHIVOS EN GNU/Linux

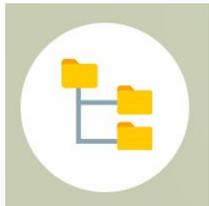
Comunes o Normales	Archivos ordinarios, contienen texto, código fuente, programas, etc.
Directorios	Archivos especiales, contienen los nombres de archivos y subdirectorios, y punteros hacia esos archivos y subdirectorios. Los archivos se muestran como si estuvieran “en” un directorio.
Especiales o de Dispositivos	Representan dispositivos físicos asociados al sistema (unidades de almacenamiento, impresoras, terminales, etc). Todo dispositivo físico que se conecte al ordenador está asociado a un archivo.
Enlaces o links	Archivos asociados a otros. Existen enlaces duros y simbólicos



Todos los sistemas Debian incluyen los siguientes directorios partiendo del directorio raíz /

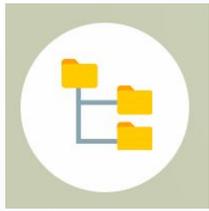
Observación: La partición raíz / siempre debe contener físicamente las particiones **/etc**, **/bin**, **/sbin**, **/lib** y **/dev**, sino el sistema no podrá arrancar.

<i>Directorio</i>	<i>Contenido</i>
bin	Binarios de órdenes esenciales
boot	Ficheros estáticos utilizados por el cargador de arranque
dev	Ficheros de dispositivos
etc	Configuración del sistema específico del host
home	Directorios de los usuarios
lib	Bibliotecas compartidas esenciales y módulos del núcleo
media	Puntos de montaje para medios extraíbles
mnt	Punto de montaje para un sistema de ficheros
proc	Directorio virtual para la información del sistema
root	Directorio Home para el usuario root
run	Datos variables en tiempo de ejecución
sbin	Binarios esenciales del sistema
sys	Directorio virtual para la información del sistema
tmp	Ficheros temporales
usr	Jerarquía secundaria
var	Datos variables
srv	Datos de los servicios ofrecidos por el sistema
opt	Paquetes de programas y aplicaciones opcionales



TIPOS DE DIRECTORIOS

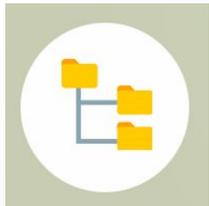
- **Estáticos:** Sólo root puede modificarlos. Estos archivos están disponibles para su lectura por cualquier otro usuario del sistema. (/bin, /sbin, /opt, /boot, /usr/bin, ...)
- **Dinámicos:** Son directorios cuyos archivos no son estáticos, es decir, cambian. Están disponibles tanto para ser leídos como para ser escritos (algunos sólo por el usuario propietario y por root). De estos tipos de directorios se suele hacer copias de seguridad más a menudo que de los directorios Estáticos. (/var/mail, /var/spool, /home, ...)
- **Compartibles:** Pueden compartirse a través de red, estar en un ordenador y ser usados desde otro. Ejemplos de esta categoría son: /usr/bin, /opt
- **Restringidos:** Son directorios y archivos que no se pueden compartir, su acceso y modificación están restringidos a root (/etc, /boot, /var/run, /var/lock, ...)



RUTA O PATH

- Es la forma de referenciar un archivo informático o directorio en un *sistema de archivos* de un *sistema operativo* determinado.
- Una ruta señala la localización exacta de un archivo o directorio mediante una cadena de caracteres concreta.
- Ejemplos

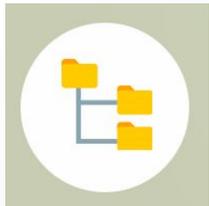
	<code>C:\Documents and Settings\Juan\Mis Documentos\carta.pdf</code>
	<code>/home/Juan/Documentos/carta.pdf</code>



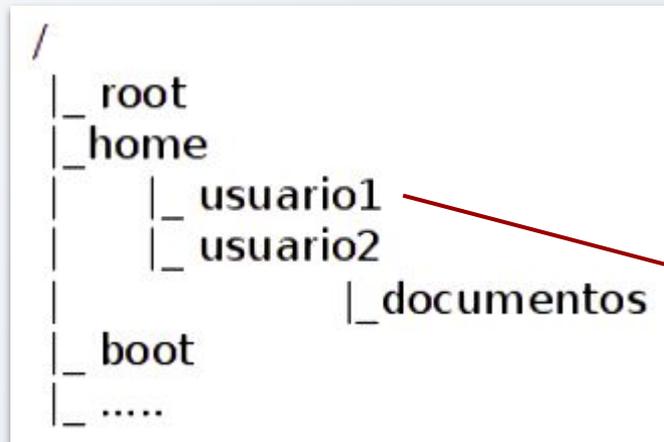
RUTA ABSOLUTA Y RUTA RELATIVA



RUTA ABSOLUTA	Señala la ubicación de un archivo o directorio desde el directorio raíz del sistema de archivos.	Siempre comienza por la RAÍZ	Si el directorio actual es <u>laura</u> /home/jose/carta.txt
RUTA RELATIVA	Es una ruta que parte del directorio actual como origen. Sólo es válida desde un directorio actual concreto , es decir es relativa a un directorio.	No lleva la RAÍZ	Si el directorio actual es <u>laura</u> ../jose/carta.txt



EJEMPLO DE RUTAS



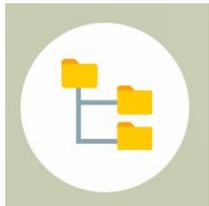
DIRECTORIO ACTUAL

- Ruta Absoluta al directorio documentos:
/home/usuario2/documentos
- Ruta Relativa al directorio documentos:
../usuario2/documentos



Administración de Directorios y Archivos

COMANDOS



Trasladarse de directorio: cd



<code>\$cd directorio</code>	Lleva al directorio
<code>\$cd</code>	Lleva al directorio personal
<code>\$cd /</code>	Lleva a la raíz
<code>\$cd .</code>	Deja en el directorio actual
<code>\$cd ..</code>	Lleva al directorio padre

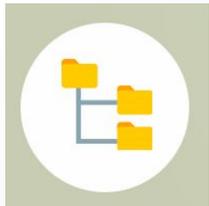
```
jperez@mipc:~$pwd  
/home/jperez
```

```
jperez@mipc:~$cd /etc/systemd
```

```
jperez@mipc:/etc/systemd$cd ../../
```

```
jperez@mipc:/$
```

¿Dónde está ubicado jperez?



Ver el contenido de un directorio: `ls`



<code>\$ls directorio</code>	Muestra contenido del directorio
<code>\$ls -a</code>	Muestra contenido del directorio incluidos los archivos ocultos
<code>\$ls -l</code>	Muestra contenido del directorio en forma detallada
<code>\$ls -R</code>	Muestra el contenido de todos los subdirectorios de forma recursiva.
<code>\$ls -S</code>	Ordena los resultados por tamaño de archivo.

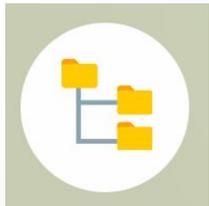
```
jperez@mipc:~$pwd  
/home/jperez
```

```
jperez@mipc:~$ls
```

```
jperez@mipc:~$ls /etc/systemd
```

```
jperez@mipc:~$ls /
```

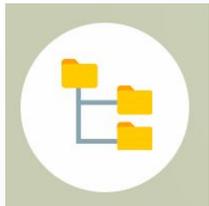
¿Qué lista en cada línea de comando?



Crear directorios: `mkdir`



<code>\$mkdir dir</code>	Crea un directorio llamado "dir" en el directorio actual	<pre>jperez@mipc:~\$pwd /home/jperez jperez@mipc:~\$mkdir libros jperez@mipc:~\$mkdir libros/terror jperez@mipc:~\$mkdir -p agenda/agosto</pre> <p>¿Cómo queda el árbol de directorios?</p>
<code>\$mkdir dir1 dir2</code>	Crea dos directorios llamados "dir1" y "dir2" en el directorio actual	
<code>\$mkdir dir3/dir4</code>	Crea el directorio "dir4" en dir3, siempre que dir3 exista	
<code>\$mkdir -p dir5/dir6</code>	Crea una rama completa de directorio. En este caso crea "dir5" y luego "dir6" como hijo.	

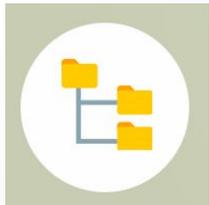


EJEMPLOS mkdir

```
jpervez@mipc:~$pwd  
home/jpervez  
  
jpervez@mipc:~$mkdir libros  
  
jpervez@mipc:~$mkdir libros/terror  
  
jpervez@mipc:~$mkdir -p agenda/agosto
```

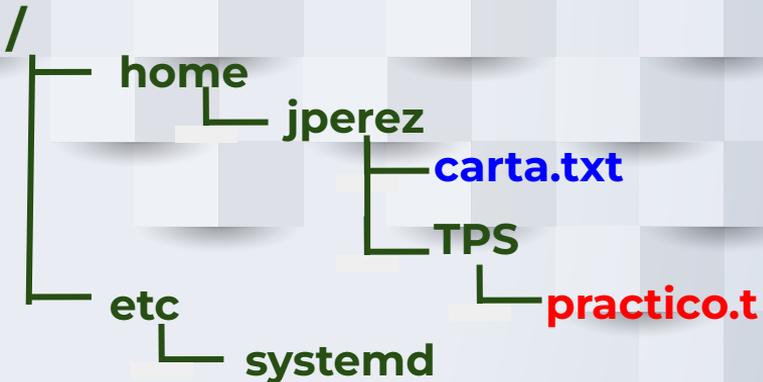
¿Cómo queda el árbol de directorios?



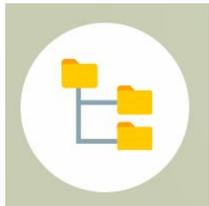


Crear archivos vacíos: touch

Crea archivos vacíos si no existen. Si existen les cambia la fecha de acceso/modificación.



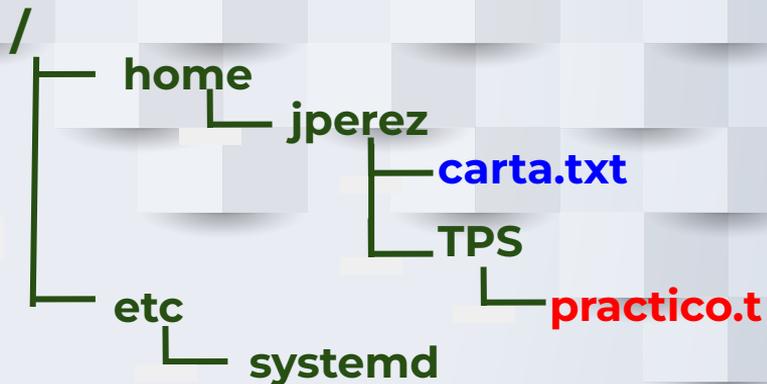
<code>\$touch archivo1</code>	Crea un archivo vacío llamado "archivo1" con la fecha actual y en el directorio actual	<pre>jperez@mipc:~\$pwd /home/jperez</pre>
<code>\$touch arch1 arch2</code>	Crea dos archivos vacíos llamados "arch1" y "arch2" con la fecha actual y en el directorio actual	<pre>jperez@mipc:~\$touch carta.txt jperez@mipc:~\$touch TPS/practico.t</pre>



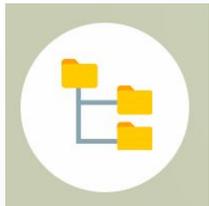
Eliminar directorios y archivos: `rmdir` y `rm`

rmdir: Elimina directorios vacíos.

rm: elimina archivos y directorios con contenido.



<code>\$rmdir directorio1</code>	Elimina el directorio1 si y sólo si está vacío	<pre>jperez@mipc:~\$pwd /home/jperez</pre>
<code>\$rm -r directorio2</code>	Elimina el directorio2 y TODO su contenido.	<pre>jperez@mipc:~\$rm carta.txt</pre>
<code>\$rm archivo1</code>	Elimina el archivo1	<pre>jperez@mipc:~\$rm TPS/practico.t jperez@mipc:~\$rmdir /opt/firefox</pre>



Copiar directorios y archivos: cp

\$cp origen destino



```
jperez@mipc:~$cp carta.txt /opt/firefox
```

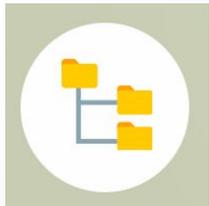
Copia el archivo carta.txt en el directorio firefox

```
jperez@mipc:/opt$cp -r firefox /home/jperez
```

Copia el directorio firefox y TODO su contenido en el directorio jperez

```
jperez@mipc:/home$cp practico.t jperez/practico.z
```

Copia el archivo practico.t en el directorio jperez y a la vez le cambia el nombre por practico.z



Mover directorios y archivos: mv

`$mv origen destino`



```
jperez@mipc:~$mv carta.txt /opt/firefox
```

Mueve el archivo carta.txt al directorio firefox

```
jperez@mipc:/opt$mv firefox /home/jperez
```

Mueve el directorio firefox y TODO su contenido al directorio jperez

```
jperez@mipc:/home$mv practico.t jperez/practico.z
```

Mueve el archivo practico.t al directorio jperez y a la vez le cambia el nombre por practico.z

```
jperez@mipc:~$mv carta.txt original.txt
```

Renombra el archivo carta.txt por original.txt

¡Gracias!

¿Preguntas?

