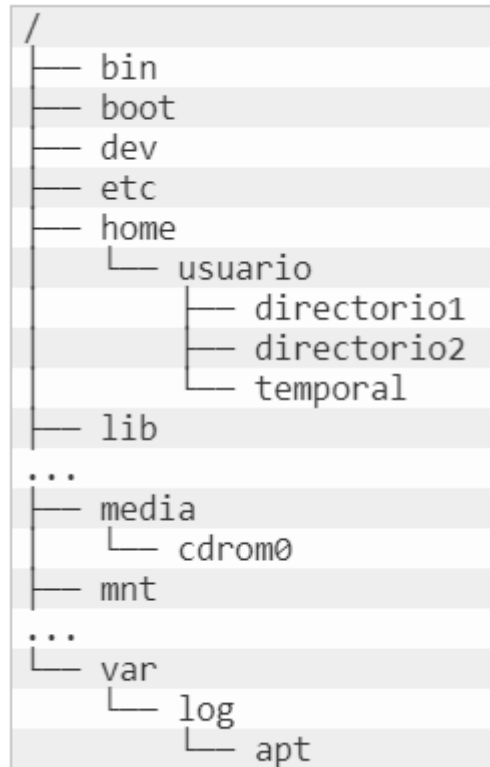


Rutas absolutas y relativas

En los sistemas basados en [Unix](#), el [sistema de ficheros](#) es una estructura jerárquica que parte de la [raíz](#). La [raíz](#), se representa con el carácter `/` y contiene los directorios principales del sistema y sus subdirectorios.



Tipos de rutas

Rutas absolutas

Como hemos visto en la Introducción, el sistema de ficheros en un sistema Linux es una estructura jerárquica que tiene una [raíz](#), que se indica con la barra (`/`).

La ruta absoluta a un directorio indica todos los directorios por los que hay que pasar empezando desde la raíz del sistema de archivos (`/`).

Ejemplo. Fijémonos en el **árbol de directorios** listado más arriba, supongamos que nuestro **directorio actual o de trabajo** es el **home del usuario** (`~`), y queremos ir al directorio **temporal** que se encuentra en el **home del usuario** utilizando una **ruta absoluta**.

Para hacerlo, ejecutaríamos el siguiente comando:

```
usuario@mipc:~$ cd /home/usuario/temporal  
y quedaría.
```

```
usuario@mipc:~/temporal$
```

Para comprobarlo, ejecutamos el comando `pwd`

```
usuario@mipc:~/temporal$ pwd  
/home/usuario/directorio1
```

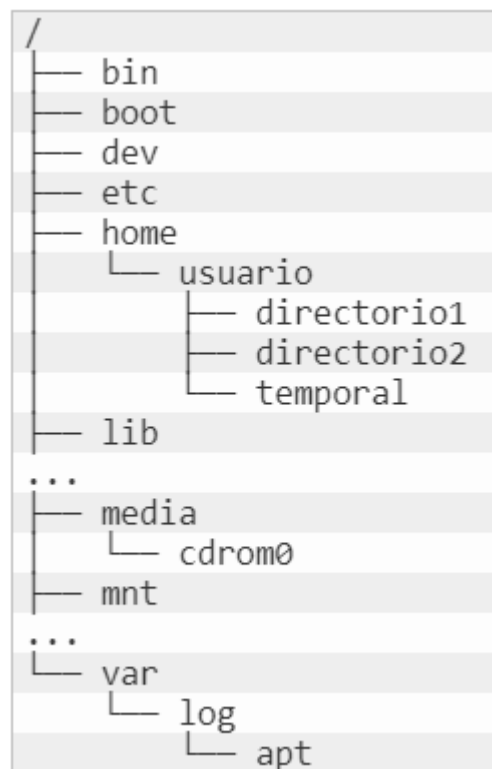
Como se ve en el ejemplo, hemos escrito la ruta completa desde el directorio raíz hasta el directorio **temporal**.

Las rutas absolutas suelen ser más largas que las rutas relativas, pero tienen la ventaja de que siempre funcionan, independientemente del lugar del **árbol de directorios** desde el que se ejecute el comando.

Rutas relativas

La **ruta relativa** a un directorio, indica el camino hacia el directorio al que queremos llegar, pero basándose en el directorio desde el que se ejecuta el comando, es decir, desde el **directorio actual o de trabajo**.

Ejemplo. Fijémonos de nuevo en el **árbol de directorios**.



1. supongamos que nuestro **directorio actual o de trabajo** es el **home del usuario** (~), y queremos ir al directorio **directorio1** que se encuentra en el **home del usuario**, podemos hacerlo indicando la ruta desde donde nos encontramos:

```
usuario@mipc:~$ cd directorio1
Quedaría
usuario@mipc:~/directorio1$
```

Para comprobarlo, ejecutamos el comando pwd:

```
usuario@mipc:~/directorio1$ pwd
Y nos mostraría:
/home/usuario/directorio1
```

En el ejemplo anterior nos hemos movido desde el **directorio actual** hacia delante, pero también podemos hacerlo hacia atrás. Para hacerlo, debemos tener claro el concepto de **directorio padre**.

Directorio padre

El **directorio padre** es aquel que contiene el **directorio actual o de trabajo**. Para construir rutas relativas es necesario saber que **..** indica el **directorio padre**.

Si ejecutamos **cd ..** lo que estamos haciendo es cambiar al **directorio padre** del **directorio actual**, un directorio justo antes en el **árbol de directorios**.

Si hacemos un listado del **directorio actual** incluyendo los elementos ocultos obtendremos la siguiente información:

```
usuario@mipc:~/temporal$ ls -la
total 8
drwxr-xr-x  2 karfer karfer 4096 ago  3  2019 .
drwxr-xr-x 66 karfer karfer 4096 may  4 08:17 ..
```

Si listamos con el comando **ls -la** siempre se muestran los directorios ocultos **.** y **..**, los cuales tienen el siguiente significado:

- a) **.** : Hace referencia al directorio donde nos encontramos. Cuando ejecutamos un programa y escribimos **./programa.sh**, ese punto delante del ejecutable indica que este se encuentra en el **directorio de trabajo**.

b) `..` : Hace referencia al **directorio padre**. Utilizando `..` podemos indicarle al comando `cd` (o a cualquier otro comando) que retroceda de nivel.

Veamos unos ejemplos:

Supongamos que nos encontramos en el directorio `/home/usuario/directorio1`:

- Para movernos del **directorio actual** (`/home/usuario/directorio1`) al **directorio padre** (`/home/usuario`), un nivel hacia atrás, podemos ejecutar el siguiente comando:

```
usuario@mipc:~/directorio1$ cd ..
```

Para comprobarlo, ejecutamos el comando `pwd`:

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

- Para movernos varios niveles hacia atrás hasta llegar al directorio raíz `/` podemos ejecutar el siguiente comando:

Comprobamos con `pwd` cual es nuestro **directorio actual**:

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

Nos movemos dos niveles hacia atrás:

```
usuario@mipc:~$ cd ../../
```

Y volvemos a comprobar con `pwd` que ahora nuestro **directorio actual** es el directorio raíz `/`:

```
usuario@mipc:~$ pwd
/
```

Veamos un ejemplo en el que bajamos de nivel y luego subimos con una sola orden. Fijémonos en el **árbol de directorios** listado más arriba y vamos a movernos desde el directorio

`/home/usuario/directorio1` hasta el directorio `/home/usuario/directorio2`, pero utilizando una **ruta relativa**:

En el ejercicio anterior, nuestro **directorio de trabajo** era `/home/usuario`. Pues bien, vamos a cambiar nuestro **directorio de trabajo** a `/home/usuario/directorio1`:

```
usuario@mipc:~$ cd directorio1
```

Comprobamos cual es nuestro **directorio de trabajo** con `pwd`:

```
usuario@mipc:~/directorio1$ pwd  
/home/usuario/directorio1
```

Y desde `directorio1`, utilizando una ruta relativa, nos vamos a `directorio2`. Bajamos un nivel para luego subir un nivel:

```
usuario@mipc:~/directorio1$ cd ../directorio2
```

●Comprobamos cual es nuestro **directorio de trabajo** con `pwd`:

```
usuario@mipc:~/directorio2$ pwd  
/home/usuario/directorio2
```