

## Laboratorio 2

### Comandos básicos para la gestión de ficheros y directorios

Entrega obligatoria al final de la sesión presencial de laboratorio

#### Objetivos:

- Conocer los comandos básicos de Unix para la gestión de ficheros y directorios
- Conocer los comandos que permiten visualizar y gestionar la estructura de directorios de Unix

#### Entorno:

Ordenadores del laboratorio. Pero las actividades que se describen pueden realizarse en cualquier equipo con un sistema de tipo Unix.

#### Actividades previas:

Antes de la sesión de laboratorio que corresponda a su grupo deberá usted leer este documento y realizar las **actividades guiadas** que en él se describen (excepto el apartado de “Entrega de resultados”), bien en un ordenador del mismo laboratorio (en horario libre) o en cualquier otro ordenador. De no hacerlo, es posible que no le dé tiempo a realizar la entrega. De este modo, la sesión de laboratorio tiene la función de una clase de tutoría para resolver las dudas que se le hayan podido presentar.

**Resultados:** **Cuatro ficheros de texto** según se indica en los enunciados, que subirá al Moodle al final de la sesión de laboratorio. No se podrán subir dichos ficheros después de la hora de terminación del laboratorio de su grupo. En concreto:

- a) Uno de los ficheros de texto contendrá todas las órdenes que ha utilizado en las **actividades guiadas** descritas en los siguientes apartados, como se indica al final de este documento. **El fichero de texto debería llevarlo al laboratorio ya generado si ha realizado todas las actividades previamente.**
- b) Un segundo fichero será un **programa JavaScript** que responda a las especificaciones que se describen en el correspondiente apartado al final de este documento.
- c) El profesor proporcionará al comienzo del laboratorio un enunciado con la práctica a realizar durante la **sesión presencial**. Los resultados se guardaran en dos ficheros de texto adicionales que se corresponderán con el **tercer y cuarto fichero de texto** que subirá al Moodle.

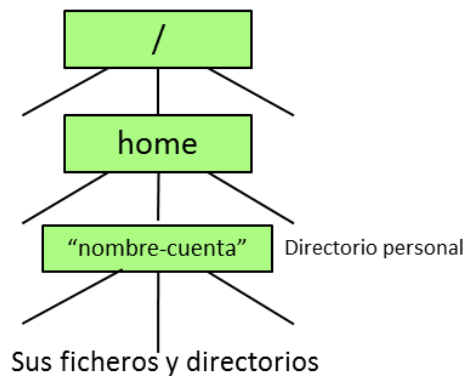
#### Inicio

Abra un terminal de texto desde la interfaz gráfica y realice las actividades guiadas que se indican. En lo sucesivo, el símbolo **\$** significa que debe usted escribir la orden indicada.

#### 1. Directorio home

El árbol del sistema de ficheros parte del directorio raíz (/). De dicho directorio cuelgan a su vez varios subdirectorios. Uno de ellos es el directorio `home`, de donde cuelga a su vez los directorios personales de los usuarios. El nombre de su directorio personal es `nombre-cuenta` y de ese

segundo directorio es de donde cuelgan todos sus ficheros y directorios.



El nombre del directorio personal es diferente para cada usuario, se corresponde con el nombre de usuario asignado a cada uno, `nombre-cuenta`, por ejemplo `juan.garcia`.

Denominaremos “directorio de trabajo actual” al directorio en el árbol del sistema de ficheros, donde se encuentra situado en un instante determinado. Todas las órdenes que ejecute lo harán respecto al directorio de trabajo actual.

Cuando se conecta a su cuenta en el ordenador, el sistema sitúa su directorio de trabajo actual en su directorio personal.

Para descubrir qué hay en su directorio personal teclee:

```
$ ls
```

Observará que aparece el directorio `lab1` de la práctica anterior y el directorio `Escritorio` con enlaces que corresponden a su entorno gráfico.

## 2. Creación de directorios

Para crear subdirectorios se utiliza la orden **mkdir** (**make directory**):

```
$ mkdir lab2
```

Crea el subdirectorio `lab2` que cuelga del directorio de trabajo actual. El directorio de trabajo actual, que en este momento, si no ha ejecutado ninguna orden más, es su directorio personal. Por tanto, se crea el directorio `lab2` que cuelga de su directorio personal.

Compruébelo tecleando:

```
$ ls -l
```

## 3. Cambiarse de directorio

Con la orden **cd** (**change directory**) se cambia el directorio de trabajo, es decir se mueve en el árbol del sistema de ficheros. Cambie el directorio de trabajo actual a `lab2`, cambiando por tanto su posición actual en el árbol del sistema de ficheros:

```
$ cd lab2
```

Mire el contenido de ese directorio, que debe estar vacío:

```
$ ls -l
```

En Unix “.” significa el directorio actual y “..” el directorio inmediatamente superior, es decir, el padre del actual directorio de trabajo. Suba al directorio padre del actual, que si no ha tecleado nada más es su directorio personal, tecleando:

```
$ cd ..
```

Independientemente de donde se encuentre en un momento dado en el árbol del sistema de ficheros, podrá cambiarse a su directorio personal tecleando `cd` sin opciones. Esto es muy útil cuando se está perdido en el árbol del sistema de ficheros.

Tecleando “`cd /`” le lleva al directorio raíz.

#### 4. Rutas

Para identificar un fichero o directorio debe dar la ruta o camino de dicho fichero o directorio en el árbol del sistema de ficheros. La ruta puede ser absoluta o relativa. Una ruta absoluta señala la localización exacta de un fichero o directorio en el árbol del sistema de ficheros desde la raíz. Una ruta relativa señala la localización exacta de un fichero o directorio en el árbol del sistema de ficheros desde el directorio de trabajo actual.

Por ejemplo, para identificar el directorio `lab2` que ha creado previamente, se puede hacer mediante su ruta absoluta: `/home/nombre-cuenta/lab2` o mediante su ruta relativa, que, si por ejemplo su directorio de trabajo actual es su directorio personal, sería `lab2` o si su directorio de trabajo actual es `home`, sería `nombre-cuenta/lab2`.

Mediante la orden **`pwd` (print working directory)** podrá saber dónde se encuentra en cada momento en el árbol del sistema de ficheros.

Si no se ha cambiado de directorio y sigue en su directorio personal, tecleando `pwd` conocerá la ruta absoluta de su directorio personal:

```
$ pwd
```

Utilice las órdenes `cd`, `ls` y `pwd` para explorar el sistema de ficheros. Recuerde que con la orden `cd` sin opciones se cambia a su directorio personal.

#### 5. Entendiendo las rutas

Se puede referir a su directorio personal con el carácter “`~`”<sup>1,2</sup>. Para cambiarse al directorio `lab2`, que ha creado previamente, independientemente desde donde se encuentre actualmente en el árbol del sistema de ficheros, teclee:

```
$ cd ~/lab2
```

Cree con `nano` un fichero que se llame `ejemplo.txt`. Escriba una línea con un texto cualquiera en dicho fichero y salga del editor. Teclee:

```
$ cd ..  
$ ls -l  
$ ls -l ejemplo.txt
```

Mediante estás órdenes se ha cambiado al directorio padre, que es su directorio personal, se ha listado los nombres de ficheros y directorios que se encuentran en dicho directorio y por último se ha ordenado generar un listado largo del fichero `ejemplo.txt`.

Como resultado de ésa última orden ha obtenido un mensaje que le indica que no existe el fichero o directorio `ejemplo.txt`. La razón es que `ejemplo.txt` no se encuentra en el directorio actual de trabajo, su directorio personal, sino que cuelga del directorio `lab2`.

---

<sup>1</sup> `cd` es equivalente a `cd ~` y a `cd ~/`

<sup>2</sup> Una de las formas de escribir la virgulilla (`~`) es con la combinación de teclas “`ALTGr`” + “`4`” y un espacio a continuación

Para identificar el fichero o directorio que va a utilizar una orden, por ejemplo, la orden `ls -l` sobre el fichero `ejemplo.txt`, deberá dar la ruta absoluta o relativa de dicho fichero o directorio. La ruta relativa dependerá del directorio actual de trabajo. Por ejemplo:

- Que el fichero o directorio se encuentre en el directorio actual de trabajo. Para ello copie primero el fichero `ejemplo.txt` a su directorio actual de trabajo y cámbiele de nombre a `ejemplo1.txt`

```
$ cp lab2/ejemplo.txt ejemplo1.txt
$ ls -l ejemplo1.txt
```

- Se puede previamente cambiar el directorio actual de trabajo al directorio donde se encuentre el fichero `prueba`

```
$ cd lab2
$ ls -l ejemplo.txt
```

- Se puede utilizar una ruta relativa para identificar el fichero o directorio desde el directorio de trabajo actual. Teclee `cd` para el directorio de trabajo actual vuelva a ser su directorio personal

```
$ cd
$ ls -l lab2/ejemplo.txt
```

- Se puede utilizar la ruta absoluta para identificar al fichero o directorio desde el directorio raíz, independientemente de donde esté situado su directorio actual de trabajo. Suponiendo que su directorio de trabajo actual es cualquiera, teclee por ejemplo:

```
$ cd ~/lab2
$ ls -l /home/nombre-cuenta/lab2/ejemplo.txt
```

Pruebe a volver a utilizar todas las órdenes anteriores con otros ejemplos, para familiarizarse con ellas y muévase por los directorios del sistema para comprender el árbol del sistema de ficheros y cómo moverse en el mismo.

## 6. Borrar directorios

Con la orden **`rmdir`** (**remove directory**) se borran directorios. En concreto si se teclea `rmdir nombre-directorio`, se borrará dicho directorio pero únicamente si está vacío.

Si `rm` se utiliza con la opción `-r` borra además del directorio todos los ficheros y directorios que cuelgan de él, de forma recursiva. Esta opción es muy peligrosa ya que puede borrar mucha información no deseada si se equivoca al indicar el nombre del directorio que quería borrar.

Si `rm` se utiliza con la opción `-i`, se pide confirmación antes de proceder a borrar la información.

Sitúese en su directorio personal y borre el directorio `lab2` y los ficheros que cuelgan de él.

```
$ cd
$ rm -ri lab2
```

## 7. Ejercicio a realizar sobre comandos para la gestión de ficheros y directorios

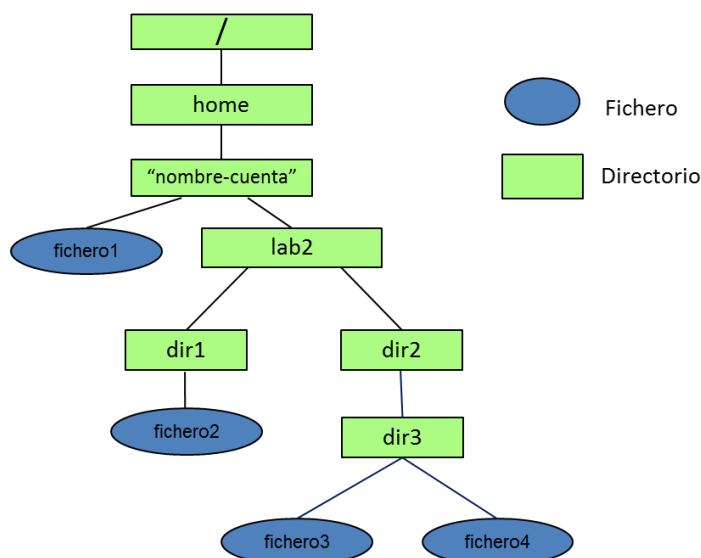
- a) Para generar el fichero de resultados que debe subir a Moodle, borre antes todos los ficheros y directorios que ha podido crear previamente, con las ordenes `rmdir` y `rm`
- b) Borre la historia de las órdenes que ha tecleado previamente y escriba su nombre y apellidos, para que quede registrado en la historia

```
$ history -c
$ Nombre Apellido Apellido
```

Si en algún momento se equivoca al teclear o en las órdenes que debe ejecutar, no se preocupe y repita las órdenes. No es necesario que borre todo el historial y repita de nuevo el ejercicio, se consideraran válidas las últimas órdenes ejecutadas de cada apartado.

- c) Cree la siguiente estructura de directorios y ficheros. Cree los directorios con la orden `mkdir`. Con el editor `nano`, cree los ficheros teniendo en cuenta que cada uno de ellos deberá tener el siguiente contenido:

- Su nombre y dos apellidos
- Su grupo de FTEL
- Nombre del fichero



Cambie su directorio de trabajo actual a su directorio personal y compruebe que ha creado la estructura anterior con la orden `tree` o con `ls -R`:

**\$ tree**

Observe que además de los ficheros y directorios creados previamente, verá un directorio `Escritorio` con los enlaces de su entorno gráfico.

- d) **Sitúese en su directorio personal y utilizando rutas absolutas** para identificar a los directorios, ejecute las órdenes necesarias que le permitan:
- a. Moverse al directorio `dir3`
  - b. Ir al directorio `dir1`
  - c. Volver a su directorio personal
- e) **Sitúese en su directorio personal y utilizando rutas relativas** para identificar a los directorios, ejecute las órdenes necesarias que le permitan:
1. Moverse al directorio `dir2`
  2. Ir al directorio `dir1`
  3. Ir al directorio `dir3`
  4. Volver a su directorio personal
- f) **Sitúese en su directorio personal, sin moverse de dicho directorio y utilizando rutas absolutas** para identificar a los ficheros y directorios, ejecute las órdenes necesarias (utilice parte de las órdenes que aprendió en el laboratorio1) que le permitan:
1. Sacar por pantalla el contenido del `fichero1` (no utilice un editor)
  2. Editar el `fichero4` y añadir el nombre de esta asignatura

3. Cambiar el nombre del `fichero3` a `fichero5`
- g) **Sitúese en su directorio personal, sin moverse de dicho directorio y utilizando rutas relativas** para identificar a los ficheros y directorios, ejecute las órdenes necesarias (utilice parte de las órdenes que aprendió en el laboratorio1) que le permitan:
1. Ver un listado largo con las características (nombre, permisos, tamaño,...) de los ficheros que hay en el directorio `dir3`
  2. Editar el `fichero4` y añadir la fecha actual
  3. Sacar por pantalla el contenido del `fichero2` (no utilice un editor)
  4. Cambiar el nombre del `fichero4` a `fichero6`
- h) Cambie su directorio de trabajo actual a su directorio personal y compruebe con la orden `ls -R` o con la orden `tree` que los cambios de nombre de los ficheros se han realizado correctamente.
- i) **Sitúese en el directorio `dir1`, sin moverse de dicho directorio y utilizando rutas absolutas** para identificar a los ficheros y directorios, ejecute las órdenes necesarias que le permitan:
1. Crear un nuevo fichero llamado `fichero7` que cuelgue del directorio `dir2` y cuyo contenido sea el mismo que el del fichero `fichero1` (no utilice un editor)
  2. Borrar todos los ficheros cuyo nombre comience por `fich` y se encuentren en el directorio `dir3`
- j) **Sitúese en el directorio `dir1`, sin moverse de dicho directorio y utilizando rutas relativas** para identificar a los ficheros y directorios, ejecute las órdenes necesarias que le permitan:
1. Crear un nuevo fichero llamado `fichero8` que cuelgue del directorio `dir3` y cuyo contenido sea el mismo que el del fichero `fichero1` (no utilice un editor)
  2. Borrar todos los ficheros cuyo nombre comience por `fich` y se encuentren en el directorio `dir2`
- k) Cambie su directorio de trabajo actual a su directorio personal y compruebe con la orden `ls -R` o con la orden `tree` que los cambios realizados se han ejecutado correctamente.

### Entrega de resultados, fichero de texto con la historia

Grabe la historia de todas las órdenes que ha ejecutado para resolver este ejercicio:

```
$ history -w Grupo-Apell11-Apell12-L2-1.txt
```

Compruebe con `less` el contenido del fichero generado y súbalo a Moodle durante la sesión presencial de laboratorio. Asegúrese de que el fichero que sube es el correcto (que no esté vacío o tenga otros contenidos).

**Aquellos ficheros que se suban a Moodle y no respeten estrictamente el formato del nombre, se ignorarán y por tanto no se evaluará la entrega correspondiente.**

## 8. Ejercicio a realizar sobre JavaScript

Se trata en este ejercicio de escribir un programa JavaScript que **muestre por consola la tabla completa de unidades binarias** que van desde Kibi hasta Yobi. Para ello, se definirá una función de nombre "unidad" que transforme las potencias de 2 en el nombre de prefijo correspondiente. A la función se le pasará en un parámetro el exponente y devolverá las letras que indican el prefijo

asociado, es decir:

10 se transformará en “Kibi”  
20 se transformará en “Mebi”  
30 se transformará en “Gibi”  
40 se transformará en “Tebi”  
50 se transformará en “Pebi”  
60 se transformará en “Exbi”  
70 se transformará en “Zebi”  
80 se transformará en “Yobi”

La función deberá realizarse utilizando sentencias `if/else` encadenadas para determinar el valor de retorno y deberá dar el resultado indicado al invocarse en la forma que se muestra a continuación como ejemplo:

```
console.log("1 x 2^10 Bytes son 1 " + unidad(10) + "bytes");  
=> "1 x 2^10 Bytes son 1 Kibibytes"  
console.log("1 x 2^20 Bytes son 1 " + unidad(20) + "bytes");  
=> "1 x 2^20 Bytes son 1 Mebibytes"  
console.log("1 x 2^30 Bytes son 1 " + unidad(30) + "bytes");  
=> "1 x 2^30 Bytes son 1 Gibibytes"  
...
```

El programa también deberá mostrar en la consola, como cabecera de la tabla, **su nombre, apellidos y grupo** al que pertenece.

Suba a Moodle un fichero con el programa JavaScript al que habrá puesto el siguiente nombre:

**Grupo-Apell11-Apell12-L2.js**

## 8. Resumen

Orden	Uso
<code>mkdir &lt;dir&gt;</code>	Crea el directorio <dir>
<code>cd &lt;dir&gt;</code>	Cambia el directorio de trabajo actual a <dir>
<code>cd</code>	Cambia el directorio de trabajo actual al directorio personal
<code>cd ~</code>	Cambia el directorio de trabajo actual al directorio personal
<code>cd ..</code>	Cambia el directorio de trabajo actual al directorio padre
<code>cd /</code>	Cambia el directorio de trabajo actual al directorio raíz
<code>pwd</code>	Saca la ruta absoluta del directorio de trabajo actual
<code>rmdir &lt;dir&gt;</code>	Borra un directorio vacío
<code>rm &lt;dir&gt;</code>	Opción: <code>-r</code> : borrar de forma recursiva todos los ficheros y directorios que cuelgan del directorio a borrar Opción: <code>-i</code> : pide confirmación antes de borrar
<code>tree</code>	Lista los ficheros, directorios y contenidos de los directorios recursivamente en forma de árbol