

## Laboratorio Sesión 01: Introducción a Linux

### Objetivo

El objetivo de esta sesión de laboratorio es introducir al alumno en el uso del Sistema Operativo Linux. Se deja a la discreción del alumno realizar las pruebas que considere oportunas para asimilar los conceptos necesarios, así como consultar el manual de Linux siempre que lo considere necesario.

En las siguientes sesiones de laboratorio se irán presentando otros temas.

### Manuales

Una de las características más interesantes de Linux es que dispone de una versión electrónica del manual accesible desde cualquier sesión de trabajo. El manual está dividido en secciones.

El comando **man** es el que utilizaremos para acceder a los manuales. Este comando está parametrizado con la palabra clave sobre la que queremos buscar información. Por ejemplo, **man ls** nos dará información sobre el comando **ls**.

La información mostrada por el comando **man** está estructurada en partes. La forma de estructurar la información depende de la sección involucrada. Por ejemplo, si hacemos **man ls** aparece la siguiente información:

```
prompt% man ls
LS(1)                                FSF                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...
    ...

DESCRIPTION
    ...

AUTHOR
    ...

REPORTING BUGS
    ...

COPYRIGHT
    ...

SEE ALSO
    ...
```

De esta estructura podemos destacar lo siguiente:

- El número que aparece en la primera línea "LS(1)" indica la sección que estamos consultando.
- NAME, muestra información muy resumida acerca de la palabra clave.
- SYNOPSIS, presenta la interficie. En el caso de un comando, son las opciones y los parámetros. Para una llamada al sistema, son los parámetros, tipo del resultado, ficheros de cabecera. También aparece un listado de las posibles opciones.
- DESCRIPTION, detalla cuál es el funcionamiento del comando o llamada al sistema.
- SEE ALSO, referencia otras páginas del manual relacionadas con ésta.

Una de las opciones del comando **man** es el *número de sección* que queremos consultar. Es muy útil en el caso de que una misma palabra clave aparezca en varias secciones. Por ejemplo, **man 1 read** nos daría información sobre el comando **read**, pero **man 2 read** nos daría información sobre la llamada al sistema **read**.

Para obtener información sobre **man**, simplemente hay que consultar el manual: **man man**.

Existen otros comandos que nos facilitan la búsqueda de información en el manual: **whatis** y **apropos**. Ejecutando los comandos **man finger**, **whatis finger** y **apropos finger** se puede comprender su funcionamiento.

## Gestión de Ficheros

Los ficheros en Linux, desde el punto de vista del S.O., se consideran como una secuencia de bytes sin formato. Los usuarios identifican el fichero mediante un nombre. Un nombre válido de fichero es una secuencia de caracteres de longitud inferior a 256. Es muy conveniente que no contenga ningún carácter especial del shell (`? * ` ` / \`). En principio, no se distingue entre nombre y extensión, aunque en muchos casos se utilizan extensiones para distinguir rápidamente el tipo de fichero. Un caso especial son los ficheros cuyo primer carácter del nombre es un punto (`.`). Estos ficheros son ocultos y no aparecen por defecto al hacer un listado de ficheros. Nombres válidos de ficheros son:

```
mis.datos.txt
.login
DirDocs
```

Es importante resaltar que Linux (al igual que C) distingue entre mayúsculas y minúsculas. Es decir, `Datos.txt` y `datos.txt` son ficheros diferentes.

En Linux existen varios tipos de ficheros:

- Ficheros de datos: texto, programas ejecutables, ...
- Directorios
- Dispositivos físicos: discos, terminales, ...
- ...

## Estructura Jerárquica del Sistema de Ficheros

Los ficheros se organizan de manera jerárquica de modo que un fichero de tipo directorio contiene cualquier tipo de fichero, e incluso nuevos ficheros de tipo directorio. Esta jerarquía está estructurada según una serie de convenios.

El directorio raíz (`root`) es el directorio de nivel superior de la jerarquía y viene representado por una barra (`/`). A partir de ahí existen unos directorios que aparecen en cualquier instalación Linux con las siguientes características:

- El directorio `/bin` contiene las utilidades y comandos más empleados en el sistema.
- En el directorio `/etc` se encuentran ficheros con datos esenciales para mantener el sistema.
- El directorio `/tmp` sirve para almacenar ficheros temporales. Los comandos y utilidades del sistema, utilizan este directorio regularmente. Los ficheros almacenados en este directorio pueden ser borrados en cualquier momento.
- En el directorio `/lib` se encuentran las librerías de programas que se pueden montar con programas del usuario.
- El directorio `/dev` contiene ficheros de tipo dispositivo: impresoras, discos, terminales, ...

Para identificar un fichero podemos utilizar su nombre absoluto. Este nombre incluye el camino que hay que seguir para llegar al fichero desde el directorio raíz. Los sucesivos directorios se separan por barras (`/`). Un ejemplo de fichero podría ser:

```
/homeA/cap/ac/agustin/PracticasEC2/Assembler/SumVectorInt.s
```

El nombre absoluto es poco práctico, ya que el sistema de ficheros puede tener muchos niveles. También es posible identificar los ficheros mediante nombres relativos. Para ello, a toda sesión de trabajo se le asocia un *directorio de trabajo* (*working directory*). El nombre relativo sólo ha de incluir el camino a seguir desde el directorio de trabajo hasta el fichero. Por ejemplo, si el directorio de trabajo es `"/homeA/cap/ac/agustin/PracticasEC2"`, el nombre del fichero es `"Assembler/SumVectorInt.s"`. Notad que el nombre absoluto siempre empieza por el símbolo `"/"`.

En todos los directorios existen dos ficheros especiales: `."` y `".."`. El fichero `."` hace referencia al directorio actual. Es útil para referenciar el directorio actual sin necesidad de

conocer su nombre. El fichero `..` hace referencia al directorio padre del actual. Facilita la navegación por los directorios.

Al establecer una sesión de trabajo, se asocia como directorio de trabajo el *home directory* del usuario. El *home directory* de un usuario es un directorio del sistema de ficheros asignado al usuario y donde éste puede mantener sus ficheros. El *home directory* de un usuario no puede variar, pero el directorio actual es modificable durante la sesión de trabajo.

## Comandos de Navegación y gestión del sistema de ficheros

A continuación se muestran algunos de los comandos básicos de gestión del sistema de ficheros. Para más información se recomienda consultar el **man**.

comando	descripción	ejemplo
<b>pwd</b>	Muestra la ruta completa hasta el directorio actual.	
<b>cd</b> [nombre directorio]	Cambia al directorio especificado. Si no se indica ninguno, va al <i>home directory</i> del usuario.	cd / cd /bin
<b>mkdir</b> nombre	Crea un directorio con el nombre indicado.	mkdir Practicas
<b>rmdir</b> nombre	Borra el directorio indicado (ha de estar vacío).	rmdir EC2
<b>ls</b>	Lista el contenido de un directorio determinado	ls -l ls -la
<b>cp</b> fOrigen fDestino	Copia ficheros del origen al destino especificados	cp p1.c p2.c
<b>rm</b> fichero	Comando para borrar ficheros. ¡Atención! una vez borrado el fichero, no se puede recuperar.	rm p1.c
<b>mv</b> fOrigen fDestino	Mueve ficheros del origen al destino especificados. Se usa habitualmente para cambiar el nombre de un fichero. El fichero origen desaparece.	mv p2.c p1.c
<b>cat</b> fichero(s)	Muestra el contenido de los ficheros especificados.	cat p1.c
<b>more</b> fichero(s)	Muestra el contenido del fichero por pantallas. Para navegar por el contenido del fichero consultad el <b>man</b> .	more p1.c
<b>chmod</b> mod fichero	Cambia las protecciones de un fichero	chmod 660 p1.c

Vamos a explicar con más detalle dos de estos comandos: **ls** y **chmod**. El resto de comandos podéis (debéis) consultarlos en el manual.

### El comando ls

Para listar el contenido de un directorio utilizaremos el comando **ls**. Si utilizamos el comando **ls** con la opción **-la**, obtenemos lo siguiente:

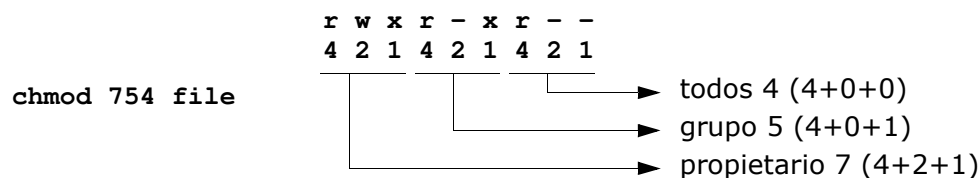
```
$> ls -la
total 376
drwx----- 3 agustin ac      2048 Mar  3 15:37 ./
drwx----- 5 agustin ac     1024 Mar  3 15:26 ../
-rw-r--r--  1 agustin ac       348 Mar  3 15:27 AddProd.c
-rw-r--r--  1 agustin ac     1129 Mar  3 15:27 AddProd.s
-rw-r--r--  1 agustin ac       151 Mar  3 15:27 CuentaLetra.c
-rw-r--r--  1 agustin ac       743 Mar  3 15:27 CuentaLetra.s
drwx--x--x  2 agustin ac        24 Mar  3 15:27 Datos/
-rw-r--r--  1 agustin ac       390 Mar  3 15:27 QuickSort.c
-rw-r--r--  1 agustin ac    2000 Mar  3 15:27 QuickSort.s
-rw-r--r--  1 agustin ac       210 Mar  3 15:27 SumVector.c
-rw-r----- 1 agustin ac    2008 Mar  3 15:27 SumVector.o
-rw-r--r--  1 agustin ac       964 Mar  3 15:27 SumVector.s
-rw-r----- 1 agustin ac    1228 Mar  3 15:27 SumVectorInt.o
-rw-r--r--  1 agustin ac       442 Mar  3 15:27 SumVectorInt.s
-rwxr----- 1 agustin ac    14109 Mar  3 15:27 VectorMain*
-rwxr----- 1 agustin ac    13931 Mar  3 15:27 VectorSort*
```

El comando `ls -la` produce un listado de los ficheros del directorio con una serie de información adicional acerca de cada fichero. El significado de esta información es el siguiente:

- El primer carácter representa el tipo de fichero. Los valores más frecuentes son - (ficheros de texto, ejecutables, ...) y d (directorios).
- Los 9 siguientes caracteres se agrupan de 3 en 3 y codifican los derechos de acceso del propietario, de los usuarios del mismo grupo del propietario y de todos los usuarios en general. Las 3 letras indican: r derecho de lectura, w derecho de escritura y x derecho de ejecución.
- El número representa la cantidad de nombres diferentes que referencian el mismo fichero.
- El username del propietario del fichero.
- El grupo al que pertenece el usuario propietario del fichero.
- Tamaño del fichero en bytes.
- Fecha y hora de la última modificación del fichero.
- Finalmente, encontramos el nombre del fichero. En Linux el carácter "\*" que hay al final de algunos nombres de fichero, indica que esos ficheros son ejecutables (ver protecciones). Igualmente el carácter "/", indica que ese fichero es un directorio. Notad que esta información es redundante.

## El comando chmod

El comando `chmod` es utilizado para modificar las protecciones de un fichero. Los bits de permiso que se especifican en el comando `chmod` son 3 dígitos codificados en octal de la siguiente forma:



## El intérprete de comandos

Hasta el momento, sólo habéis interactuado con Linux mediante el intérprete de comandos (el *shell*). El shell puede verse como un intérprete que ejecuta las instrucciones (comandos) de un determinado lenguaje de programación que lee del terminal o de un fichero.

Cuando se introduce un comando, el shell busca el fichero ejecutable con ese mismo nombre y, si lo encuentra, crea un proceso para ejecutarlo.

## Metacaracteres

Los metacaracteres sirven de gran ayuda a la hora de designar ficheros. Supongamos que en el directorio actual tenemos los siguientes ficheros: `at.c`, `trabajo`, `ta`, `test`, `texto.txt`, `tb`, `t1`, `test.old`, `Test`, `.login`, `.test` y `.Test`.

- El asterisco (\*) puede sustituir a cualquier número de caracteres. Por ejemplo, `t*` se sustituiría por `trabajo`, `ta`, `test`, `texto.txt`, `tb`, `t1` y `test.old`.
- El interrogante (?) se sustituye por un único carácter. `t?` se sustituiría por `ta`, `tb` y `t1`.
- Los corchetes ([]) se sustituyen por cualquiera de los caracteres escritos entre ellos. Por ejemplo `[A-Z]*` se sustituiría por los ficheros que empiezan con una letra mayúscula y cualquier serie de caracteres detrás. En el ejemplo se sustituiría por `Test`.

Los metacaracteres son interpretados por el shell y proporcionan al comando la lista de ficheros correspondiente a la interpretación. Por ejemplo, el comando `cat t*` ejecutaría el comando `cat` con la lista de ficheros `trabajo`, `ta`, `test`, `texto.txt`, `tb`, `t1` y `test.old`.

## Entrada y Salida Estándar. Redireccionamiento

Otra característica esencial que ofrece el intérprete de comandos es la posibilidad de redireccionar la entrada y salida estándar de un programa. Todos los programas se comunican con el exterior a través de la *entrada estándar* y la *salida estándar*. Por defecto, la entrada y salida estándar es el terminal de trabajo.

Los símbolos que permiten redireccionar la entrada y salida estándar se muestran en la siguiente tabla.

símbolo	descripción	ejemplo
< file	El fichero especificado se convierte en la entrada estándar	./p < datos.txt
> file	La salida estándar se escribe en el fichero especificado borrando el contenido original del mismo.	ls -l > dir.txt
>> file	La salida estándar se escribe al final del fichero especificado. El contenido original no se pierde.	ls -l .. >> dir.txt
	Este símbolo se denomina pipe y conecta la salida y entrada estándar de dos programas	ls -l   more

## Filtros

Teniendo en cuenta la posibilidad de redireccionar la entrada o salida estándar entre los diferentes comandos, Linux dispone de una serie de utilidades que denominamos *filtros*. La misión de un filtro es leer la entrada estándar, procesarla y escribir el resultado en un salida estándar. En la tabla siguiente se muestran algunos de los filtros más comunes.

comando	descripción	ejemplo
<b>more</b>	Muestra la entrada página a página, siguiendo las ordenes del usuario. Es posible obtener ayuda pulsando la tecla h.	ls -l   more
<b>sort</b>	Ordena la entrada estándar y escribe el resultado por la salida estándar. Por defecto, ordena alfabéticamente.	sort datos.txt
<b>wc</b>	Cuenta las líneas, palabras y caracteres de la entrada estándar.	who   wc
<b>tee file</b>	Envía la entrada estándar hacia la salida estándar (igual que <b>cat</b> ), pero además escribe la entrada estándar en el fichero especificado.	ls -l   tee dir.txt
<b>grep cadena file(s)</b>	Busca la cadena de caracteres suministrada en los ficheros indicados. Si encuentra la cadena muestra el nombre del fichero y la línea donde aparece	ls -l   grep 2003
<b>head file</b>	Muestra las primeras líneas del fichero especificado.	ls   grep 2003   head
<b>tail file</b>	Muestra las últimas líneas del fichero especificado.	ls   grep 2002   tail

## Resultados de la sesión

Antes de acabar la sesión tenéis que haber hecho, por lo menos, lo siguiente:

- Navegar por el árbol de directorios.
- Crear un directorio.
- Borrar un directorio.
- Crear un fichero mediante la redirección de la salida estándar.
- Cambiar las protecciones de un fichero.
- Cambiar las protecciones de un directorio.
- Aplicar filtros.
- Utilizar una pipe.
- Utilizar metacaracteres.
- Hacer un listado de los ficheros de un directorio que no está en el directorio actual.
- Consultar el man (varias veces).