
Seminario de introducción al uso del sistema operativo UNIX

Michael González Harbour
Departamento de Electrónica y Computadores
Universidad de Cantabria

Introducción

El *sistema operativo* es un programa que:

- Controla el acceso a todos los *recursos del sistema* (memoria, teclado, pantalla, etc.)
- Implementa un *sistema de ficheros* para el almacenamiento de información en la memoria secundaria o masiva
- Tiene un *intérprete para las órdenes* que el usuario introduce por teclado (“*shell*”). Con estas órdenes podemos:
 - gestionar la información almacenada en el disco
 - ejecutar programas de aplicación

El sistema operativo se carga en la memoria del computador al encenderlo

El UNIX es un sistema operativo avanzado diseñado principalmente para estaciones de trabajo (“*workstations*”), aunque hoy en día existe para multitud de sistemas

- es multiusuario, multiproceso y protegido
- está estandarizado a nivel internacional (*POSIX*)
- hay versiones de libre distribución (*Linux, Free BSD, ...*)

El Windows-95/98/ME/XP es un sistema operativo diseñado para los computadores personales de IBM y compatibles

- es monousuario, multiproceso y semi-protegido
- pertenece a una empresa única (*Microsoft*)

El Windows NT/2000 es protegido y multiusuario (s. versión)

El sistema de ficheros

Los programas y la información no volátil se almacenan en el sistema de ficheros del computador, basado en memoria secundaria (discos, diskettes, CD-ROM, etc.)

La información se almacena en ficheros, que se identifican por un nombre, con el siguiente formato:

- cualquier secuencia de hasta 256 caracteres (excepto espacio en blanco, y “/”)

Ficheros y directorios

Los ficheros pueden ser:

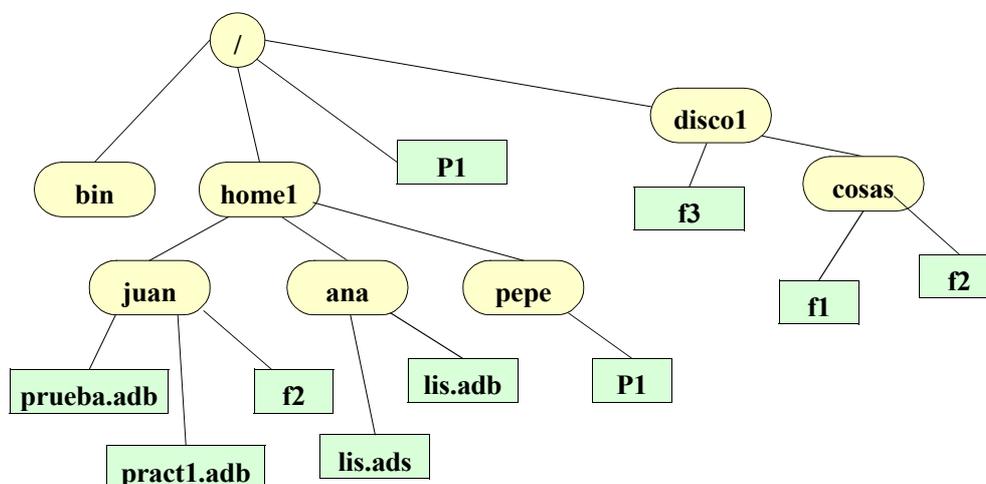
- ficheros de información (programas o datos)
- directorios, que a su vez contienen otros ficheros

Los ficheros y directorios se organizan con una estructura jerárquica, en forma de árbol

- La raíz del árbol se denomina: “/”
 - Nota: al revés que en windows: “\”
- El directorio actual se denomina “.”
- El padre de un directorio se llama “..”

Árboles de ficheros en UNIX

Los directorios dan lugar a una estructura en forma de un único árbol, con todas las unidades de memoria secundaria



Nombres de ficheros: UNIX

El directorio principal (raíz) se llama “/”

El nombre completo de un fichero (“*pathname*”) se compone de:

- nombres de los directorios de los que depende en el árbol de ficheros, separados por “/”
- nombre del fichero

Ejemplo: `/home1/juan/prueba.adb`

Son distintas las mayúsculas de las minúsculas

Directorio de trabajo

Para no escribir el nombre completo del fichero existe un directorio de trabajo:

- si el fichero está en el directorio de trabajo éste se puede omitir

Ejemplo: suponiendo que el directorio de trabajo es `/home1/juan`, son equivalentes:

`/home1/juan/prueba.adb`
`prueba.adb`

El directorio padre se representa por “..”

Ejemplos:

`../ana/lis.ads` `../../bin`

En ocasiones podemos referirnos globalmente a múltiples ficheros cuyos nombres se parecen. Para ello se usan los caracteres comodín:

- “?” puede ser sustituido por cualquier carácter
- “*” por cualquier secuencia de cero o más caracteres
- Ejemplos :

`/home1/juan/*.adb` (`prueba.adb` y `pract1.adb`)

`/home1/*` (`juan`, `ana` y `pepe`)

El sistema operativo dispone de un intérprete de las órdenes que se introducen por teclado (“*shell*”)

Cuando el intérprete está listo para recibir una orden, muestra el símbolo de preparado (“*prompt*”), generalmente con el directorio de trabajo o el nombre del computador. Ejemplo:

- `puesto031>`

Bajo el intérprete se puede:

- introducir una orden
- ejecutar un programa

UNIX	DOS	Función	Sintaxis
ls ls -l	DIR	Muestra una lista del contenido del directorio	ls -l [nombre-directorio]
cd	CD	Cambiar el directorio de trabajo	cd nombre-directorio
rm	DEL	Borrar un fichero	rm nombre-fichero
		Borrar varios ficheros	rm nombre-comodín
cp	COPY	Copiar un fichero en otro;	cp origen destino
		Copiar uno o varios ficheros en otro directorio	cp comodín [dir-destino]

Órdenes del sistema operativo (cont.)

UNIX	DOS	Función	Sintaxis
mv	MOVE	Mover un fichero a otro (es decir cambiarle el nombre);	mv fich-origen destino
		Mover uno o varios ficheros a otro directorio	mv comodín dir-destino
more	TYPE	Mostrar un fichero en pantalla	more nombre-fichero
mkdir	MKDIR	Crear un nuevo directorio	mkdir nombre-directorio
rmdir	RMDIR	Borrar un directorio vacío	rmdir nombre directorio
rm -r	DEL-TREE	Borrar un directorio y todos sus contenidos	rm -r nombre-directorio
lpr	PRINT	Imprimir un fichero	lpr nombre-fichero
man	HELP	Pedir info sobre una orden	man orden

Uso del disquete en LINUX

- Los nombres de ficheros siguen las convenciones del UNIX, usando “/” para separar directorios y nombres
- Se admiten nombres largos
- A los nombres de objetos del disquete se les antepone “a:”

Orden	Función	Sintaxis
mdir	Muestra una lista del contenido del directorio	mdir [nombre-directorio] mdir a:[nombre-directorio]
mcd	Cambiar el directorio de trabajo en el disquete	mcd nombre-directorio mcd a:nombre-directorio
mdel	Borrar un fichero	mdel nombre-fichero mdel a:nombre-fichero
	Borrar varios ficheros	mdel nombre-comodín mdel a:nombre-comodín

Uso del disquete en LINUX (cont.)

Orden	Función	Sintaxis
mcopy	Copiar un fichero de texto del disquete en el disco	mcopy -t a:origen destino
	Copiar un fichero de texto del disco en el disquete	mcopy -t origen a:destino
	Copiar un fichero binario del disquete en el disco	mcopy a:origen destino
	Copiar un fichero binario del disco en el disquete	mcopy origen a:destino
	Copiar uno o varios ficheros de texto en el disquete	mcopy -t comodín a:[dir-dest.]
	Copiar uno o varios ficheros de texto del disquete	mcopy -t a:comodín dir-destino

Ejecución de programas

Para ejecutar un programa basta escribir su nombre bajo el intérprete de órdenes:

- nombre del fichero ejecutable; por ejemplo si se ha creado el programa `practil`:

```
/home1/juan/practil  
practil
```

Ruta de acceso (“PATH”)

Si el programa no está en el directorio de trabajo, se puede:

- escribir su nombre completo:
`/home1/juan/practil`
- o escribir su nombre simple si el directorio en el que está pertenece a la ruta de acceso (“*path*”) del intérprete de órdenes

La ruta de acceso se puede consultar con la orden:

```
echo $PATH
```

Teclas especiales

Interrumpir la ejecución de un programa:

```
<ctrl>c
```

Detener la salida de datos en pantalla:

```
<ctrl>s y luego <ctrl>q
```

Reiniciar el computador, desde el terminal alfanumérico (no gráfico), cuando todo falla:

```
<ctrl><alt><supr> o <ctr><alt><del>
```

Aspectos avanzados del intérprete de órdenes

Redirección de entrada/salida:

- la entrada de teclado o la salida a pantalla se pueden redirigir a ficheros en disco
- por ejemplo la siguiente orden almacena en el fichero `lista.dat` la información de la orden `ls -l`
`ls -l > lista.dat`

Historial de órdenes:

- para avanzar atrás o adelante en el historial de órdenes pulsar `↑` o `↓`
- la orden se puede editar con `←` y `→`

Aspectos avanzados del intérprete de órdenes (cont.)

Secuencias de órdenes, para redirigir la salida de una orden a un programa que la transforma:

- “**more**” muestra el texto pantalla a pantalla:
`ll | more`
- en LINUX, “**less**” muestra también el texto, pudiéndose recorrer hacia arriba o hacia abajo (se sale con “**q**”):
`ll | less`
`less fich.ads`

Completar nombres de fichero:

- pulsar los primeros caracteres del nombre y luego
`<TAB>` o `<ESC><ESC>`