

Bloque III. Herramientas

Capítulo 1. Uso de sistemas operativos

- Conceptos básicos
- Sistemas operativos comunes
- Arranque y finalización de sesiones
- El sistema de ficheros
- El intérprete de órdenes
- Ejecución de programas
- El gestor gráfico de ficheros
- Aspectos avanzados

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**”) o ratón y pantalla (gestor de ventanas). Con estas herramientas podemos:
 - gestionar la información almacenada en el disco
 - ejecutar programas de aplicación

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

Sistemas operativos comunes

El UNIX es un sistema operativo avanzado diseñado originalmente 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 98 era un sistema operativo diseñado para los computadores personales de IBM y compatibles

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

Windows NT/XP/Vista es protegido y multiusuario (según versión)

Arranque del sistema operativo

Al encender el computador se arranca el sistema operativo predeterminado

Si existen varios sistemas operativos, se da la opción de elegir

En nuestro laboratorio se dan varias opciones, seleccionables con el ratón

Para nuestra asignatura elegir:

- **Arrancar Linux**

Inicio de la sesión en los puestos LINUX



Para entrar, proporcionar el *nombre de usuario* y luego la *palabra clave*.

Para salir:

- Elegir en el menú “**Acciones**” de la parte superior de la pantalla, la opción “**Terminar la sesión**”
- luego, “**Terminar la sesión**”

Para apagar el computador



Nunca apagarlo directamente, pues se podrían estropear los datos del disco

Windows: En el botón de “**Inicio**”, seleccionar “**Apagar el sistema**”, y luego “**apagar**” o “**reiniciar**”, según se desee

Linux: Se puede hacer de dos formas:

- Al abandonar la sesión elegir en el menú “**Acciones**” de la parte superior de la pantalla, la opción “**Terminar sesión**” y, luego, “**Apagar**”
- En la pantalla de inicio de sesión, elegir el menú “**Acciones**”, y luego “**Apagar**”

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 “/” en UNIX y “\” en Windows)
- se suele identificar el tipo de fichero con una **extensión**:
 - programa java: **nombre.java**
 - fotografía jpeg: **nombre.jpg** o **nombre.jpeg**
 - página web: **nombre.html** o **nombre.htm**
 - etc.

Ficheros y directorios

Los ficheros pueden ser:

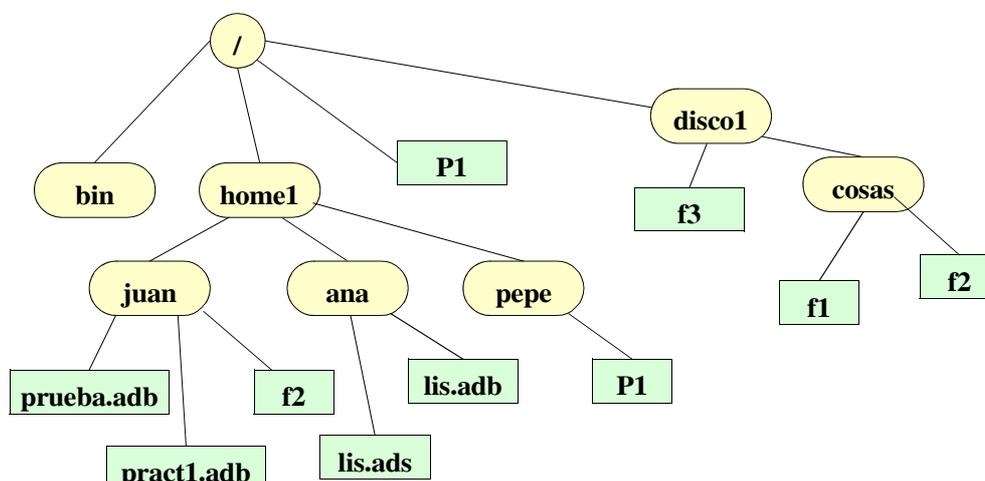
- ficheros de información (programas o datos)
- directorios, que a su vez contienen otros ficheros

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

- La **raíz** del árbol se denomina en UNIX: “/”
 - Nota: al revés que en Windows: “\”

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 él, se puede omitir su nombre
- El **directorio de trabajo** se denomina “.”
- El **padre** de un directorio se llama “..”

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

/home1/juan/prueba.adb
prueba.adb

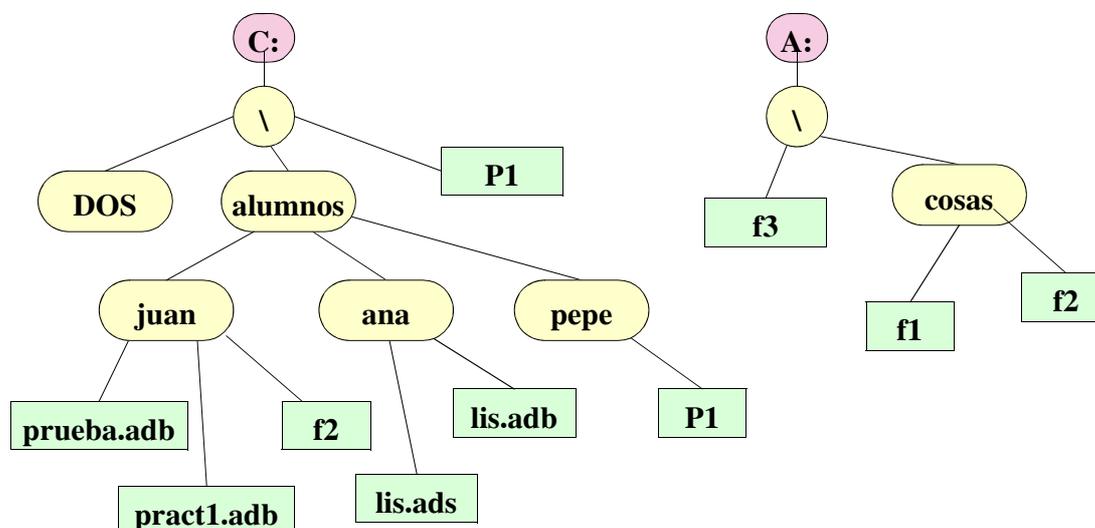
Ejemplos:

../ana/lis.ads

../../bin

Árboles de ficheros en Windows

Los directorios dan lugar a una estructura en forma de árbol, con un árbol por cada unidad de memoria secundaria



Nombres completos de ficheros: Windows



Cada unidad de almacenamiento secundario (disco, disquete, CD-ROM) tiene como nombre una letra, seguida de “:”

- disquete: **a:** disco duro: **c:**

El directorio principal (raíz) se llama “\”

El nombre completo de un fichero (“pathname”) tiene:

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

En los nombres no se distinguen mayúsculas de minúsculas

Unidad y directorio de trabajo



Para no escribir el nombre completo del fichero existen:

- una unidad de trabajo: si el fichero está en la unidad de trabajo ésta se puede omitir
- un directorio de trabajo: idem.

Ejemplo: suponiendo que la unidad de trabajo es **c:** y el directorio de trabajo es **\alumnos\juan**, son equivalentes:

c:\alumnos\juan\prueba.adb

\alumnos\juan\prueba.adb

prueba.adb

El directorio padre se representa por “..”; ejemplos:

..\ana\lis.ads

..\..\dos

Caracteres comodín

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 UNIX:
 - `/home1/juan/*.adb` (prueba.adb y pract1.adb)
 - `/home1/*` (juan, ana y pepe)
- Ejemplos Windows:
 - `c:\alumnos\juan*.adb` (prueba.adb y pract1.adb)
 - `c:\alumnos*.*` (juan, ana y pepe)

Intérprete de órdenes

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

- llamado “*Símbolo del sistema*” en Windows (ver accesorios)
- llamado “*sh*”, “*cs*h”, “*ba*sh”, ..., en Unix

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:

- UNIX: `pc37 usr>`
- Windows: `c:\ALUMNOS\JUAN>`

Bajo el intérprete se puede:

- introducir una orden o ejecutar un programa

Órdenes comunes: ls

Orden	<code>ls</code>	Opciones	<code>-l</code>
Función	Muestra una lista del contenido del directorio		
Sintaxis	<code>ls nombreDirectorio*</code> <code>ls -l nombreDirectorio</code>		
Ejemplos			
<code>ls</code>	Muestra el contenido del directorio actual		
<code>ls -l prueba</code>	Muestra el contenido del directorio <code>prueba</code>		

* Este color representa un argumento opcional: `nombreDirectorio`

Órdenes comunes: cd

Orden	<code>cd</code>		
Función	Cambiar el directorio de trabajo		
Sintaxis	<code>cd nombreDirectorio</code>		
Ejemplos			
<code>cd</code>	Hace que el directorio de trabajo sea el inicial del usuario		
<code>cd prueba</code>	Hace que el directorio de trabajo sea <code>prueba</code>		

Órdenes comunes: cp

Orden	<code>cp</code>	Opciones	<code>-r</code>
Función	Copia ficheros		
Sintaxis	<code>cp ficheroOrigen ficheroDestino</code> <code>cp ficheroOrigen dirDestino</code> <code>cp -r dirOrigen dirDestino</code>		
Ejemplos			
<code>cp pepe juan</code>	Copia el fichero <code>pepe</code> en otro llamado <code>juan</code>		
<code>cp Hola.* proyecto1</code>	Copia los ficheros que empiecen por <code>Hola.</code> , con el mismo nombre, en el directorio llamado <code>proyecto1</code>		
<code>cp -r proyecto1 proyecto2</code>	Copia recursivamente (con sus subdirectorios) el directorio <code>proyecto1</code> , en el directorio <code>proyecto2</code>		

Órdenes comunes: rm

Orden	<code>rm</code>	Opciones	<code>-r</code>
Función	Borra ficheros		
Sintaxis	<code>rm nombreFichero</code> <code>rm -r nombreDirectorio</code>		
Ejemplos			
<code>rm pepe</code>	Borra el fichero <code>pepe</code>		
<code>rm *.java</code>	Borra todos los ficheros del directorio de trabajo que acaben en <code>.java</code>		
<code>rm -r proyecto1</code>	Borra el directorio <code>proyecto1</code> y todos sus contenidos, recursivamente		

Órdenes comunes: mkdir

Orden	<code>mkdir</code>		
Función	Crea un directorio		
Sintaxis	<code>mkdir nombreDirectorio</code>		
Ejemplos			
<code>mkdir proyecto3</code>	Crea un directorio vacío llamado <code>proyecto3</code>		

Órdenes comunes: mv

Orden	<code>mv</code>	Opciones	
Función	Cambia de nombre un fichero o lo mueve a otro directorio		
Sintaxis	<code>mv nombreViejo nombreNuevo</code> <code>mv nombreFichero nombreDirectorio</code> <code>mv nombreDirectorio nombreDirectorio</code>		
Ejemplos			
<code>mv pepe juan</code>	Cambia el nombre del fichero <code>pepe</code> al nuevo <code>juan</code>		
<code>mv juan proyecto1</code>	Mueve el fichero <code>juan</code> al interior del directorio <code>proyecto1</code>		
<code>mv proyecto1 proyecto3</code>	Mueve recursivamente el directorio <code>proyecto1</code> al interior del directorio <code>proyecto3</code>		

Órdenes comunes: less

Orden	<code>less</code>		
Función	Muestra en pantalla un fichero de texto Permite la navegación arriba y abajo (<i>flechas</i>) Salir con q (<i>quit</i>)		
Sintaxis	<code>less nombreFichero</code>		
Ejemplos			
	<code>less pepe.txt</code>	Muestra el contenido del fichero	<code>pepe.txt</code>

Órdenes comunes: rmdir

Orden	<code>rmdir</code>		
Función	Borra un directorio vacío		
Sintaxis	<code>rmdir nombreDirectorio</code>		
Ejemplos			
	<code>rmdir proyecto1</code>	Borra el directorio <code>proyecto1</code> , si está vacío	

Otras órdenes

Orden	Función	Sintaxis
<code>lpr</code>	Imprimir un fichero	<code>lpr nombreFichero</code>
<code>man</code>	Pedir info sobre una orden	<code>man orden</code>
<code>passwd</code>	Cambiar la palabra clave	<code>passwd</code>

Ejecución de programas

Para ejecutar un programa basta escribir su nombre bajo el intérprete de órdenes (sin extensión):

- UNIX: nombre del fichero ejecutable; por ejemplo si se ha creado el programa `pract1`:

`/home1/juan/pract1`

o también:

`pract1`

- Windows: nombre sin extensión; por ejemplo si se ha creado el programa `pract1.exe`:

`c:\alumnos\juan\pract1`

o también:

`pract1`

Ruta de acceso ("PATH")

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

- escribir su nombre completo:
`/home1/juan/practi1`
`c:\alumnos\juan\practi1`
- 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:

- UNIX: `echo $PATH`
- Windows: `PATH`

Gestor gráfico de ficheros

Permite operar con los ficheros de forma cómoda

- ver los ficheros de una carpeta o directorio
- seleccionar un fichero `<botón izquierdo ratón>`
- seleccionar varios ficheros
 - uno más: `<Ctrl> + <botón izq. ratón>`
 - todos entre el actual y el anterior: `<Mayus> + <botón izq. ratón>`
- mover los ficheros de un directorio a otro: arrastrar con `<botón izq. ratón>`
- copiar los ficheros de un directorio a otro: arrastrar con `<Ctrl> + <botón izq. ratón>`
- copiar y luego pegar: con `<botón derecho ratón>`

Desde el gestor gráfico de ficheros

- es imprescindible acordarse de **desmontar** el dispositivo
 - seleccionar "eject" o "expulsar" con el botón derecho del ratón
- en Windows no siempre es preciso
- en Linux es imprescindible
 - si no, se corrompe la información

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 (LINUX):

- para avanzar atrás o adelante en el historial de órdenes pulsar \uparrow
o \downarrow
- la orden se puede editar con \leftarrow y \rightarrow

Aspectos avanzados del intérprete de órdenes (cont.)



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

- “**less**” muestra el texto pantalla a pantalla:
`ls -l | less`

Completar nombres de fichero (LINUX):

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

Interrumpir la ejecución de un programa:

`<ctrl>c`

