



Introducción a UNIX

Departamento de Lenguajes y
Sistemas Informáticos

Introducción



1. Introducción a UNIX
2. Sistema de Ayuda
3. El intérprete de comandos
 1. Sintaxis de los comandos
 2. Variables de entorno
 3. Redireccionamiento de E/S
 4. Tuberías (Pipes)
 5. Programación Shell
4. Comandos de UNIX
 1. Comandos para el manejo de ficheros
 2. Comandos para el manejo de la Entrada/Salida
5. Secure SHell (SSH)
6. File Transfer Protocol (FTP)

Introducción a UNIX



- El sistema operativo UNIX tiene su origen en 1969. Escrito en ensamblador para PDP-7.
- En 1972 Ritchie y Thompson reescriben el núcleo en C.
- En 1974 fue introducido en las universidades con fines educacionales.
- Hoy en día la palabra UNIX se emplea para referirse a distintos conceptos:
 - UNIX como familia de sistemas operativos que comparten unos criterios de diseño e interoperabilidad común. (Estándar POSIX, IEEE 1003).
 - UNIX como el sistema operativo original.
 - UNIX como marca comercial.

Introducción a UNIX

- Características de UNIX:
 - Portable
 - Flexible
 - Robusto
 - Multiusuario
 - Multiprogramado

- **Linux** es un sistema operativo tipo UNIX orientado a implementar el estándar POSIX y que utiliza primordialmente filosofía y metodologías libres.

UNIX ≠ Linux

Introducción a UNIX

- UNIX actuales
 - FreeBSD (<http://www.freebsd.org>)
 - OpenBSD (<http://www.openbsd.org>)
 - NetBSD (<http://www.netbsd.org>)
 - OpenSolaris (<http://www.opensolaris.org/os/>)
 - HP-UX (HP)
 - AIX (IBM)
 - Xenix (Microsoft)

- Distribuciones Linux
 - Debian (<http://www.debian.org/index.es.html>)
 - Fedora (<http://fedora.redhat.com/>)
 - Ubuntu (<http://www.ubuntu-es.org/>)
 - SUSE (<http://www.novell.com/linux/>)
 - Red HAT (<http://www.redhat.es/>)
 - Guadalinux (<http://www.guadalinux.org>)

Sistema de ayuda

- ***man*** es el sistema estándar de ayuda en UNIX.
 - man [seccion] materia
 - man -k clave
 - apropos clave

El interprete de comandos

- Los intérpretes de comandos o shells interpretan las ordenes introducidas por el usuario y las convierte en instrucciones para el sistema operativo.

- Un Shell se caracteriza, entre otros, por:
 - Comandos propios del Shell.
 - Soporte de edición.
 - Historial de comandos.
 - Programación en Shell Script.
 - Configurable.

El interprete de comandos

Tipo de Shell	Shell estándar	Clones libres
AT&T Bourne Shell	sh	ash, bash, bash2
Berkeley "C" shell	csh	tcsh
AT&T Korn shell	ksh	Pdksh, zsh
Otros	-	esh, gush, nwsh

- Es posible instalar y trabajar con tantos intérpretes de comandos como deseemos.

Shells por defecto...

- Murillo: ksh
- Fedora: bash
- Ubuntu: bash

Sintaxis de los comandos

- > comando [opciones] [arg₁ arg₂...arg_n]
 - Comando: Identifica el programa a ejecutar (Ej. rm)
 - Opciones: Especifica determinadas opciones de ejecución (Ej.: -r)
 - [arg₁ arg₂ ... arg_n]: Parámetros de entrada (Ej. fichero.c)

- Pueden introducirse varios comandos en una misma línea:
 - Ejecución incondicional: ls ; ps -ef
 - Ejecución condicional: ls hola.txt > /dev/null && echo "Existe el fichero"

- Es posible usar caracteres comodín (*, ? , [xyz]) para hacer más flexible la ejecución de comandos.

Caracteres comodín ≠ Expresiones regulares

Variables de entorno

- Una **variable de entorno** es un nombre asociado a una cadena de caracteres.
 - Variable = Valor

- Pueden ser definidas por el usuario o por el propio shell.
 - PATH, HOME, SHELL, DISPLAY, etc.

- Las variables definidas por el usuario sólo tienen valor a nivel local, es decir, en el shell actual. Para que dichas variables puedan ser accesibles por los procesos hijos es necesario exportarlas.
 - export Variable
 - export Variable = Valor

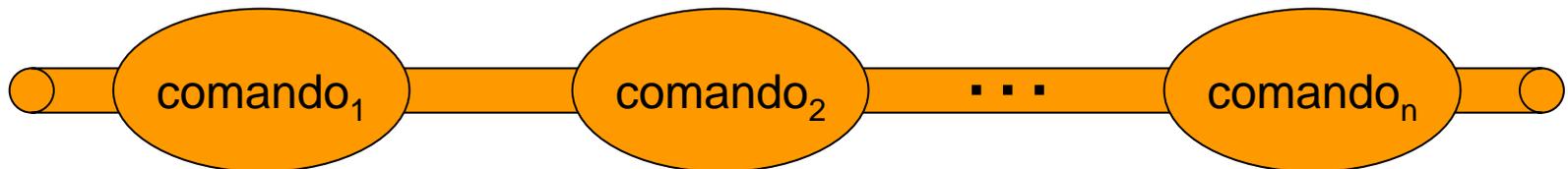
Redirección de E/S



- Todos los programas tienen por defecto una entrada estándar (teclado) y dos salidas: la salida estándar (pantalla) y la salida de error (pantalla).
- Para sustituir la entrada/salida estándar de un programa por un fichero podemos usar los caracteres '<' y '>'.
El carácter de redirección de salida, '>', destruye el fichero al cual apunta. Si en lugar de machacarlo, deseemos anexar la información debemos usar el doble carácter '>>'.

Tuberías

- Una tubería o pipe es una combinación de varios comandos que se ejecutan simultáneamente, donde el resultado del primero se envía a la entrada del siguiente. Este tipo de comunicación entre procesos se especifica mediante el carácter barra vertical, '|'.
 - `> comando1 | comando2 | ... | comandon`

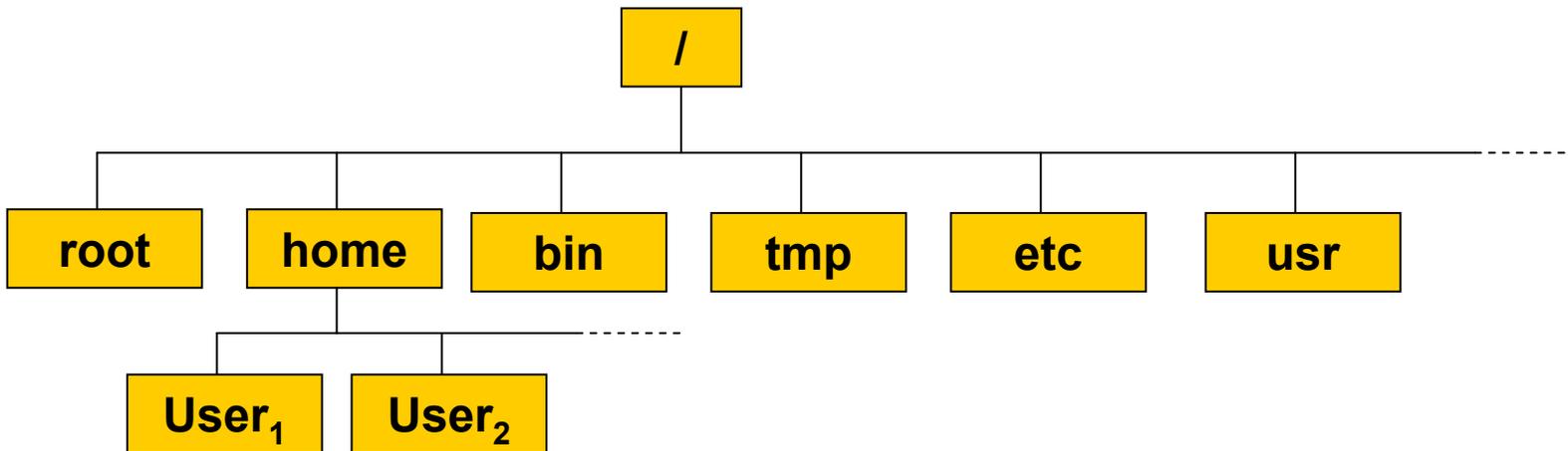


Programación Shell

- Consiste en la creación de scripts creados para ser interpretado por un shell determinado.
- Permite automatizar tareas complejas o comandos repetitivos.
- Usos comunes: manipulación de ficheros, ejecución de programas, monitorización, etc.
- En muchos casos suelen programarse para que se ejecuten automáticamente en un determinado momento, algo de gran utilidad para los administradores de sistemas.
- Similar a los .bat de windows pero mucho más potente.

Comandos de UNIX

- Estructura de directorios de UNIX



- **Ruta absoluta:** Toma como referencia la raíz de sistema (/).
- **Ruta relativa:** Toma como referencia el directorio de trabajo actual.

Comandos de UNIX

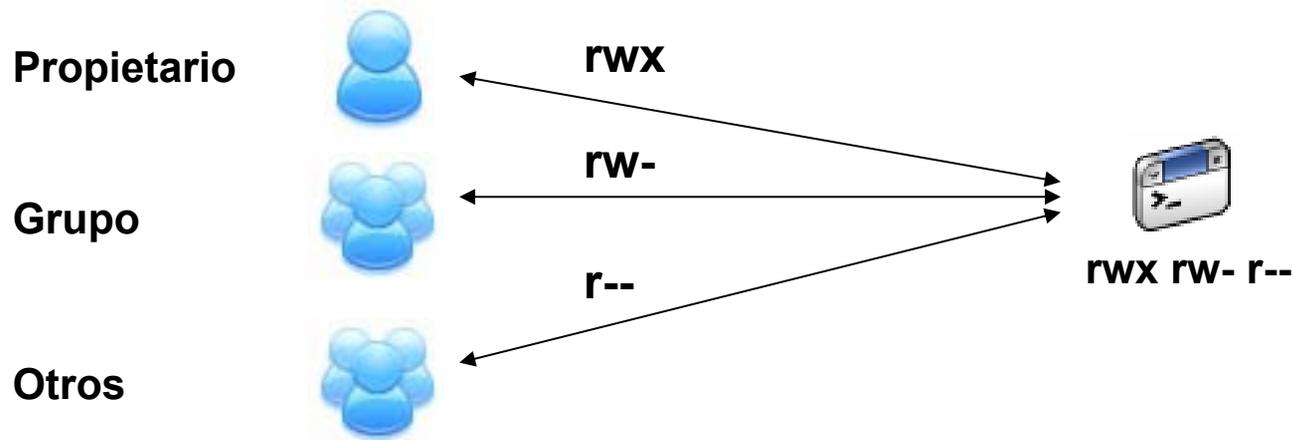
- Algunos conceptos básicos
 - `./` → Directorio actual.
 - `../` → Directorio padre.
 - `pwd` → Imprime por pantalla la ruta actual.
 - `$HOME`, `~` → Directorio home del usuario.
 - `cd path/to/dir` → Cambia el directorio de trabajo actual por el especificado.
 - `cd ..` → Cambia el directorio de trabajo actual por el directorio padre.
 - `mkdir dir` → Crea un directorio.
 - `rmdir dir` → Elimina un directorio vacío.
 - `rm -r` → Elimina un directorio y todo su contenido

Comandos de UNIX

- En UNIX todos los ficheros y directorios tienen asignado privilegios o permisos. Existen tres tipos de permisos:
 - Permiso de lectura (r)
 - Permiso de escritura (w)
 - Permiso de ejecución (x)

- A cada fichero o directorio se le asignan permisos asociados a cada uno de los grupos de usuarios que pueden hacer uso del mismo. UNIX divide los usuarios del sistema en tres grupos:
 - Propietario del fichero o directorio.
 - Grupo al que pertenece el propietario del fichero o directorio.
 - Todos los demás.

Comandos de UNIX



	Fichero Regular	Directorio
r	Leer	Ver contenido (ls)
w	Modificar	Añadir/Borrar
x	Ejecutar	Acceso (cd)

Comandos de UNIX

- Comandos básicos para la manipulación de ficheros
 - ls
 - cp
 - mv
 - chmod

Comandos de UNIX



- **ls [opciones] [ruta]:** Lista el contenido de un directorio o árbol de directorios.
 - -a: Muestra también los ficheros ocultos
 - -l: Muestra la salida en formato largo. Para cada fichero o directorio se muestra:
 - Tipo (d: Directorio, -: Fichero, l: Enlace)
 - Permisos
 - Número de enlaces que apuntan al fichero o directorio
 - Propietario
 - Grupo
 - Espacio en disco
 - Fecha de creación/modificación
 - Nombre del archivo

Comandos de UNIX

- **cp [opciones] origen₁ [origen₂ ... origen_n] destino:** Permite copiar ficheros y directorios desde una ruta origen a una ruta destino.
 - -r: Copiar subdirectorios de forma recurrente.
 - -f: No avisar si la operación machaca ficheros destino

- **mv [opciones] origen₁ [origen₂ ... origen_n] destino:** Permite mover/renombrar ficheros y directorios desde una ruta origen a una ruta destino.
 - -f: No avisar si la operación machaca ficheros destino.

Comandos de UNIX

- **chmod:** Se emplea para establecer las autorizaciones de acceso (permisos) de un archivo o directorio. Pueden emplearse dos sintaxis distintas para la ejecución del comando:
 - **Representación Octal.** Cada tipo de permiso tiene una representación numérica (r=4, w=2, x=1). Así, el permiso final para un grupo de usuarios dado se representa mediante la suma del valor de los permisos para dicho grupo.
 - Ej. `chmod 755 file.c` (Propietario: rwx , Grupo: r-x, Otros: r-x)
 - **Representación simbólica.** u = user, g = group, o = others, a = all, + = Añadir permisos, - = Quitar permisos
 - Ej. `chmod g+w file.c`

Comandos de UNIX

- Comandos para el manejo de la Entrada/Salida
 - cat
 - more
 - echo / print
 - read
 - grep
 - sort

Comandos de UNIX



- **cat [opciones] [fichero₁, fichero₂, ..., fichero_n]**: Muestra el contenido de uno o varios ficheros en la salida estándar.
 - -b: Enumera todas las líneas que no estén en blanco, a partir de 1.
 - -n: Enumera todas las líneas tanto las que están en blanco como las que no.

Comandos de UNIX



- **more [opciones] fichero₁ [fichero₂, ..., fichero_n]:** Permite visualizar el contenido de un fichero de texto página a página. Suele utilizarse combinado con otros comandos a través de tuberías.
 - -n: Número de líneas que se muestra por cada página.
 - +n: Comienza a mostrar el documento a partir de la línea n-ésima.
 - +patrón: Busca la primera ocurrencia en el texto del patrón y comienza a mostrar la salida a partir de dicho punto.

- **less [opciones] fichero₁ [fichero₂, ..., fichero_n]:** Variante de more que permite navegar en el texto.

Comandos de UNIX

- **echo cadena:** Muestran en la salida una cadena de texto dada. Tras la cadena mostrada se produce un salto de línea.
 - -n: Indica que no se debe producir el salto de línea a continuación de la cadena

- **printf cadena:** Muestran en la salida una cadena de texto dada empleando la sintaxis de C (Uso de `\n` para indicar salto de línea).

- Ojo con las comillas:
 - `> echo "El shell actual es $SHELL"` → El shell actual es `/bin/bash`
 - `> echo 'El shell actual es $SHELL'` → El shell actual es `$SHELL`

Comandos de UNIX

- **read variable₁ [variable₂, ..., variable_n]:** Lee de la entrada estándar el valor de una o más variables. El comando lee una línea completa de texto asignando una palabra (cualquier cadena delimitada por espacios en blanco o tabulador) a cada variable.

```
> read a b c
```

```
> En un lugar de la mancha
```

```
a = En
```

```
b = un
```

```
c = lugar de la mancha
```

Comandos de UNIX



- **grep [opciones] patrón [fichero₁, fichero₂, ..., fichero_n]:** Toma como entrada uno o más ficheros y muestra en la salida estándar todas aquellas líneas en las que se encuentre un patrón dado.
 - -i: No distingue entre mayúsculas y minúsculas.
 - -l: Muestra sólo una lista con los ficheros de la entrada que en algún lugar contienen el patrón.
 - -v: Muestra las líneas que no contienen el patrón.
 - -w: Requiere que el patrón coincida con una palabra completa.

Comandos de UNIX



□ Expresiones Regulares

- `.`: Cualquier carácter.
- `[c1..cn]`: Cualquier carácter de los situados entre corchetes.
- `[c1-cn]`: Cualquier carácter del rango.
- `[^c1-cn]`: Cualquier carácter que no esté en el rango especificado.
- `exp*`: 0 o más repeticiones de la expresión regular anterior, `exp`.
- `^exp`: La expresión regular, `exp`, debe aparecer al principio de la línea.
- `exp$`: La expresión regular, `exp`, debe aparecer al final de la línea.
- `\`: Elimina el significado especial del metacarácter colocado a continuación.

Comandos de UNIX



- **who [opciones]:** Proporciona información sobre los usuarios conectados a la máquina.
 - -m: who am I?
 - -q: Proporciona el nombre de los usuarios conectados e indica cuántos hay en total.
 - -u: Formato largo. Tras la hora de conexión, muestra el tiempo que el usuario lleva inactivo.

Comandos de UNIX



- **sort [opciones] [fichero₁, fichero₂, ..., fichero_n]:** Se emplea para ordenar, fusionar ordenadamente o comprobar si están ordenadas todas las líneas de los ficheros de entrada.
 - -c: Comprueba si los ficheros de entrada están ordenados.
 - -m: Fusiona ordenadamente los ficheros de entrada. Los ficheros de entrada tienen que estar ordenados
 - -r: Ordena en orden inverso (de mayor a menor)
 - -t delim: Establecemos un nuevo carácter delimitador (Por defecto espacio y tabulador).

Secure Shell (SSH)

- ❑ SSH es el nombre de un protocolo de aplicación y del programa que lo implementa.
- ❑ Permite acceder a máquinas remotas de forma segura a través de una red y controlar a éstas a través de un intérprete de comandos.
- ❑ Es ampliamente empleado para aumentar la seguridad de otros protocolos no seguros.
- ❑ Conexión a murillo mediante SSH:

```
> ssh <userlogin>@murillo.eii.us.es
```

File Transfer Protocol (FTP)



- FTP es un protocolo no seguro para la transmisión de ficheros a través de Internet.
- Usado junto con SSH pasa a ser un protocolo seguro denominado SFTP (Secure FTP). Para conectarnos a nuestra cuenta de murillo mediante SFTP:

```
> sftp <userlogin>@murillo.eii.us.es
```

- Comandos más relevantes:
 - **get file:** Descarga un fichero del servidor FTP.
 - **mget file₁ file₂ ... file_n:** Descarga un conjunto de ficheros del servidor.
 - **put file:** Sube al servidor FTP un fichero.
 - **mput file₁ file₂ ... file_n:** Sube al servidor varios ficheros.