



El sistema operativo UNIX

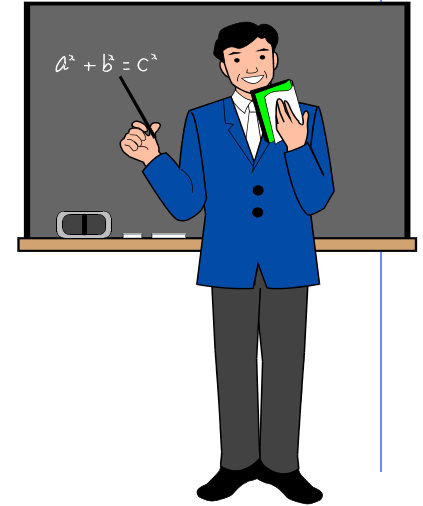
La interfaz de usuario I

Juan Carlos Yelmo

Contenidos

3. La interfaz de usuario

- Las shells de UNIX
- Sintaxis y entorno de comandos
- Redirección de entrada y salida
- Composición de comandos
- Variables de la Shell
- Inicialización de sesión



Introducción

- ◆ Interfaz de usuario de un sistema software
 - Conjunto de aspectos de un sistema software que un usuario puede ver (en general percibir) y los comandos y mecanismos que el usuario utiliza para operar el sistema e introducir datos.
 - Hay dos tipos básicos: GUIs e interfaces textuales o de línea de comandos
- ◆ La shell es la interfaz de usuario del sistema operativo UNIX
- ◆ Es un intermediario entre el usuario y el kernel del sistema operativo

La shell

- ◆ UNIX puede tener varias interfaces, textuales o gráficas, sustituibles y configurables por el usuario
- ◆ La shell original es un intérprete de línea de comandos: Bourne Shell (sh)
- ◆ La shell estándar de LINUX es una reimplementación de la Bourne Shell: Bourne Again Shell (bash)
- ◆ Hay interfaces gráficas a UNIX similares a los entornos MS-Windows: OS X Aqua, GNOME (GNU), KDE, Unity, etc.

Usos de la Shell de UNIX

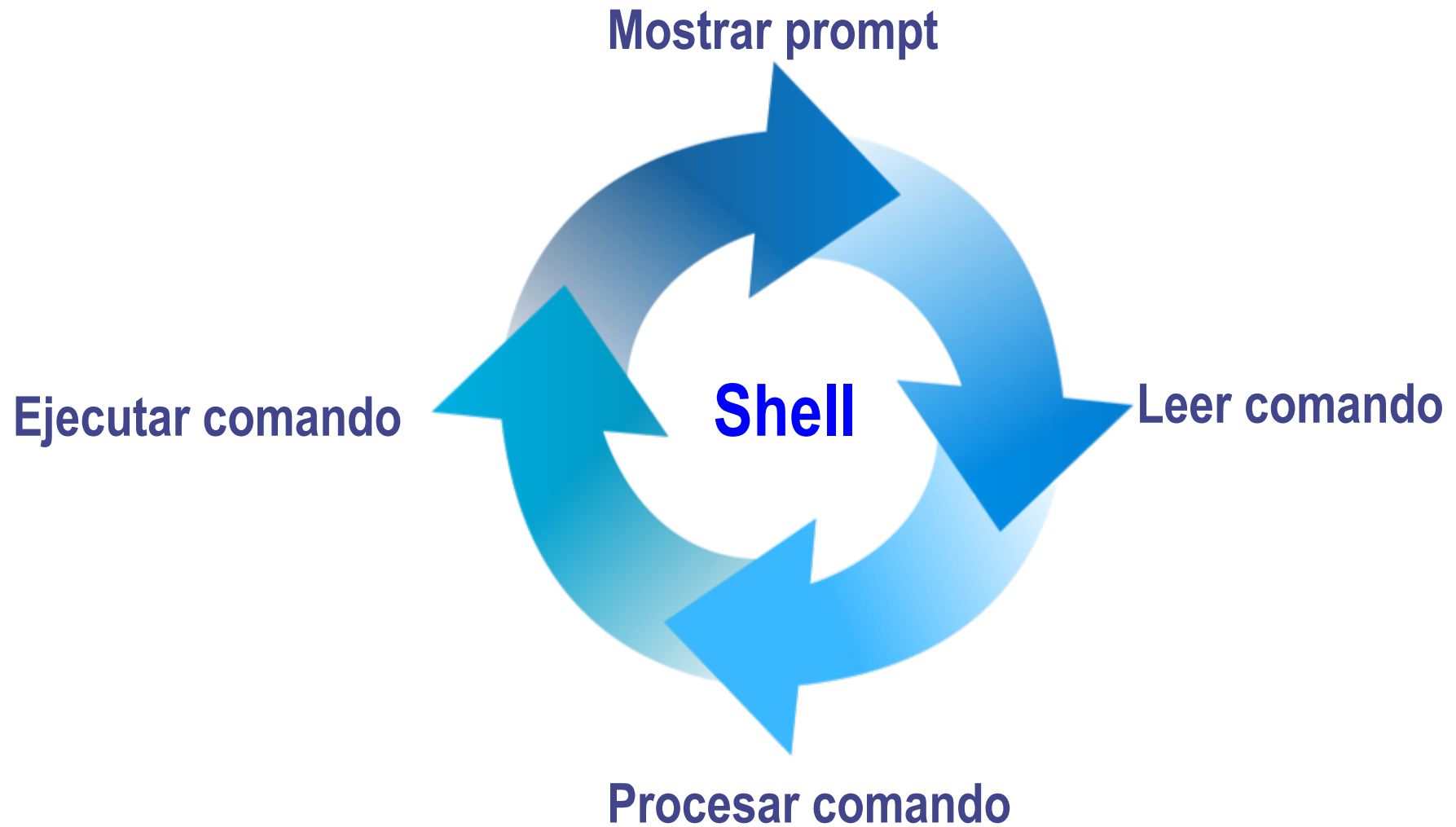
- ◆ Interacción con el usuario
- ◆ Personalización de sesión
- ◆ Programación por combinación de comandos (*shell script*)

Usos de la Shell

◆ Uso interactivo

- La shell muestra el prompt, espera que el usuario introduzca comandos y argumentos, analiza sintácticamente la línea, ejecuta el comando y devuelve el prompt
- Tiene facilidades de uso de metacaracteres, redireccionamiento de entrada-salida, combinación de comandos, historia de comandos, etc.

Bucle de procesamiento de comandos



Usos de la Shell

◆ Personalización de la sesión UNIX

- La Shell define variables que controlan la sesión de usuario. Definiendo estas variables el usuario puede configurar la sesión según gustos y necesidades
- La sesión de usuario se configura inicialmente a través de ficheros que se leen cuando se inicia la shell. En estos ficheros se pueden definir variables y comandos que se ejecutan al comienzo de la sesión

Usos de la Shell

◆ Programación

- La shell permite combinar comandos UNIX en un fichero (*shell script*) y proporciona comandos de control típicos de programación: bucle, bifurcación, etc. Esto convierte a la shell en un lenguaje de programación de alto nivel para escribir comandos nuevos o combinaciones frecuentes en base a comandos preexistentes

Los comandos UNIX

- ◆ Un comando (orden) es una cadena de caracteres que indica a la shell la realización de una acción específica
- ◆ Normalmente la acción es la búsqueda de un programa o shell script en el sistema de archivos y su ejecución o interpretación, aunque la shell tiene también comandos internos (*built in*)
- ◆ Un comando suele admitir opciones y argumentos

Los comandos de UNIX

◆ Sintaxis de comandos de la shell de UNIX

\$ comando [opciones] [argumentos]

Programa/acción a ejecutar

Modificadores

Datos de entrada

Los comandos UNIX

- ◆ **Comando.** Casi siempre el nombre de un fichero en el sistema de archivos
 - Con path absoluto: `/bin/ls`
 - Búsqueda en variable PATH: `ls`
- ◆ **Opciones.** Modificadores de la acción básica realizada por el comando
 - Casi siempre precedidas de “-”
 - Casi siempre una letra: `-a`
 - Casi siempre acumulativas:
 - ◆ `ls -a -l = ls -al`
- ◆ **Argumentos:** ficheros, máquinas, usuarios...

Argumentos: ficheros

¿Cuál?	
which <i>comando</i> Ejemplo: <code>\$ which pwd</code>	Devuelve el path absoluto del comando Devuelve el path del comando pwd
¿Dónde está?	
find <i>path opciones</i> Ejemplo: <code>\$ find . -name page.html -print</code>	Encuentra ficheros a partir de <i>path</i> Busca a partir del directorio actual la localización del fichero page.html
¿Qué ha cambiado?	
diff <i>file1 file2</i> Ejemplo: <code>\$ diff filev1 filev2</code>	Muestra las diferencias entre dos ficheros de texto línea a línea Muestra las diferencias entre dos versiones de un fichero

Argumentos: máquinas

Conexión segura	
ssh <i>hostname</i> Ejemplo: <code>\$ ssh -T git@github.com</code>	Login remoto seguro en la máquina especificada Comprueba accesibilidad ssh a github (hay que autenticarse)
Alcanzabilidad	
ping <i>hostname</i> Ejemplo: <code>\$ ping google.com</code>	Comprueba si la máquina es alcanzable en red y mide tiempo de eco Comprueba que google.com es alcanzable
Transferencia de archivos	
ftp <i>hostname</i> Ejemplo: <code>\$ ftp ftp.mozilla.org</code>	Establece conexión para transferencia de archivos Establece conexión con el servidor público de Mozilla (anónimo)



El sistema operativo UNIX

La interfaz de usuario I

Fin del tema

Juan Carlos Yelmo



El sistema operativo UNIX

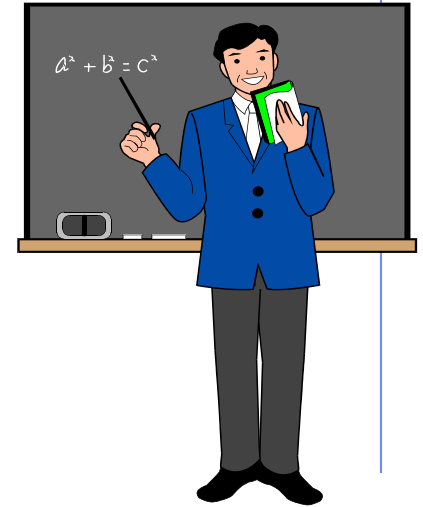
La interfaz de usuario II

Juan Carlos Yelmo

Contenidos

3. La interfaz de usuario

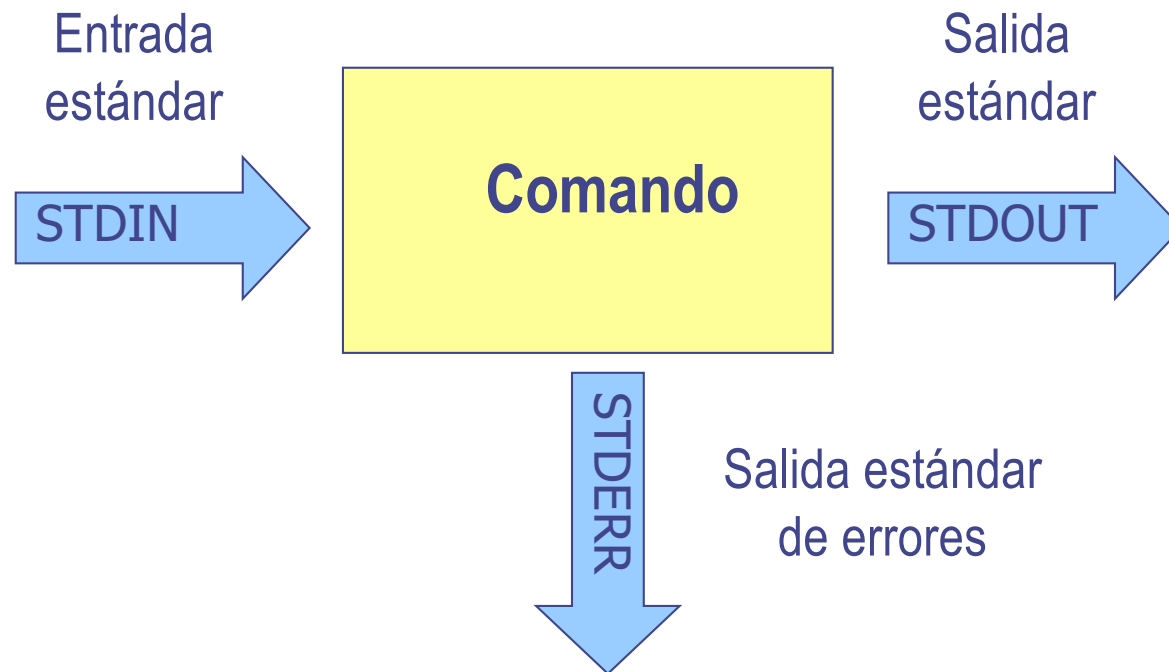
- Las shells de UNIX
- Sintaxis y entorno de comandos
- Redirección de entrada y salida
- Composición de comandos
- Variables de la Shell
- Inicialización de sesión



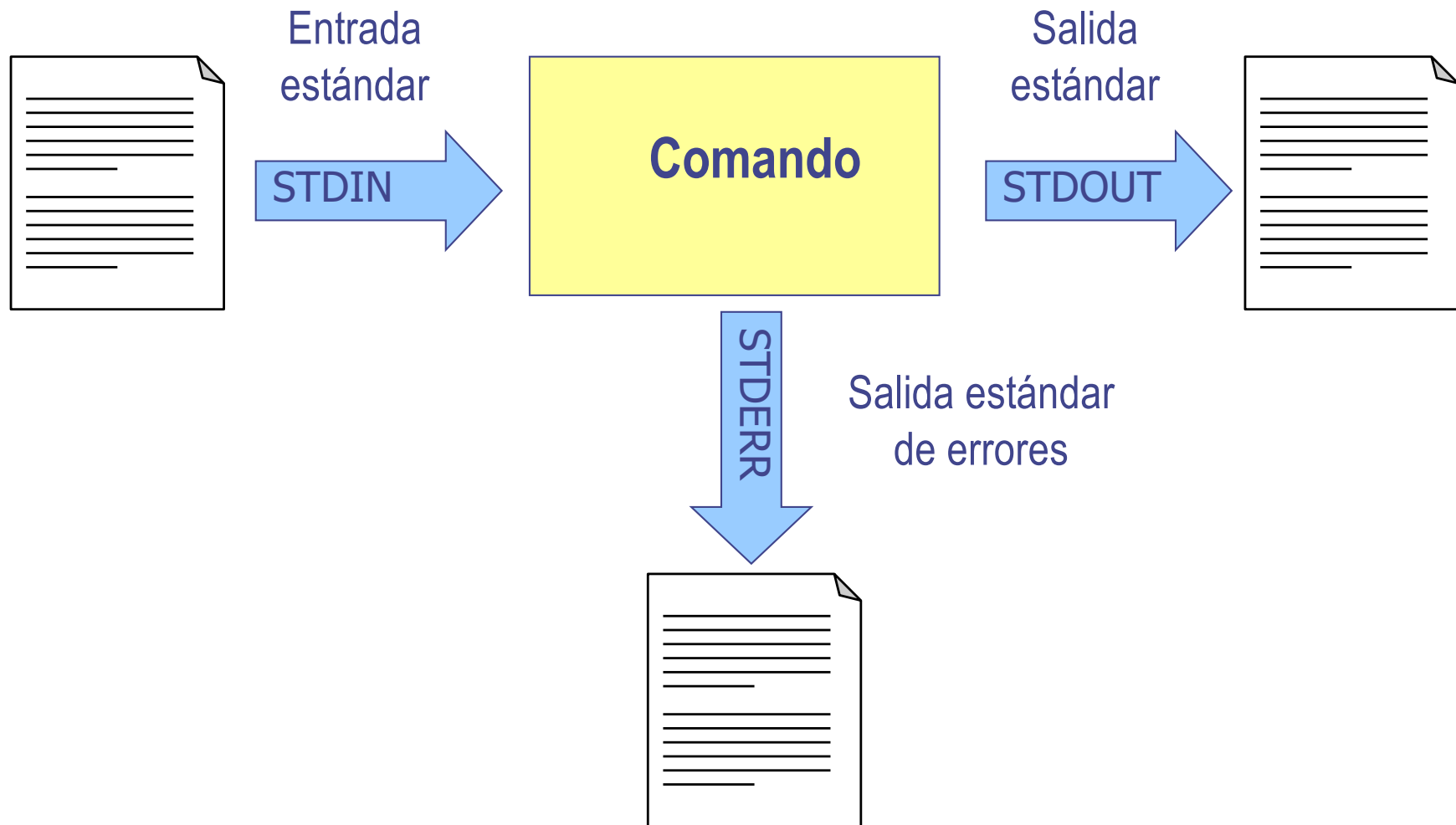
Entrada/salida de comandos

- ◆ Los argumentos de un comando suelen indicar la fuente de información de entrada (o el destino de los resultados de salida)
- ◆ Además de los argumentos, un comando UNIX típico tiene definidos unos canales (ficheros) de entrada y salida por defecto:
 - Entrada estándar (por defecto el teclado)
 - Salida estándar (por defecto la pantalla)
 - Salida estándar de errores (por defecto la pantalla)

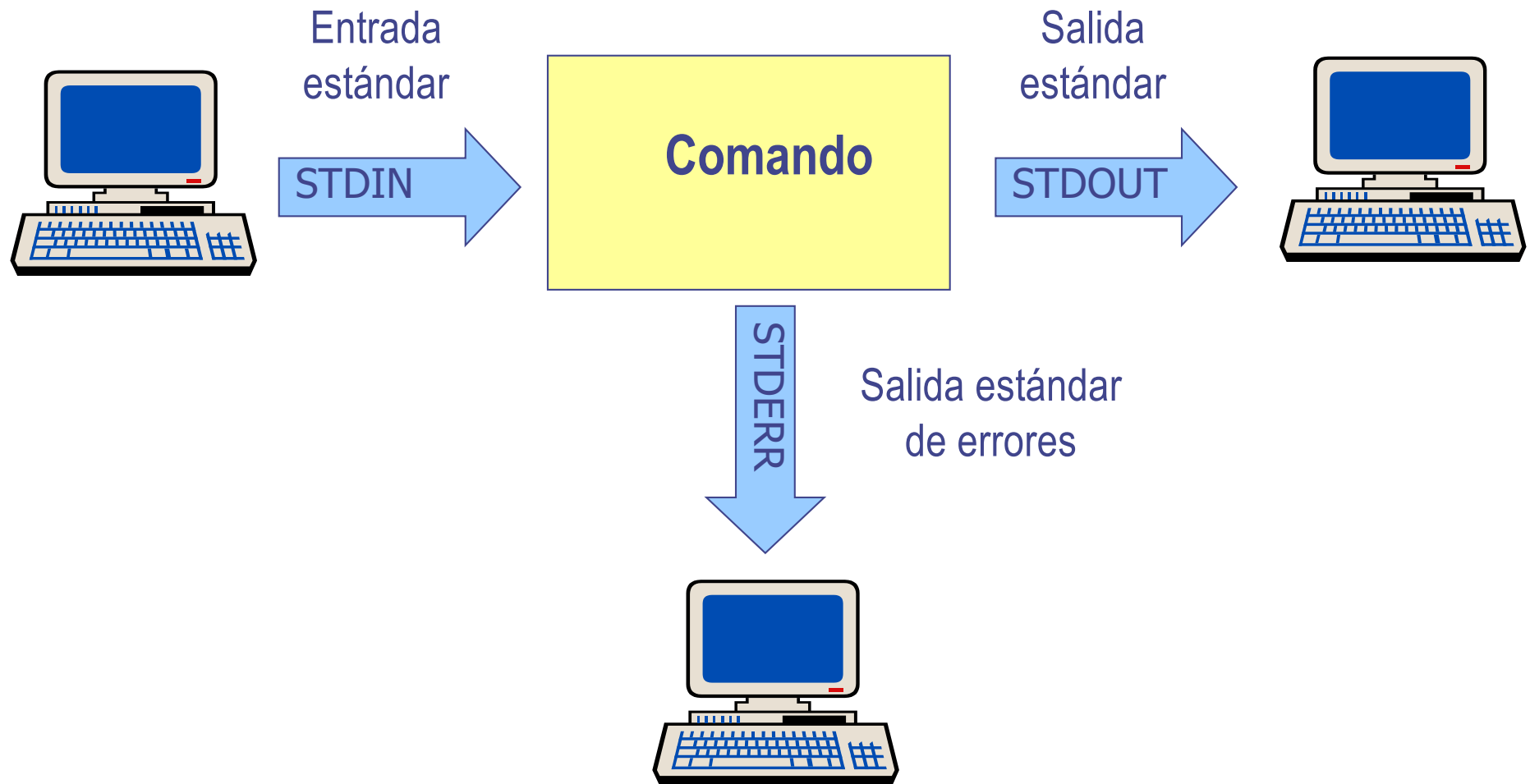
Entrada/salida de comandos



Entrada/salida de comandos



Entrada/salida de comandos



Entradas de un comando

◆ Sus opciones y argumentos.

- Son la entrada básica y explícita de información para el comando

◆ Ficheros predefinidos

- Ficheros de personalización de sesión (.login) o de configuración propios del comando (.exrc para vi)

◆ Variables de entorno

- TERM, PATH, PAGER, etc.

◆ La entrada estándar

Salidas de un comando

◆ Ficheros

- Nombre dado como argumento del comando

```
$ gcc programa.c -o programa programa
```

- Nombre por defecto

```
$ gcc programa.c (a.out)
```

◆ Salida estándar

◆ Salida estándar de errores

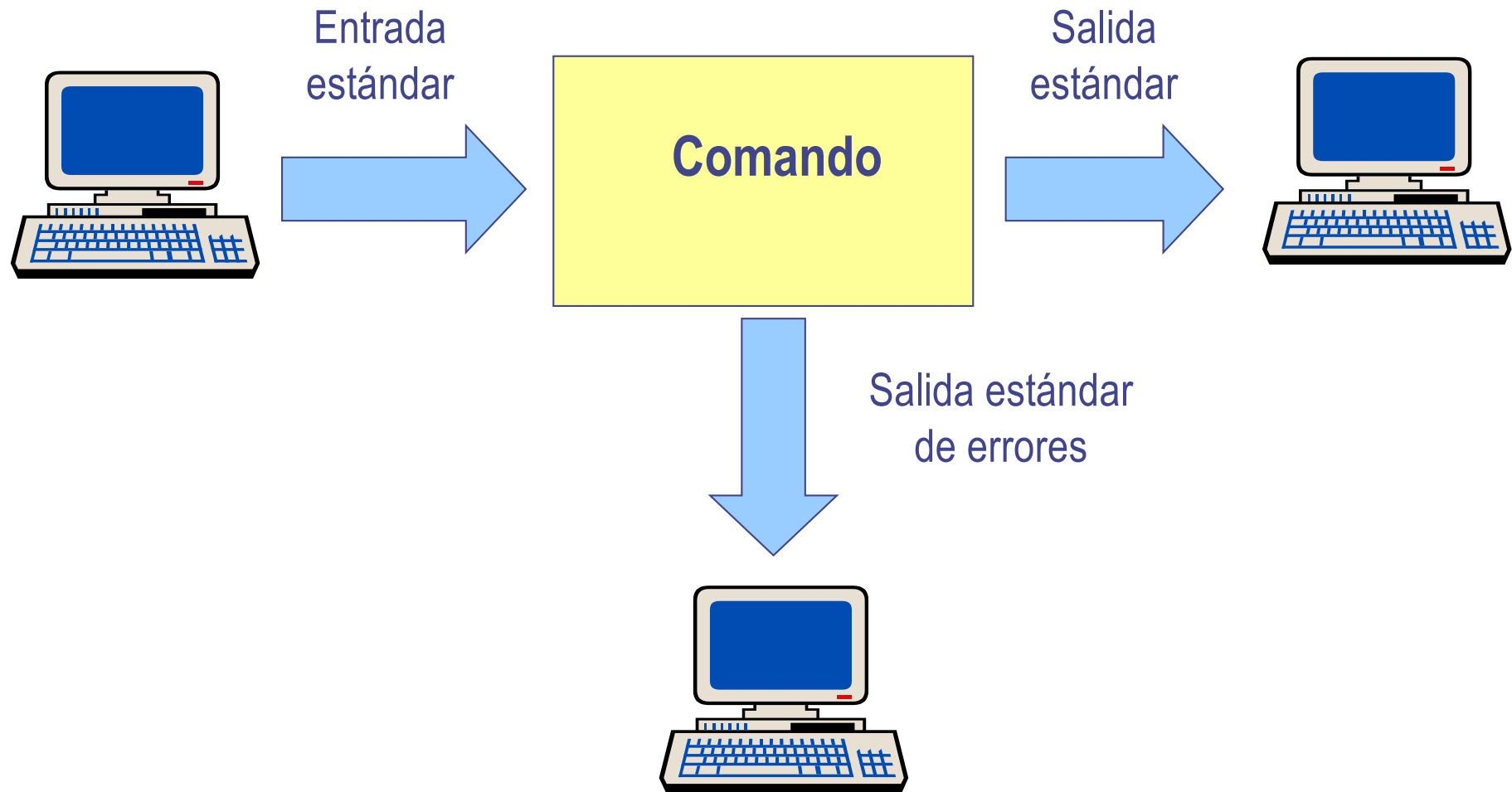
◆ Valor de retorno a la shell

- Cero (0) es ejecución con éxito, cualquier otro entero es ejecución con errores
- Se utiliza el comando `exit`: `exit 0`

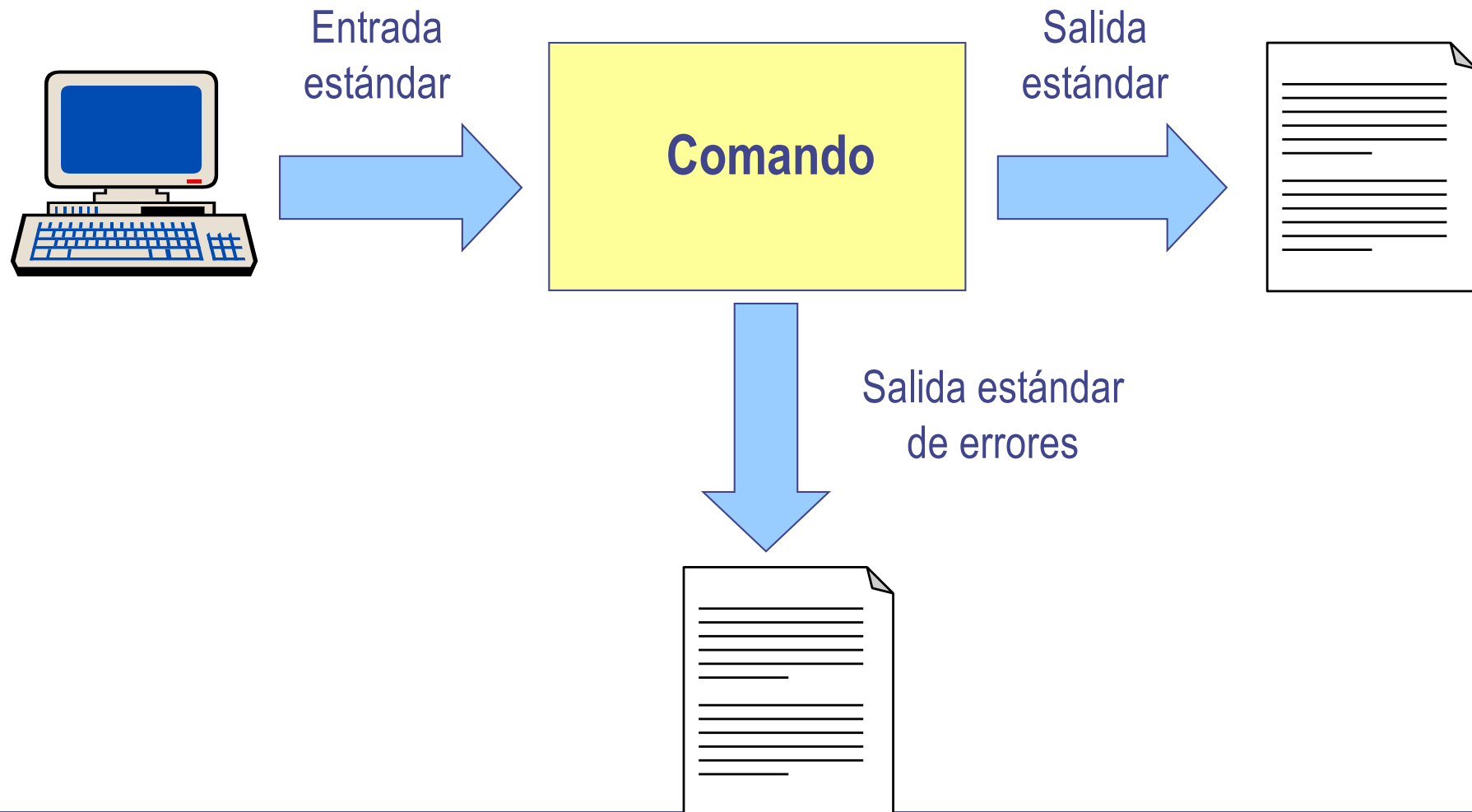
Redirección de entrada/salida

- ◆ Los canales de entrada y salida estándar de un comando se asocian por defecto al terminal de usuario, pero pueden asociarse a cualquier otro fichero (redirección de entrada/salida)
- ◆ La salida estándar de un comando puede asociarse con la entrada estándar de otro para conseguir la composición de comandos en cadena (pipeline)

Redirección de salida



Redirección de salida



Redirección de salida estándar

◆ Redirigir salida estándar

- comando > fichero
- `$ cal > calendario`

◆ Añadir salida a fichero preexistente

- comando >> fichero
- `$ cal >> calendario`

◆ Redirigir salida estándar a salida de error

- comando >&2
- `$ echo "Error" >&2`

Redirección de entrada estándar

◆ Redirigir entrada estándar

- comando < fichero
- \$ wc < poema

◆ Entrada en la misma línea de comando (here document)

- comando << marca

...

marca

- \$ wc << fin

Volverán del amor en tus oídos
las palabras ardientes a sonar;
fin

Redirección. Resumen

Salida estándar (stdout)			
Defecto	> file	>> file	>&2
Pantalla	Redirigir a fichero	Añadir a fichero	Combinar con stderr
	>! machaca file si existe	>>! crea file si no existe	

Redirección. Resumen

Entrada estándar (stdin)			
Defecto	< file	<< END	
Teclado	Entrada desde fichero	Entrada en la misma línea de comando hasta la marca END	
		Here document La marca de fin de entrada puede ser cualquiera	

Composición de comandos

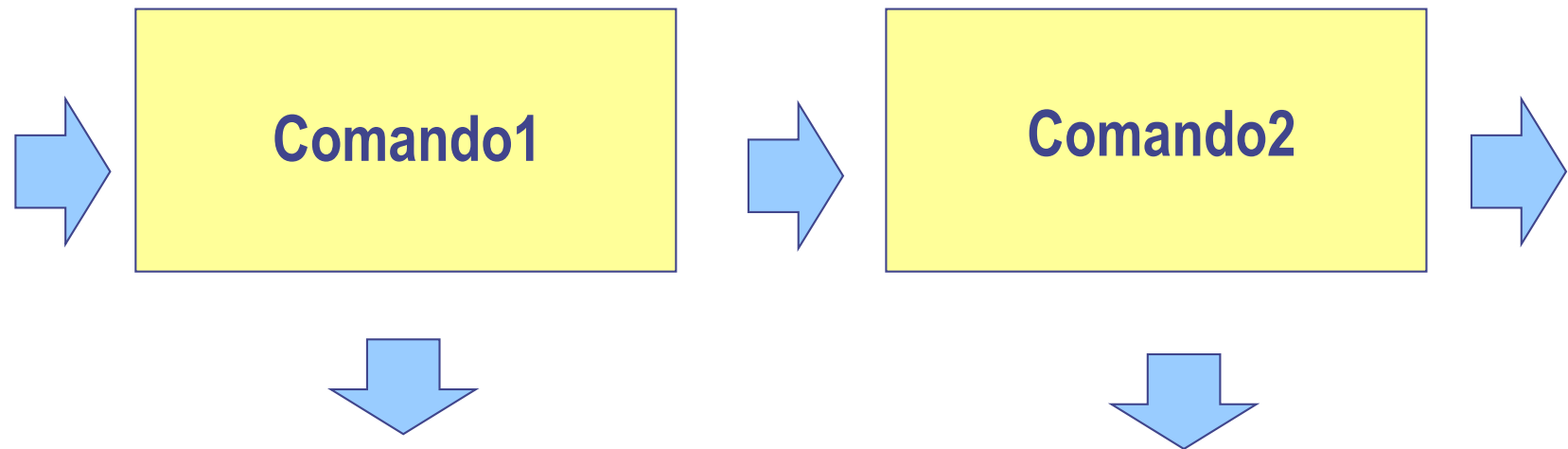
◆ Secuencia de comandos

- `comando1; comando2`
- `$ date; who`

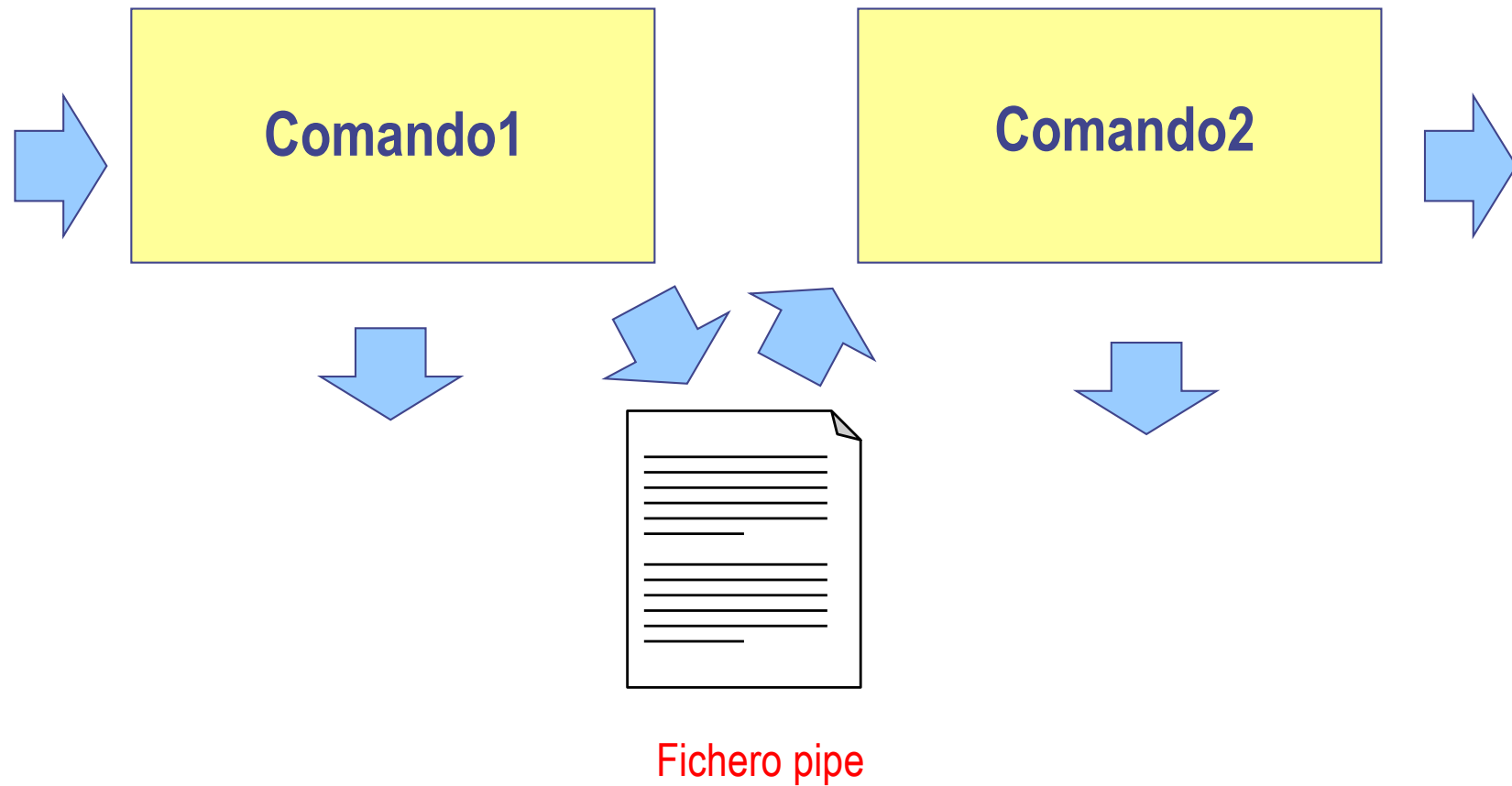
◆ Encadenamiento de comandos con un solo flujo de información (pipeline)

- `comando1 | comando2 ...`
- `$ grep -i amor poema | wc -l`

Pipelines



Pipelines



Comandos UNIX

Cuenta palabras	
<i>wc opciones fichero</i> Ejemplo: <code>\$ wc -l soneto.txt</code>	Cuenta caracteres, líneas y palabras en <i>fichero</i>. Sin <i>fichero</i> usa la entrada estándar Cuenta líneas en el fichero <i>soneto.txt</i>
Búsqueda de patrones	
<i>grep opciones patron file</i> Ejemplo: <code>\$ grep -w cara fichero.txt</code>	Busca palabras que encajan con el patrón. El patrón puede representar un conjunto de cadenas de caracteres mediante metacaracteres y otras expresiones Busca la palabra <i>cara</i> , completa, en el fichero <i>fichero.txt</i> e imprime las líneas que la contienen. No mostraría, por ejemplo, líneas que contengan la palabra <i>caramelo</i>



El sistema operativo UNIX

La interfaz de usuario II

Fin del tema

Juan Carlos Yelmo



El sistema operativo UNIX

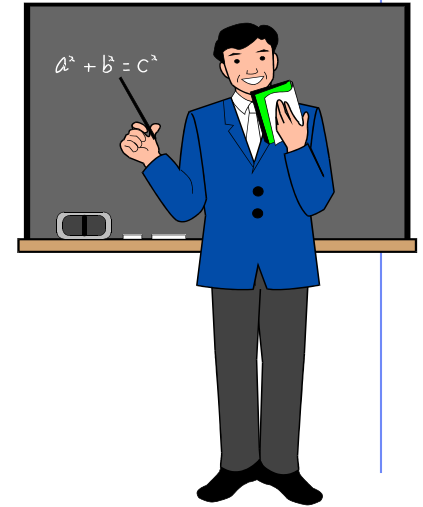
La interfaz de usuario III

Juan Carlos Yelmo

Contenidos

3. La interfaz de usuario

- Las shells de UNIX
- Sintaxis y entorno de comandos
- Redirección de entrada y salida
- Composición de comandos
- Variables de la Shell
- Inicialización de sesión



Variables de shell

- ◆ La shell permite la asignación y consulta de valores a variables. Estas variables pueden ser utilizadas por la propia shell o por los comandos y programas ejecutados por ésta
- ◆ Los valores de las variables son cadenas de cero o más caracteres
- ◆ Un identificador de variable puede ser cualquier palabra que comience con una letra y continúe con letras, dígitos y el carácter de subrayado

Uso de variables

◆ Asignación

■ En sh y bash

- ◆ `$ nombre=manu`

◆ Uso

- ◆ `$ echo $nombre`
manu

- ◆ `$ echo "${nombre} perez"`
manu perez

- ◆ `$ echo "${nombre}el perez"`
manuel perez

Tipos de variables

- ◆ **Variables ordinarias.** Variables locales de propósito general
- ◆ **Variables de entorno.** Describen el contexto de ejecución y se heredan
- ◆ **Variable especiales de la shell.** Configuran el entorno de la propia shell.
 - Variables para parámetros posicionales en la invocación de comandos

Variables de entorno

- ◆ Describen el entorno de ejecución:
 - TERM: Tipo de terminal
 - PATH: Lista de directorios por defecto para búsqueda de comandos
 - HOSTNAME: Nombre de la máquina
 - USER: Nombre del usuario
 - SHELL: Shell por defecto
 - HOME: Directorio base del usuario
 - etc.

Variables de entorno

◆ Asignación (sh y bash)

- `$ TERM=vt100; export TERM`
- `$ PATH=$PATH:/home/jcyelmo; export PATH`
- `$ export PATH=$PATH:/home/jcyelmo`

◆ Consulta (sh y bash)

- `$ echo $PATH`
- `$ printenv`

Variables posicionales

- ◆ Albergan los parámetros de entrada a los comandos
 - `$ <comando> <arg1> arg2> ...`
 - `$0 $1 $2 ...`
 - `$ cc -o programa parte1.c parte2.c`
 - ◆ `$0= cc`
 - ◆ `$1= -o`
 - ◆ `$2= programa`
 - ◆ `$3= parte1.c`
 - ◆ `$4= parte2.c`

Evaluación de variables

<code>\$var</code>	Valor de la variable <i>var</i> , si está definida
<code>\${var}</code>	Lo mismo pero delimita el nombre de la variable cuando está inserta en una cadena mayor
<code>\${var-valor}</code>	Valor de la variable <i>var</i> , si está definida. Si no se usa <i>valor</i>
<code>\${var=valor}</code>	Valor de la variable <i>var</i> , si está definida. Si no se usa <i>valor</i> y se asigna <i>valor</i> a <i>var</i>
<code>\${var?mensaje}</code>	Valor de la variable <i>var</i> , si está definida. Si no imprime mensaje en salida de error
<code>\${var+valor}</code>	Usa <i>valor</i> si la variable <i>var</i> está definida

Selección de la shell

- ◆ Cada usuario tiene una shell por defecto para interactuar con el sistema operativo
- ◆ La shell por defecto está especificada en el último campo de la línea de registro del usuario en el fichero `/etc/passwd`
- ◆ El usuario puede cambiar su shell por defecto con `chsh`
 - Pide password y path de la nueva shell

Inicialización de la shell

- ◆ Cuando se ejecuta una shell interactiva, ésta ejecuta unos scripts de inicialización que asignan valores predeterminados por el usuario para variables especiales y de entorno y ejecutan comandos
- ◆ Estos ficheros de inicialización sirven para particularizar la sesión a los gustos y necesidades de cada usuario

Inicialización de la shell

- ◆ Los ficheros de inicialización de la shell residen en el directorio base del usuario
- ◆ También hay ficheros de inicialización globales para todos los usuarios
- ◆ Tienen nombres predeterminados
 - Inicialización de sesión para un usuario
 - ◆ \$HOME/.profile
 - Inicialización de sesión común para todos
 - ◆ /etc/profile

Tipos de Shell

- ◆ **Interactiva.** Permite interactuar al usuario con el sistema operativo
 - **Login shell:** La que se inicia tras el proceso de login. En un sistema con desktop gráfico, el propio desktop es la única login shell
 - **Non-login shell:** Una shell ejecutada como comando desde otra shell. Hereda el entorno y ejecuta ficheros de inicialización específicos. La shell de una ventana de terminal en Ubuntu es una non-login shell
- ◆ **No interactiva.** Entorno de ejecución de un comando lanzado por una shell interactiva, de la que hereda el entorno

Inicialización en bash

◆ Varios ficheros para particularización de comienzo y fin de sesión

- /etc/profile
 - ◆ Inicialización global para login shells
- /etc/bashrc
 - ◆ Configuración global
- \$HOME/.bash_profile
 - ◆ Ejecuta al comienzo de sesión
- \$HOME/.bashrc
 - ◆ Ejecuta al comienzo de una shell
- \$HOME/.bash_logout
 - ◆ Particulariza el fin de sesión

Ejemplo de .bash_profile

```
# .bash_profile

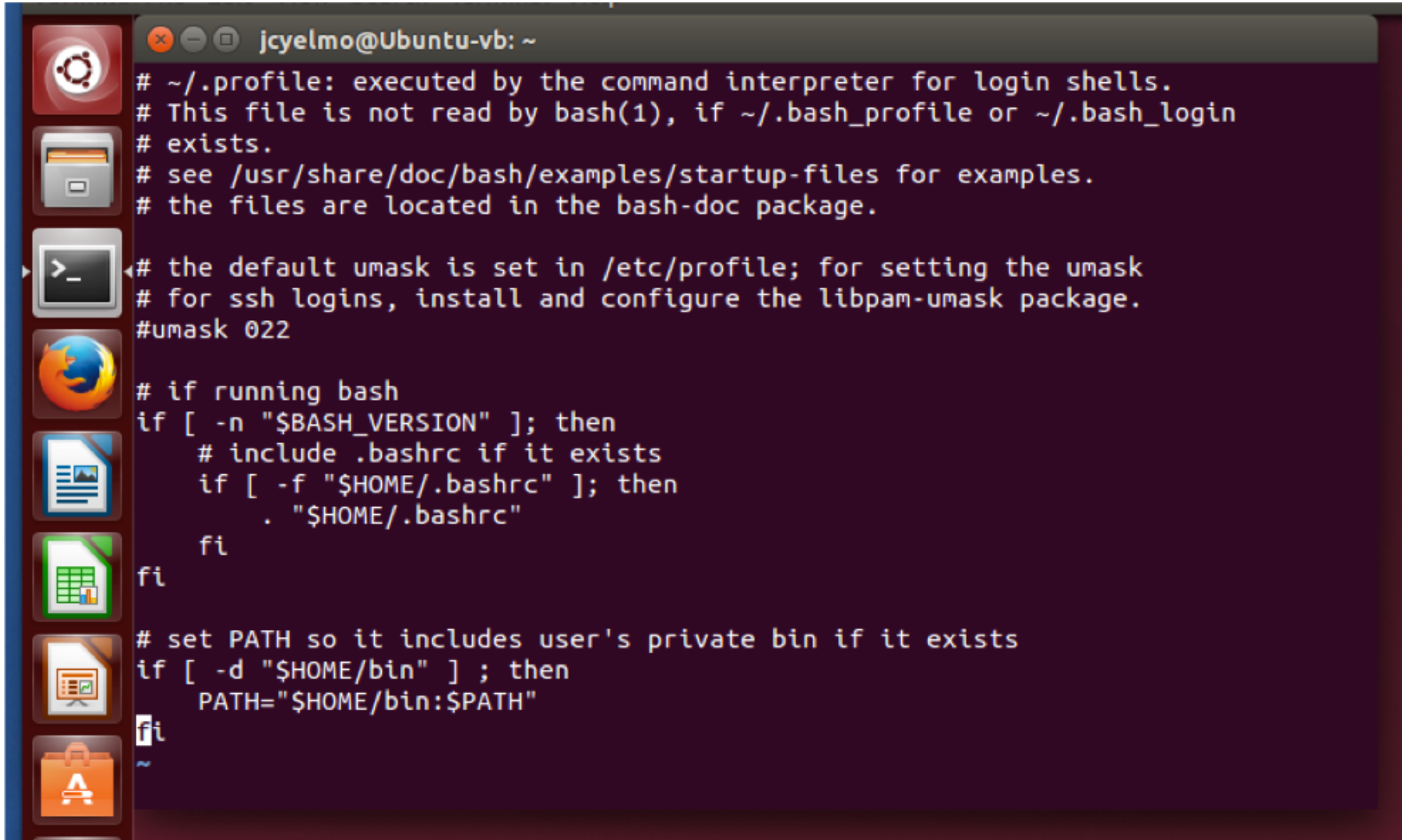
# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin
BASH_ENV=$HOME/.bashrc
USERNAME=" "

export USERNAME BASH_ENV PATH
```

Ejemplo de .profile



The image shows a terminal window titled 'jcyelmo@Ubuntu-vb: ~'. The terminal displays the contents of the ~/.profile file, which is a shell script for login shells. The script includes comments explaining its purpose and location, and sets the default umask to 022. It also includes conditional logic to source the .bashrc file if it exists and to set the PATH variable to include the user's private bin directory if it exists.

```
jcyelmo@Ubuntu-vb: ~  
# ~/.profile: executed by the command interpreter for login shells.  
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login  
# exists.  
# see /usr/share/doc/bash/examples/startup-files for examples.  
# the files are located in the bash-doc package.  
  
# the default umask is set in /etc/profile; for setting the umask  
# for ssh logins, install and configure the libpam-umask package.  
#umask 022  
  
# if running bash  
if [ -n "$BASH_VERSION" ]; then  
    # include .bashrc if it exists  
    if [ -f "$HOME/.bashrc" ]; then  
        . "$HOME/.bashrc"  
    fi  
fi  
  
# set PATH so it includes user's private bin if it exists  
if [ -d "$HOME/bin" ] ; then  
    PATH="$HOME/bin:$PATH"  
fi  
~
```



El sistema operativo UNIX

La interfaz de usuario III

Fin del tema

Juan Carlos Yelmo