

El shell

Es el intérprete de comandos: actúa como interfaz entre el usuario y el sistema.

Permite:

- Utilizar abreviaturas de nombres de ficheros
- Expandir y modificar las órdenes
- Redireccionar la E/S
- Encadenar varias ordenes

Proporciona un lenguaje de programación: shell scripts.

Tipos:

- Bourne Shell(sh)
- Bourne Again Shell(bash)
- Korn Shell(ksh)
- C Shell(csh)

Línea de órdenes

```
$ orden [-opciones] [argumentos]
```

Una orden es un archivo ejecutable del sistema

Se distinguen mayúsculas de minúsculas

Opciones

- Van precedidas de guión
- Pueden agruparse precedidas de un guión simple
- Si no se agrupan deben separarse mediante un espacio o tabulador
- El orden relativo de las opciones no es significativo

Argumentos

- Van separados por espacio o tabulador

➤ Metacaracteres

- * → cualquier número de caracteres o ninguno
- ? → uno y solo un carácter
- [] → cualquier carácter encerrado en los corchetes

- \$ → contenido de una variable
- # → comentario
- `` → ejecución de órdenes
- " " → permiten la expansión de \$ y de ``
- ‘ ‘ → no permiten la expansión de \$ y de ``
- ; → secuenciamiento de órdenes
- && → Ejecución condicional *and*
- || → Ejecución condicional *or*
- (.) → Agrupamiento de órdenes
- & → Ejecución en segundo plano
- | → Tubería
- > → Direccionamiento de salida
- >> → Direccionamiento de salida
- < → Direccionamiento de entrada

Comandos básicos

➤ Directorios

\$ pwd

- Muestra por pantalla el nombre de camino completo del directorio actual

\$ cd [directorio]

- Cambia el directorio de trabajo.
- Sin parámetros, establece como directorio de trabajo el *home*.

\$ ls [opciones] [fichero]

- Muestra el contenido de un directorio.
- Opciones:
 - F → ejecutable (*), directorio (/).
 - R → listado recursivo.
 - a → lista todas las entradas.
 - l → listado en formato largo.

\$ mkdir [opciones] directorio

- Crea un directorio.
- Opciones:
 - p → permite crear los directorios padres que falten.

\$ rmdir directorio

- Borra un directorio, siempre y cuando esté vacío.

➤ **Ficheros**

\$ cat [opciones] [fichero]

- Lee un fichero y muestra su contenido por pantalla.
- Sin parámetros, lee de la entrada estándar.

\$ cp [opciones] fichero1 fichero2

- Copia el contenido de fichero1 en fichero2.
- r → copia recursivamente. Para copiar directorios.

\$ rm [opciones] fichero

- Borra ficheros y directorios.
- r → indica comportamiento recursivo. Para borrar directorios.

\$ mv [opciones] fichero1 fichero2

- Mueve ficheros y directorios dentro del sistema de ficheros.
- Equivale a renombrar un fichero o directorio.

\$ ln [-opciones] fichero enlace

- Crea una nombre adicional (enlace) a un fichero. Por defecto crea un enlace duro.
 - s: crea un enlace simbólico. Directorios y ficheros de un *filesystem* distinto.
- Existen dos tipos de enlace, llamados usualmente enlace duro (o físico) y enlace blando (o simbólico). Un enlace duro es simplemente un nombre para un fichero.
- Un fichero puede tener varios nombres. Se borra del disco solamente cuando se elimine el último nombre. El número de nombres lo muestra *ls*. No existe el concepto de nombre 'original', todos tienen la misma categoría.

- Un enlace blando (o enlace simbólico, o acceso directo) es un fichero que contiene un nombre de camino. Así, los enlaces blandos pueden apuntar a ficheros en sistemas de ficheros diferentes, y no tienen por qué apuntar a ficheros que existan realmente.

\$ file [opciones] fichero

- Determina el tipo de un fichero.

➤ **Sistema de ficheros**

\$ du [opciones][fichero]

- Muestra un resumen del uso de disco para cada fichero, recursivamente para directorios.
- Opciones:
 - a → muestra todos los ficheros
 - b → muestra el tamaño en bytes
 - h → muestra los tamaños de forma legible (p.ej., 1K 234M 2G)

\$ df [opciones][fichero]

- Muestra información sobre el sistema de ficheros en el que reside cada fichero, o por omisión sobre todos los sistemas de ficheros.
- Opciones
 - h → imprime los tamaños en formato legible (p.e. 1K 234M 2G)
 - i → muestra la información de nodos-i en lugar del uso de bloques
 - l → limita el listado a los sistemas de ficheros locales

➤ **Información del usuario**

\$ whoami

- Muestra el usuario

\$ id [opciones] [usuario]

- Muestra información del usuario especificado (uid, gid, grupos, etc.).

\$ uname -a

- Muestra la versión de Linux

\$ hostname

- Nombre de la máquina

\$ who [opciones]

- Muestra los usuarios conectados
- Opciones
 - b → tiempo del último inicio del sistema
 - H → muestra la línea de encabezados de columnas
 - r → muestra el runlevel actual

\$ w [opciones]

- Muestra información sobre los usuarios conectados

➤ Filtros

\$ tr [opciones] conjunto1 [conjunto2]

- Traduce, comprime y borra caracteres de la entrada estándar, escribiendo el resultado en la salida estándar.
- Opciones:
 - c → opera sobre el complemento (sobre cada carácter que no coincida)
 - d → borra caracteres de conjunto1, no traduce
 - s → reemplaza cada sucesión de entrada de un carácter repetido del conjunto1 por una sola aparición de dicho carácter.

\$ head [opciones] [fichero]

- Muestra las 10 primeras líneas de cada fichero en la salida estándar.
- Opciones:
 - cTAMAÑO → muestra los primeros TAMAÑO bytes
 - nN → muestra las N primeras líneas

\$ tail [opciones] [fichero]

- Muestra las últimas 10 líneas de cada fichero en la salida estándar.
- Opciones:
 - cN → muestra los últimos N bytes
 - f → muestra a medida que el fichero crece;
 - nN → muestra las últimas N líneas en lugar de 10
- Si el primer carácter de N es un +, muestra a partir del elemento N-ésimo hasta el final, contando desde el principio de cada fichero

\$ sort [opciones] [fichero]

- Muestra la concatenación ordenada de todos los ficheros en la salida estándar.
- Opciones de ordenación:
 - b → descarta los espacios en blanco al principio
 - g, n → compara de acuerdo con el valor numérico
 - r → invierte el resultado de las comparaciones
 - k POS1[,POS2] → establece el criterio de ordenación en POS1 y la termina en POS2. Un sólo campo, POS1=POS2. Si no indicamos POS2, hasta el final. El primer campo es 1.
 - o FICHERO → escribe el resultado en FICHERO, en lugar de la salida estándar
 - t SEP → usa SEP en lugar de espacio para separar los campos
 - u → muestra solamente la primera de una serie de repeticiones
- POS es F[,C][OPCIONES]
 - F es el número de campo y C la posición del carácter en el campo.
 - OPCIONES se compone de una o más de las opciones de ordenación

\$ uniq [opciones] [ENTRADA [SALIDA]]

- Descarta todas las líneas sucesivas idénticas, menos una de ENTRADA (o entrada estándar), escribiendo en SALIDA (o en la salida estándar).
- Opciones:
 - c → precede a las líneas con el número de ocurrencias
 - d → muestra sólo las líneas duplicadas
 - u → muestra sólo las líneas que son únicas

\$ cut [opciones] [fichero]

- Extrae las partes seleccionadas de cada fichero en la salida estándar
- Opciones
 - c → muestra solamente los caracteres que ocupan la posición indicada
 - dDELIM → Utiliza DELIM en vez del tabulador para delimitar los campos
 - f → muestra solamente los campos que ocupan la posición indicada
 - s → no muestra las líneas que no contienen delimitadores

`$ paste [opciones] [fichero]`

- Escribe secuencialmente en la salida estándar cada línea de los ficheros especificados, separadas por tabuladores.
- Opciones
 - dDELIM → usa el caractere indicado en DELIM en lugar de tabulador
 - s → usa un fichero cada vez, en lugar de hacerlo en paralelo

`$ diff [opcion] fichero1 fichero2`

- Muestra las diferencias, línea por línea, entre dos ficheros o dos directorios

`$ more/ less`

- Muestra un fichero de forma paginada

`$ grep [opciones] patron [fichero]`

- Opciones
 - i → considera iguales mayúsculas y minúsculas
 - w → obliga a que `patron` coincida solamente con palabras completas
 - x → obliga a que `patron` coincida solamente con líneas completas
 - v → selecciona las líneas que no coinciden
 - mNUM → se detiene después de NÚM coincidencias
 - n → muestra el número de línea junto con las líneas de salida
- `patron` puede construirse mediante:
 - Expresión regular
 - Frase (entre comillas)

`$ wc [-cwl][fichero]`

- Cuenta los caracteres, palabras y líneas de un fichero

`$ find ruta opciones`

- Busca los ficheros que satisfacen la expresión de búsqueda a partir de los directorios señalados
- Opciones
 - name → nombre
 - type → f,d,l,etc.
 - size → tamaño
 - * c → byte, k → kilobyte
 - * + → mayor que, - → menor que, sin signo igual al tamaño

- perm → permisos
 - * en octal
 - * en símbolo ugo+=rwx;
 - * - delante indica al menos esos permisos, sin menos los permisos exactos
 - * / delante indica alguno de los permisos de cada modo (u,g,o)

- links → número de enlaces

- user → propietario de los ficheros

- Por defecto cada expresión va unida por *and*, explícitamente *-a*. También podemos unir las expresiones por *or* *-o*.

- Se pueden poner paréntesis `\(... \)`.

- Se pueden realizar ejecuciones para los ficheros que cumplan las condiciones

- `$ find ruta opciones -exec comando {} \;`

- `$ find ruta opciones -ok comando {} \;`

* Pide confirmación

➤ Expresión regular

- Formada por caracteres combinados con operadores

- Operadores:

- ^ → Principio de línea

- \$ → Final de línea

- \< → Principio de palabra

- \> → Final de palabra

- . → Cualquier carácter excepto salto de línea

- [] → Conjunto de caracteres

- [^] → Cualquier carácter no contenido

- [-] → Rango

- * → Cero o más ocurrencias del elemento que lo precede

- + → Uno o más ocurrencias del elemento que lo precede

- ? → El elemento precedente es opcional

- () → Agrupación

- | → O uno u otro

- {n} → El elemento precedente se repite n veces

- {n,} → El elemento precedente se repite al menos n veces

- {n,m} → El elemento precedente se repite al menos n y no más de m veces

- \ → Escape (Los caracteres ? + { } | () deben aparecer con escape y la cadena entrecomillada " ")

➤ Procesos

\$ ps [-opciones]

- Muestra el estado de los procesos del sistema. Por defecto selecciona todos los procesos del usuario actual asociados a la terminal que lo invoca. Muestra el PID, terminal asociada, el tiempo de CPU y el nombre del comando.
- Opciones:
 - e → Muestra todos los procesos (Identica a -A)
 - a → Todos los procesos excepto los líderes de sesión y los procesos no asociados a terminal
 - N → Todos los procesos menos los especificados
 - u, U → Selecciona los procesos del EUID o UID indicados
 - f → Muestra en formato largo
 - o → Muestra un formato definido por el usuario:
 - * %cpu → Porcentaje de uso de la CPU
 - * %mem → Porcentaje de uso de la memoria
 - * cmd → Comando que ejecuta el proceso
 - * Cputime → Tiempo de CPU utilizado
 - * pid → Identificador del proceso
 - * ppid → Identificador del proceso padre
 - * s → Estado del proceso
 - Todos los procesos del sistema con la sintaxis BSD → `$ ps aux`

\$ kill [-señal] pid

- Envía una señal a un proceso. Por defecto, se envía la señal de terminación del proceso (SIGTERM).
- Algunas señales:
 - SIGINT (2): Interrumpir (cancelar) un programa. Es la señal que se le envía a un proceso cuando presionamos Ctrl+C en teclado con un proceso en primer plano.
 - SIGKILL (9): "Matar" un proceso. Los programas no pueden decidir cómo responder a esta señal. Terminan inmediatamente cuando la reciben. Sólo debería ser usada en el caso de necesitar terminar un proceso que ha dejado de responder y no funciona adecuadamente.
 - SIGSEGV (11): Señal de violación de segmento (segmentation fault). Esta señal le es enviada a un proceso cuando intenta acceder a memoria que "no es suya", que está fuera de la zona de memoria que ese proceso puede usar. El proceso es detenido inmediatamente.
 - SIGTERM (15): Se envía esta señal a un proceso cuando queremos que termine normalmente, haciendo lo que necesite antes de terminar.

- SIGCHLD (17): Esta señal se envía automáticamente a un proceso padre cada vez que un proceso hijo termina.
- SIGTSTP (19): El proceso queda parado a la espera de recibir una señal de continuación SIGCONT. SIGKILL y SIGSTOP son las únicas señales que no pueden enmascarar.
- SIGTSTP (20): Se le envía a un proceso cuando pulsamos Ctrl+Z. Equivalente a la anterior, pero ésta sí se puede enmascarar.
- Se puede ver información más detallada sobre las señales con `man 7 signal`.

\$ nice -n num orden [argumentos]

- Permite modificar la prioridad con la que se ejecutará un proceso.

\$ pstree

- Muestra el árbol de procesos

\$ top

- Muestra los procesos en ejecución con actualizaciones periódicas

\$ exec command

- El proceso que ejecuta el *shell*, pasa a ejecutar el comando, de manera que no se crea ningún proceso nuevo.

➤ Usuarios y grupos

\$ useradd [opciones] usuario

- Crea un nuevo usuario o actualiza la información de uno existente
- Opciones:
 - D → muestra las opciones por defecto. Contenido de *useradd*
 - d homedir → Indica el directorio *home* del usuario. Sin esta opción, el directorio se crea en el lugar por defecto
 - g gid → Nombre/número del grupo principal del usuario. El grupo debe existir
 - G group → Lista de grupos suplementarios (separados por coma, sin espacio)
 - k skel → Especifica un directorio *skel* alternativo
 - m → Crea el directorio *home* para el nuevo usuario
 - u uid → Identificador para el nuevo usuario. Si no se indica, se le da el siguiente al más grande utilizado.
 - s shell → Shell del usuario

\$ adduser usuario

- Idéntico a *useradd*

\$ groupadd grupo

- Crea un nuevo grupo.
- Opciones:
 - -g GID → asigna el identificador de grupo al nuevo grupo. El valor debe ser único. Si no se indica esta opción, se le da el valor inmediatamente superior a cualquier otro grupo y mayor que el valor mínimo especificado en *login.defs*

\$ userdel [-r] usuario

- Elimina un usuario
- Con -r se elimina, además, el directorio *home* del usuario

\$ groupdel [opciones] grupo

- Elimina un grupo existente. No se permite la eliminación de grupos primarios. Antes, se debe eliminar el usuario correspondiente.

\$ passwd [opciones] usuario

- Actualiza la contraseña de un usuario

\$ su [opciones] [usuario]

- Cambia el identificador efectivo de usuario y de grupo al de *usuario*.
- Si no se da el usuario, se supone *root*
- Con “-l” o “-“, crea un *shell* de entrada para el nuevo usuario

\$ newgrp grupo

- Cambia el identificador del grupo del usuario.

\$ chown [opciones] propietario:[grupo] fichero

- Cambia el dueño y/o grupo de cada fichero indicado

\$ chgrp [opciones] grupo fichero

- Cambia el grupo cada fichero indicado

\$ chmod [opciones] modo[,modo] ficheros

\$ chmod [opciones] modo_octal ficheros

- Cambia los permisos del fichero o directorio.

- Modo simbólico
 - a, u, g, o → all, user, group, others
 - +, -, = → añade, quita o establece permisos
 - r, w, x → read, write, execute
- -R → Cambia ficheros y directorios recursivamente

➤ Miscelánea

\$ echo

- Muestra una cadena por pantalla

\$ cal

- Muestra un calendario.

\$ date

- Muestra la fecha.

\$ alias [-p] [name=orden]

- Crea un seudónimo *name* para una orden.
- Con -p, lista todos los alias

\$ unalias [-a] [name ...]

- Elimina los alias *name* indicados.
- Con -a, elimina todos los alias.

\$ umask [mascara]

- Establece la máscara para los permisos de los ficheros nuevos.
- Sin argumentos indica la máscara actual.

\$ man [opciones]

- Muestra las páginas del manual en línea.

➤ Red

`$ ifconfig interface [dirección [opciones]]`

- Configura la interface de red.
- Sin argumentos muestra el estado de las interfaces activas
- Opciones:
 - up → activa la interface
 - down → desactiva la interface
 - netmask → Establece la máscara de red

`$ netstat`

- Muestra el estado de las conexiones de red, tabla de encaminamiento y las estadísticas de las interfaces
- Opciones:
 - r → tabla de encamienamiento
 - s → estadísticas
 - c → Muestra la información cada segundo

`$ ping [opciones] destino`

- Envía indefinidamente paquetes de prueba a destino.

`$ route`

- Muestra la tabla de rutas, o la modifica añadiendo o eliminando una determinada ruta.

`$ traceroute [opciones] destino`

- Muestra la ruta que siguen los paquetes a través de la red

➤ Compresión y descompresión de ficheros

`$ gzip [-opciones] fichero`

- Permite comprimir ficheros de manera individual.
- Opciones
 - d: descomprime un fichero
 - r: comprime los ficheros de una carpeta y subcarpetas.

`$ tar [-opciones] [fichero]`

- Guarda varios ficheros juntos en un solo archivo y puede restablecer ficheros individuales a partir del archivo.
- Opciones:
 - c → crea un nuevo fichero
 - x → extrae archivos de un fichero
 - t → lista el contenido del fichero
 - v → verbose
 - z → comprime o descomprime
 - f → guarda en el fichero indicado

➤ Parada i reinicio del sistema

`$ shutdown`

- Apaga el sistema en modo seguro.
- Opciones
 - t sec → Espera t segundos antes de inciar el proceso de cierre del sistema
 - r → reinicia (reboot) después de *shutdown*.
 - h → Paro del sistema (halt) después de *shutdown*.

`$ reboot`

- Reinicia el sistema.

`$ halt`

- Para el sistema.

`$ logout`

- Sale del *shell* y devuelve a *login*

`$ exit`

- Sale del *shell*

Redirección

El *kernel* abre para cada orden tres archivos

- `stdin` → entrada estándar: teclado
- `stdout` → salida estándar: pantalla
- `stderr` → salida de error: pantalla

Los operadores de redirección permiten cambiar las E/S estándar.

Redirección de salida

```
$ orden > fichero
$ orden >> fichero
```

Redirección de la salida de errores

```
$ orden 2> fichero
$ orden 2>> fichero
```

Redirección de entrada

```
$ orden < fichero
```

Redirección de la salida estándar y de errores

```
$ orden &> fichero
```

Tuberías (*pipes*)

Conecta la salida estándar de un comando con la entrada estándar de otro. Podemos conectar más de dos órdenes

```
$ orden1 | orden2
```

Bifurcación o T (comando `tee`)

```
$ orden | tee fichero
```

- La salida de un comando se redirige a un fichero y se bifurca a la terminal.
- `-a` añade la salida al fichero.

Ejecución en *segundo plano*

Linux es un sistema multitarea, es decir, permite ejecutar programas de forma concurrente.

La ejecución en segundo plano, ejecuta el proceso dejando el shell libre para atender nuevas órdenes.

El sell y el proceso que realiza la orden se ejecutan concurrentemente.

```
$ orden &
```

Los comandos relacionados con las tareas en segundo plano son:

- **jobs** → muestra los procesos en segundo plano
- **fg** → pasa a primer plano
- **bg** → pasa a segundo plano
- **Ctrl+Z** → paraliza un proceso (envía la señal SIGTSTP → 20)

Variables de entorno del shell

Se utilizan para guardar información del sistema y del usuario

- **HOME**
 - Directorio personal.
- **HOSTNAME**
 - Nombre de la máquina.
- **PATH**
 - Lista de directorios donde buscar los programas.
- **PS1**
 - Establece el *prompt* primario del sistema.
 - Permite modificar el prompt insertando ciertos caracteres especiales:
 - * `\a` → caracter de campana
 - * `\d` → la fecha en formato "día mes día"
 - * `\h` → el nombre del host hasta el primer `.`
 - * `\t` → la hora actual en formato 24-horas
 - * `\u` → el nombre de usuario del usuario actual
 - * `\w` → el directorio actual de trabajo
- **SHELL**
 - Intérprete de comandos por defecto.
- **TERM**
 - Tipo de terminal.
- **USER**
 - Nombre del usuario.