

Comandos del shell

Administración de Sistemas y Redes

Universidad de Oviedo

El shell de Linux

- Proporciona la forma básica de operar con ficheros y directorios
- En Linux se llama "bash" (hay otros como el "sh" o el "csh")
- El "prompt" es el símbolo del dólar aunque se puede cambiar. Para el usuario root suele ser la almohadilla (#)
- Proporciona además otra serie de funcionalidades para controlar los procesos (programas en ejecución) que hay en un servidor Linux
- Se inicia la sesión en el servidor desde otro PC con el putty.exe (existen también otras formas de iniciar sesión)
- Para cerrar la sesión pulsar las teclas "<Control> D"

El editor "vi"

- Es un editor nativo de Linux y de casi todos los sistemas operativos de tipo Unix
- Hay otros editores como vim, pico, nano, joe o emacs pero pueden no estar instalados
- Los editores de texto permiten crear y modificar ficheros que contienen texto no binario. Pueden ser notas, datos o programas escritos en un lenguaje de programación
- El "vi" en Linux viene a ser el equivalente al block de notas en Windows

El editor "vi": Modos

- El editor tiene dos modos de operación: modo comando y modo inserción.
 - En **modo inserción** aparece la palabra -- INSERT-- abajo del todo
 - En modo inserción todo lo que se teclea queda en el "buffer" de edición. Funcionan también las teclas de movimiento de cursor y de borrado
 - En **modo comando** interpreta unas órdenes y ejecuta la orden dada
 - Para pasar de modo inserción a modo comando hay que pulsar la tecla <Esc> (está arriba a la izquierda en casi todos los teclados)
- Ejemplo de uso para crear o modificar el fichero "prueba.txt":
\$ vi prueba.txt

El editor "vi": Comandos

i (pasa de modo comando a modo inserción insertando donde el cursor)

a (pasa de modo comando a modo inserción insertando detrás del cursor)

A (lo mismo pero inserta al final de la línea donde está el cursor)

x (borra el carácter donde está el cursor)

dd (borra la línea donde está el cursor)

:wq (salimos del editor guardando los cambios)

:q! (salimos del editor sin guardar los cambios)

/abc (busca la cadena "abc" y pone el cursor donde aparezca)

/ (busca el siguiente lugar donde aparece la cadena anterior)

Manejo de directorios (carpetas)

\$ pwd (muestra el directorio en curso)

\$ mkdir prueba (crea un subdirectorio "prueba")

\$ rmdir prueba (borra el directorio "prueba" que debe estar vacío)

\$ cd prueba (nuestro directorio de trabajo va a ser "prueba")

\$ cd .. (salta un nivel hacia la raíz en el arbol de directorios)

\$ cd (nos lleva a nuestro directorio "home" de usuario)

\$ ls (listado simple, sólo nombres)

\$ ls -la (listado completo, protecciones, nombres, fechas, tamaños)

Manejo de ficheros (archivos)

\$ cat fich.txt (muestra por pantalla "fich.txt")

\$ more fich.txt (lo mismo, pero cada 24 líneas se detiene para poder verlo, hay que pulsar un espacio para seguir)

\$ cp f1.txt otro.txt (copia "f1.txt" a "otro.txt")

\$ mv f1.txt otro.txt (cambia de nombre "f1.txt" a "otro.txt")

\$ mv f1.txt prueba/otro.txt (mueve "f1.txt" al subdirectorio "prueba" y le cambia el nombre poniéndole "otro.txt")

\$ rm fich.txt (borra el archivo "fich.txt")

Compilación y ejecución

```
$ cc test.c -o test (compila "test.c" y crea el ejecutable "test")  
$ gcc test.c -o test (lo mismo pero con el compilador C de GNU)  
$ g++ test.c -o test (lo mismo pero con el compilador C++ de GNU)  
$ ./test (ejecuta el programa "test")
```

<Control> C (detiene la ejecución del programa si éste todavía no acabó)

```
$ (debe aparecer el "$" después de "<Control> C")
```