

# UNIX

## INTRODUCCIÓN

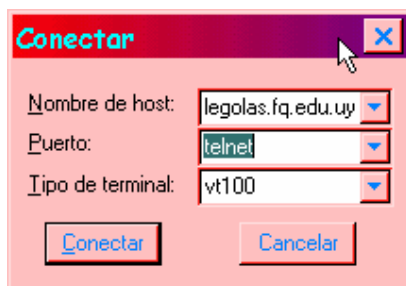
Estrictamente, UNIX no es un sistema operativo simple, sino una familia de sistemas operativos multiusuario y multitarea. Comprende el núcleo del sistema operativo más un conjunto de programas que permiten compilar lenguajes de programación, editar textos, interpretar comandos, manejar archivos y discos, acceder a otras máquinas, establecer comunicaciones telefónicas, enviar y recibir correo electrónico, manejar las colas de impresión y muchas tareas más.

El sistema se basa en un núcleo llamado Kernel (el corazón de todo sistema Unix), que reside permanentemente en la memoria, y que atiende a todas las llamadas del sistema, administra el acceso a los archivos y el inicio o la suspensión de las tareas de los usuarios.

La comunicación con el sistema UNIX se da mediante un programa de control llamado SHELL. Este es un lenguaje de control, un intérprete, y un lenguaje de programación.

## ACCESO A UN SISTEMA UNIX

Para ingresar a un sistema UNIX remoto desde un PC se usa habitualmente un "emulador de terminal". Este programa permite al PC comportarse como una terminal (teclado y pantalla, sin procesador) conectada directamente a la máquina UNIX. **Telnet** es un medio mediante el cual puede iniciarse una sesión a través de la internet.



En el emulador de terminal se deben fijar los siguientes parámetros:

- Nombre en la red de la máquina a la que se quiere conectar (por ejemplo `legolas.fq.edu.uy`).
- Tipo de terminal que se emulará (vt100 es el más conocido).

El ingreso de usuario al sistema es un proceso que se denomina "login" y requiere dos tipos de información: un nombre de usuario y un password.

Cuando se inicia una sesión, aparece un prompt como el que se muestra a continuación

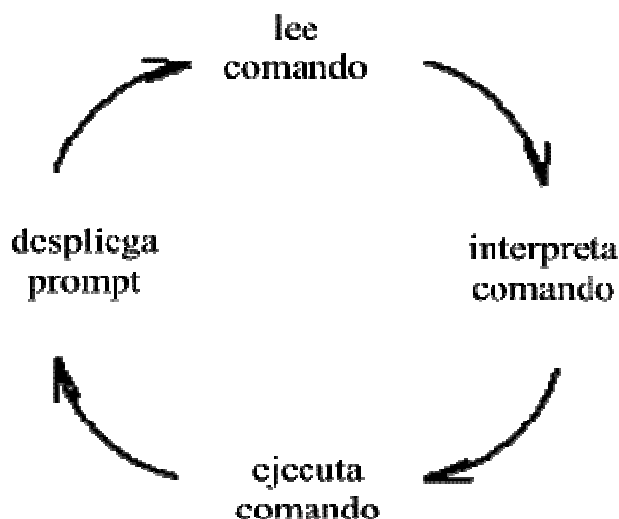
login:

Se debe escribir el nombre de usuario a continuación del prompt y presionar la tecla Enter. El sistema solicita al usuario el ingreso del password. Cuando se digita el password, la pantalla no mostrará lo que se escribe.

El nombre de usuario es el nombre identificador del usuario para ingreso al sistema y es asignado por la persona que crea la cuenta. Suele tener hasta 8 caracteres (letras, números y algunos símbolos permitidos) y debe comenzar con minúscula. **UNIX distingue entre mayúsculas y minúsculas** (case-sensitive). En el ingreso al sistema, no se permiten correcciones.

Para cerrar la sesión (login out) simplemente se debe digitar `exit`.

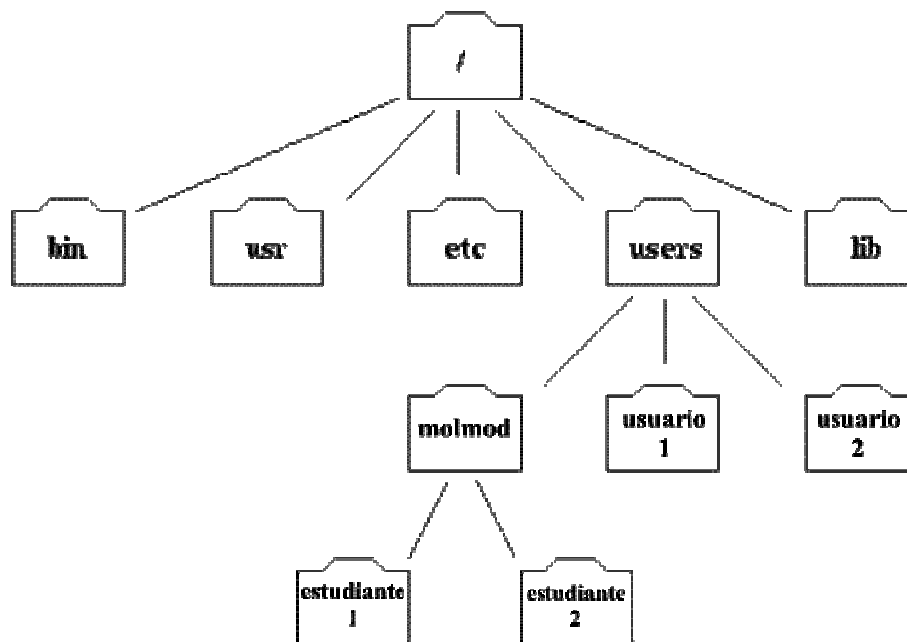
El SHELL es un tipo de programa llamado **intérprete**. Un intérprete opera en un ciclo simple: acepta un comando, lo interpreta, lo ejecuta y espera el ingreso de otro comando por parte del usuario.



El SHELL despliega un prompt o indicador de comandos (\$, %, o #) que indica que está listo para recibir un comando, y aguarda a que el usuario lo digite. Toda vez que el usuario digita un comando, el intérprete lo valida, y si es correcto crea un proceso hijo que realiza la acción encomendada. Al terminar el proceso hijo, exhibe nuevamente el indicador de comando e intenta leer una nueva línea de la entrada.

Toda la información en una máquina Unix, se almacena en un **sistema de archivos**. Cada vez que se inicia una sesión, el SHELL considera que el usuario se encuentra en algún lugar del sistema de archivos. Cada usuario tiene un directorio propio, llamado a veces "directorio home o directorio de trabajo". Cuando el usuario ingresa al sistema ya está ubicado en su directorio propio.

El sistema de archivos de Unix, es jerárquico (recuerda a la estructura de un árbol). El árbol está anclado en un lugar llamado raíz, designado por la barra "/". Cualquier objeto en el árbol del sistema de archivos es un archivo o un directorio. Un directorio es como un contenedor, que puede almacenar otros directorios o archivos. Un directorio o archivo contenido en otro directorio se conoce como "hijo" del que lo contiene. Un directorio en el sistema de archivos puede tener muchos hijos, pero solamente un directorio padre. Un archivo contiene información (un texto, un programa ejecutable, una imagen, sonido, etc.), pero no contiene a otros archivos o directorios.



Para describir un archivo o directorio específico en el sistema de archivos jerárquico, se debe especificar un camino (path). El camino a un determinado archivo o directorio debe ser definido como un *camino absoluto*, comenzando desde el directorio raíz, o como un *camino relativo*, comenzando desde la ubicación actual. Cuando se especifica un camino, simplemente se traza una ruta a través del árbol del sistema de archivos, listando la secuencia de directorios a través de los cuales se pasa para ir desde un punto al otro. Cada directorio listado en la secuencia se separa por una barra (/).

Por ejemplo, el camino absoluto desde el directorio raíz, hasta el directorio estudiante1 se especifica de la siguiente forma:

```
/users/molmod/estudiantel
```

## SINTAXIS DE LOS COMANDOS:

La forma básica de un comando Unix es:

**nombre\_del\_comando** [-opciones] [argumentos]

El nombre del comando es el nombre del programa que se desea que ejecute el SHELL. Las opciones, indicadas por un guión, permiten alterar el comportamiento del comando. Los argumentos son los nombres de archivos, directorios o programas a los que el comando debe acceder. Los paréntesis rectos, significan que son partes opcionales del comando que pueden ser omitidas. Así, cualquier opción o argumento entre paréntesis rectos es opcional, mientras que las que no se encuentran entre paréntesis recto deben ser digitadas. Las palabras que se encuentren en letras **negritas** deberán ser literales y se digitarán tal y como aparecen. Los argumentos que se encuentren en letras *cursivas*, deberán ser reemplazados por lo que sea que significan (usualmente el nombre de un archivo o directorio).

## cd

**cd** [*directorio*]

Cambia el directorio de trabajo.

Ej: `cd /users/molmod/estudiante1`

Sin argumentos cambia al directorio propio del usuario (directorio home) desde donde sea que se encuentre ubicado. Por ejemplo, si el directorio home del usuario es molmod y se encuentra en el directorio usuario2, simplemente digitando `cd` se cambiará al directorio molmod.

## clear

Borra la pantalla del terminal.

## cp

**cp** [*-opciones*] *archivo\_origen* *archivo\_destino*

Copia *archivo\_origen* hacia *archivo\_destino* en el mismo directorio.

**cp** [*-opciones*] *archivo\_origen* */directorio1/archivo\_destino*

Copia *archivo\_origen* hacia *archivo\_destino* en el directorio1 (que es un subdirectorio del raíz).

**cp** [*-opciones*] *archivo\_origen* */directorio1/archivo\_origen*

Copia *archivo\_origen* con el mismo nombre en el directorio1 (que es un subdirectorio del raíz). Por defecto no copia directorios.

*Opciones:*

**i**: avisa antes de sobrescribir archivos existentes.

**v**: escribe el nombre de cada archivo que copia.

## head

Muestra la primera parte de un archivo, por defecto 10 líneas.

**head** [*-opciones*] *archivo*

Opciones:

**N**: muestra las primeras N líneas.

**ls**

**ls** [-*opciones*] *nombre*

Para cada nombre de directorio, lista contenido de directorio; para cada nombre de archivo, indica su nombre y datos. La salida está ordenada alfabéticamente por defecto. Sin nombre, lista el directorio corriente

Opciones:

**a**: lista todos los archivos, incluso no visibles comenzados por . (punto).

**F**: indica tipo: / directorio, \* ejecutable, @ enlace simbólico

**k**: tamaños en KB.

**l**: (ele) listado en formato largo (permisos, cantidad de enlaces hard, dueño, grupo, tamaño, mes, día, hora o año, nombre).

**t**: ordena por fecha de última modificación.

**mkdir**

**mkdir** [-*opciones*] *directorio*

Crea los directorios indicados.

Opciones:

**p**: crea primero todos los directorios padre inexistentes.

**more**

**more** *archivo*

Pagina el texto del archivo dividiéndolo en pantallas, presentando una por vez. Durante el despliegue reconoce los comandos siguientes:

ESPACIO: avanza una pantalla

ENTER: muestra la siguiente línea

f: avanza una pantalla

b: retrocede una pantalla

/PATRÓN: busca hacia delante la expresión PATRÓN

?/PATRÓN: busca hacia atrás la expresión PATRÓN

q: sale

## mv

```
mv [-opciones] archivo_origen archivo_destino
```

Cambia el nombre de archivo\_origen por archivo\_destino.

```
mv [-opciones] archivo_origen /directorio1/archivo_origen  
mv [-opciones] archivo_origen /directorio1/archivo_destino
```

Mueve archivo\_origen hacia el directorio1 (subdirectorio del directorio raíz) con el mismo nombre en el primer caso y con diferente nombre (archivo\_destino) en el segundo caso.

*Opciones:*

**i**: avisa antes de sobrescribir archivos existentes.

**v**: escribe el nombre de cada archivo que copia.

## pwd

Despliega la ruta del directorio corriente.

## rm

```
rm [-opciones] nombre
```

Elimina los archivos o directorios indicados. Por defecto no elimina directorios.

*Opciones:*

**i**: interactivo, pregunta antes de eliminar cada archivo.

**v**: muestra el nombre de cada archivo que elimina.

**r**: recursivo, borra directorios y su contenido.

## tail

```
tail [-opciones] archivo
```

Muestra la última parte de un archivo, por defecto 10 líneas.

*Opciones:*

**N**: muestra las últimas N líneas.