



# Unidad 2 – Clase 2

## Sistemas Operativos



# Línea de comandos en sistemas tipo Unix

- En los sistemas tipo Unix se puede trabajar interactivamente en dos modos diferenciados: modo gráfico y modo texto (línea de comandos).
- Normalmente, los modos gráficos son los más utilizados en sistemas de escritorio, mientras que los de texto son adecuados para servidores. No obstante, se puede cambiar fácilmente de uno a otro con una secuencia de teclas (`Ctrl + Alt + tecla de función`).
- Incluso, en modo gráfico, se puede abrir una ventana especial llamada **terminal**, que permite trabajar en modo texto dentro del modo gráfico.

# Línea de comandos

- La interfaz que permite a los usuarios comunicarse con el kernel se denomina línea de comandos o **shell**.
- El shell puede ser escogido por el usuario entre los diferentes que hay. Algunos de los más conocidos son *bash* y *sh*.

¿Cuál estoy usando? `echo $SHELL`

# Ventajas de la línea de comandos

- Aunque al principio pueda parecer anticuado y complicado, pero cuando se adquiere un poco de práctica resulta muy **potente** y **eficiente**.
  - Una interfaz gráfica es sumamente útil para trabajos con gráficos pero es extremadamente ineficiente para el resto de las tareas.
- Permite la ejecución de pequeños programas llamados **shell scripts**, que son muy potentes para realizar secuencias de comandos (que pueden llegar a ser muy complejas).
- Permite trabajar sobre una máquina **remota**.

# Trucos del shell

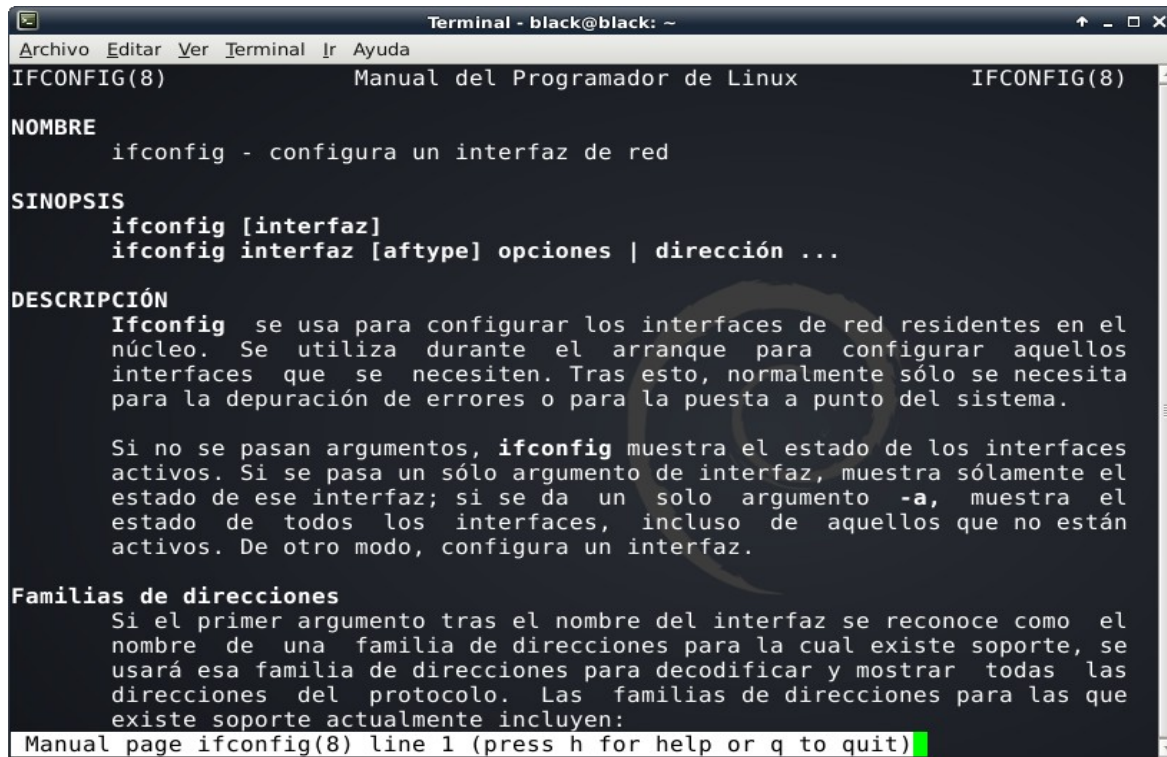
- Permite la repetición de comandos ya que se guarda la historia de los comandos utilizados.
  - ◆ Con flecha hacia arriba/abajo se recorre cada comando utilizado.
  - ◆ `history` imprime los últimos comandos que hemos ejecutado y los numera. Para ejecutar uno de esos comandos, se pone `!` seguido del número.
  - ◆ La combinación de teclas `Ctrl + r` permite buscar un comando anterior que contenga una cierta cadena de texto.
    - Con `Esc` selecciona para editar o `Enter` ejecuta.
    - Con `Ctrl + r` busca otra opción hacia atrás

# Trucos del shell

- Tiene texto predictivo (autocompletado) de los comandos y rutas (a archivos y directorios). Para autocompletar presionar tecla tabulador. Si no se autocompleta (hay más de una opción para completar), hay que apretar nuevamente la tecla tabulador para que aparezcan las opciones.
- Si estamos en modo gráfico, para salir de la terminal se puede utilizar el comando `exit`.
- Para borrar la pantalla:
  - `clear` borra la pantalla.
  - `reset` reinicia completamente la terminal.
- `Ctrl + c` detiene la ejecución de un comando.

# Comando man y apropos

- man muestra información relacionada a un comando:  
`man nombre_comando`



```
Terminal - black@black: ~
Archivo Editar Ver Terminal Ir Ayuda
ifconfig(8) Manual del Programador de Linux ifconfig(8)

NOMBRE
  ifconfig - configura un interfaz de red

SINOPSIS
  ifconfig [interfaz]
  ifconfig interfaz [aftype] opciones | dirección ...

DESCRIPCIÓN
  Ifconfig se usa para configurar los interfaces de red residentes en el
  núcleo. Se utiliza durante el arranque para configurar aquellos
  interfaces que se necesitan. Tras esto, normalmente sólo se necesita
  para la depuración de errores o para la puesta a punto del sistema.

  Si no se pasan argumentos, ifconfig muestra el estado de los interfaces
  activos. Si se pasa un sólo argumento de interfaz, muestra sólomente el
  estado de ese interfaz; si se da un solo argumento -a, muestra el
  estado de todos los interfaces, incluso de aquellos que no están
  activos. De otro modo, configura un interfaz.

Familias de direcciones
  Si el primer argumento tras el nombre del interfaz se reconoce como el
  nombre de una familia de direcciones para la cual existe soporte, se
  usará esa familia de direcciones para decodificar y mostrar todas las
  direcciones del protocolo. Las familias de direcciones para las que
  existe soporte actualmente incluyen:

Manual page ifconfig(8) line 1 (press h for help or q to quit)
```

## ¿Que información muestra?

- Nombre y descripción corta.
- Sinopsis (formato).
- Descripción del comando.
- etc.

## ¿Cómo salir del man?

Al igual que muchos otros comandos, hay que presionar la tecla “q” (de quit).

- ¿Cómo encontrar comandos?  
`apropos palabra_clave`

# Comando para conocer la jerarquía del sistema de archivos

- `man hier` describe cada uno de los directorios que aparecen creados en la jerarquía del sistema de archivos.



# Comando para conocer la ubicación de los programas que implementan los comandos

- `echo $PATH` imprime las rutas de directorios en donde se encuentran los programas ejecutables (también conocidos como archivos “binarios”). Algunos son: `/bin`, `/usr/bin`, `/sbin` (se describen en `man hier`).
  - Los comandos no son más que programas ejecutables que están en alguno de esos directorios. Si hay más de un ejecutable con el mismo nombre, se ejecuta el programa que está en el directorio de más a la izquierda en la lista.
- ¿Cuál es la ruta absoluta del archivo ejecutable correspondiente a un comando?

`whereis comando`

Reemplazar comando por el nombre del comando.

# Ubicación de archivos de usuarios

- Cada usuario tiene acceso a un directorio ubicado dentro de `/home`
- En cada directorio de usuario se almacenan archivos de:
  - videos, música, imágenes, descargas, etc.
  - configuraciones propias del usuario, por ejemplo: los colores que eligió para las ventanas, el idioma preferido, etc. Las carpetas y archivos de configuración normalmente comienzan con “.”.

# Comandos para archivos y directorios

- Mostrar ruta absoluta del directorio actual: `pwd`
- `ls` lista archivos y directorios:
  - opción `-a` muestra archivos ocultos, es decir, cuyos nombres comienzan con “.”
  - opción `-l` muestra más información
  - las opciones se pueden combinar:  
`ls -l -a` o `ls -la`
  - uso de “expresiones regulares”. El uso más típico es el del comodín “\*”, por ejemplo:  
`ls hola*.pdf` mostrará todos los archivos/directorios como `hola.pdf`, `holamundo.pdf`, `holachau.pdf`, etc.

# Comandos para directorios

- Cambiar directorio: `cd directorio`
- Crear directorio: `mkdir directorio`
- Borrar directorio vacío: `rmdir directorio`
- Borrar directorio lleno: `rm -r directorio` . La opción “r” significa “recursivamente”<sup>1</sup>, es decir, todo su contenido.  
Para borrar todo el contenido del directorio actual: `rm -r *`  
El “\*” es un comodín que significa “todo”.

En *directorio* hay que poner la ruta (absoluta o relativa) del directorio.

<sup>1</sup> RAE: proceso que se aplica de nuevo al resultado de haberlo aplicado previamente.

# Comandos para archivos y directorios

- Copiar archivos:

- `cp archivo_origen archivo_destino`

- Ejemplo: `cp a.txt b.txt`

- `cp múltiples_orígenes directorio_destino`

- Ejemplo: `cp a1.txt a2.txt hola/`

- opción `-r` se usa para copiar directorios recursivamente.

- Ejemplo: `cp -r hola hola1`

- Eliminar archivos:

- `rm archivo`

- En `archivo` hay que poner la ruta del archivo.

- Mover archivos o directorios:

- `mv origen destino`

- Funciona como usar `cp` y luego eliminar el origen.

# Comandos para archivos

- Imprimir contenido “en línea”:  
`cat archivo`
- Visualizar el archivo “fuera de línea”:  
`less archivo`
- Editar (o crear) archivo:
  - `nano archivo` (fácil y poco potente)
  - `vim archivo` (difícil y potente)
- Determinar tipo de archivo: `file archivo`

# Comandos para archivos

- Buscar archivos en **todo el árbol** de directorios:

`locate archivo`

La búsqueda es rápida porque utiliza una base de datos de nombres de archivos y ubicaciones previamente armada.

- Buscar archivos en una **rama específica** del árbol:

`find directorio -name archivo`

*Nota: busca en cada directorio de la rama*

# Bibliografía

- Páginas del manual (man)
- <http://es.wikipedia.org/>
- Administración de Sistemas GNU/Linux, 4ed 2014.  
[https://www.researchgate.net/publication/276920698\\_Administracion\\_de\\_Sistemas\\_GNULinux\\_4ed\\_2014](https://www.researchgate.net/publication/276920698_Administracion_de_Sistemas_GNULinux_4ed_2014)