

COMANDOS BÁSICOS DE BASH

Carlos Eduardo Martinez Núñez
Departamento de Física
Universidad de Sonora

September 1, 2017

1 ¿Qué es la linea de comando a terminal?

Es una interfaz del sistema de basada en texto, donde podras entrar comandos y recibir respuesta en texto tambien.

1.1 ¿Qué es Bash?

Es una mascara para la terminal que determina como luce y se comporta al usarla. Es un intermediario entre el usuario y el sistema.

2 Atajos básicos en Bash

Cuando se introducen comandos en Bash, existen un par de atajos claves que permiten ahorrar tiempo:

- **Teclas direccionales (Arriba, abajo):** Se pueden usar para acceder a los comandos usados c
- **Tecla tab:** Permite realizar un autocompletado del comando o camino a digitar.

3 Navegación en Bash

Existen unos comando básicos que ayudan para explorar archivos y carpetas en Bash. Algunos de ellos son:

- **pwd:** Muestra la ruta hasta al posición del directorio actual en Bash, ejemplo: pwd (forma básica).
- **ls:** Muestra el contenido del directorio actual, ejemplo: ls (forma básica).
- **ls -l:** Muestra el contenido del directorio enlistados por categorias.

- **las -a:** Muestra todos los documentos hasta los ocultos en un directorio.
- **file:** Permite obtener información sobre el tipo de archivo de un archivo o directorio, ejemplo: `file` (nombre de archivo o camino).

4 Tipos de rutas en Bash

La ubicación de un archivo o directorio en Bash, puede especificarse de dos formas:

- **Ruta Absoluta:** Se refiere a la ubicación de un directorio o archivo en relación con el directorio home de un usuario específico, ejemplo: `ls /home/caedumanu/Documents`.
- **Ruta relativa:** Indica la ubicación del directorio o archivo en relación a la ubicación actual del usuario, ejemplo: `ls Documents`.

5 Manual de Bash

Bash provee un manual con el que se puede consultar para aclarar dudas sobre el uso de los comandos. La forma de utilizarlo es:

- **man:** Permite obtener información de ayuda sobre un comando en particular, ejemplo: `man mkdir`.
- **man -k:** Realiza una búsqueda y enumera todos los comandos relacionados con la palabra clave, ejemplo: `man -k copy`.

6 Trabajando con directorios

Los siguientes comandos básicos corresponden a las acciones más usuales al trabajar con directorios:

- **cd:** Permite cambiar de directorios, ejemplo: `cd Music`. Si se coloca sin argumentos lleva al directorio raíz, ejemplo: `cd`. Se puede utilizar en combinación con las distintas rutas mencionadas antes, ejemplo: `cd/home/caedumanu/Desktop` ó `cd Desktop`.
- **mkdir:** Crea un directorio con el nombre especificado, ejemplo, `mkdir Notas`. Se puede usar rutas absolutas o relativas para crear los directorios en la posición deseada.
- **mkdir -p:** Crea directorios dentro de otros directorios, ejemplo: `mkdir -p /caedumanu/Notas/Notas2`, permite crear el directorio `Notas` y dentro de él `Notas2`.
- **mkdir -pv:** Crea los directorios uno dentro de otros indicando los detalles que se llevaron a cabo en el proceso.

- **rmdir**: Elimina el directorio seleccionado, ejemplo: `rmdir Notas2`. Se puede usar rutas absolutas o relativas.
- **rmdir -p**: Realiza una función análoga a la de `mkdir`.
- **rmdir -r**: Elimina al directorio con todo su contenido, ejemplo: `rmdir -r Notas2`.
- **touch**: Crea un archivo en blanco dentro de un directorio dado, ejemplo: `touch prueba1`.
- **cp**: Copia el archivo seleccionado a la dirección especificada, ejemplo: `cp prueba1 Desktop`. También puede copiar el contenido de un archivo a otro de nombre dado, ejemplo: `cp prueba1 prueba2`. Puede usarse con rutas absolutas o relativas.
- **cp -r**: Realizada una copia de todo el contenido dentro del directorio especificado al destino, ejemplo: `cp Notas Documents`.
- **mv**: Mueve archivos o carpetas al destino seleccionado. No requiere el uso de `-r`. Ejemplo: `mv Notas Desktop`. Se puede usar este comando también para cambiar el nombre de un archivo en la misma ubicación, ejemplo: `mv Notas Notas2`.
- **rm**: Permite eliminar un archivo, ejemplo: `rm prueba1`.

7 Editor de texto "vi"

Es un editor básico de texto en Bash. Podemos usar el modo de entrada de texto, para introducir contenido, o el modo de edición para realizar acciones de corrección como copiado, eliminado, búsqueda, sustitución y guardado. Algunos comandos básicos son:

- **vi**: Permite crear un archivo de texto, ejemplo: `vi notas1`.

Dentro del documento, existen varias opciones para la fácil manipulación y vagación, estas opciones son:

- **i**: Se entra en el modo de edición del documento.
- **ZZ (Mayúsculas)**: Se ejecuta al dar enter al acción de guardar y salir.
- **q!**: Al dar enter se descartan los cambios. Se obtienen solo los últimos cambios guardados.
- **w**: Al dar enter guarda el archivo, pero no se sale de él.
- **wq**: Al dar enter se guarda y se sale del documento.
- **cat**: Permite visualizar el contenido del archivo, ejemplo: `cat notas1`.

- **less:** Permite visualizar archivos grandes con facilidad moviéndote dentro de él con las teclas de dirección. Las paginas se pasan hacia adelante con la tecla espaciadora y hacia atrás con la tecla b. con q se sale del documento.
- **Ctrl c:** Esta combinación de letras permite salir de cualquier acción inapropiada de tecleo de comandos.

Las siguientes opciones permiten moverse con facilidad dentro de un documento de texto con vi:

- **Teclas direccionales:** Permite moverse en todas la direcciones.
- **Teclas j,k,h,l:** Permiten movimientos similares al de las teclas de dirección.
- **nG:** Mueve el cursor a la línea n especificada.
- **G:** Mueve el cursor a la línea n especificada.
- **w:** Mueve el cursor al inicio de la siguiente palabra.
- **nw:** Mueve el cursor n palabras hacia adelante en la línea.
- **b:** Mueve el cursor al inicio de la palabra anterior.
- **nb:** Mueve el cursor a la posición de n palabras hacia atrás.
- **x:** Elimina un carácter.
- **nx:** Elimina n caracteres.
- **dd:** Elimina la línea actual.
- **dn:** d seguido por un comando de movimiento. Elimina donde el comando lleve en la línea.
- **u:** Deshace la ultima acción.
- **U:** Deshace todos los cambios en la línea actual.

8 Tipos de permisos en Linux

Un archivo en linux puede tener uno, dos o tres de los siguientes permisos:

- **r (read):** Se puede ver el contenido del archivo solamente.
- **w (write):** Se puede cambiar el contenido del archivo.
- **x (execute):** Se puede ejecutar o correr un programa o script.

Tambien se puede especificar a las personas a las que se le concede permisos, como:

- **Propietario (owner):** La persona a quien pertenece el archivo.
- **Grupo (group):** Todo archivo perteneciente a un grupo.
- **Otros (others):** Cualquiera diferente al grupo a al propietario.

Los comandos usualmente usados para ver y cambiar los permisos, corresponden a:

- **ls -l:** Permite ver los permisos de la ruta en el camino absoluto o relativo especificado.
- **chmod:** Permite modificar el tipo de permiso del archivo en la ubicación dada.

Con respecto a los directorios, tambien se pueden otorgar permisos dados de la siguiente forma:

- **r:** El usuario tiene la habilidad de leer el contenido de un directorio.
- **w:** Tiene la habilidad de escribir en el directorio.
- **x:** Tiene la habilidad de entrar en el directorio.