

Introducción a la interfaz de línea de comandos

Universidad de Cuenca - Sistemas Operativos I

1. Objetivos

1. Tener una primera aproximación con la interfaz de línea de comandos de un sistema basado en Unix.
2. Adquirir la capacidad de realizar operaciones elementales con el emulador de terminal.
3. Contrastar las acciones que se llevan a cabo desde una interfaz gráfica a su equivalente desde la línea de comandos.

2. Instrucciones

2.1. Emulador gráfico de terminal

Busque el emulador de terminal en su sistema. Probablemente aparecerá listado como un programa más. Ejecútelo, familiarícese con él y examine las opciones disponibles en el menú de configuraciones.



En algunos sistemas el emulador (e.g., Ubuntu) de terminal se abre con `ctrl + alt + t`.



Su emulador de terminal puede tener una barra de menú o la capacidad de desplegar uno al hacer Click derecho con el ratón.

Ajuste las opciones disponibles a su gusto:

- Color de fondo
- Color de fuente
- Fuente
- Tamaño de fuente



El carácter `$` ("*prompt*") es usado para indicar que el texto a continuación es un comando.

2.1.1. Responda

- Escriba su nombre en la terminal y pulse **Enter**. ¿Qué ocurre?
- Ejecute el comando `$ ls` ¿Qué ocurre? ¿Cuál es la utilidad del comando?
- Ejecute el comando `$ clear` ¿Qué ocurre? ¿Cuál es la utilidad del comando?



Puede terminar la sesión de usuario utilizando el comando `$ exit`.



Puede cancelar (*interrumpir*) la ejecución de un comando pulsando la combinación de teclas **ctrl + x**.

3. Control del sistema de archivos

3.1. Desplazándose

- **pwd** (*Print Working Directory*): permite conocer en qué directorio se encuentra.
- **cd** (*Change Directory*): permite cambiar a un directorio específico.
- **ls** (*List*): permite listar el contenido de un directorio.

3.1.1. Responda

- Utilice el comando **\$ pwd** ¿En qué directorio se encuentra cuando inicia una nueva terminal?
- Ejecute **\$ cd ..** ¿En qué directorio se encuentra ahora?
 - Liste el contenido con **\$ ls** ¿Puede ver el directorio donde se encontraba al principio?
- Ejecute el comando **\$ cd** (sin argumentos) ¿Qué ha ocurrido?
- Muévase un par de directorios hacia atrás con **\$ cd ../../** ¿En qué directorio se encuentra ahora?
 - Ejecute **\$ cd ~** ¿Qué ha ocurrido? ¿Qué significa el argumento ~?
- Diríjase a la raíz del sistema de archivos (/) con el comando **\$ cd /** y liste su contenido. ¿Qué directorios le resultan familiares?
- Busque su carpeta personal en el entorno gráfico ¿Cómo puede llegar ahí usando la terminal? ¿Qué relación tiene con el argumento ~ de **cd**?

3.2. Tipos de rutas

Puede usar el comando **cd** especificando rutas **absolutas** o **relativas**. Una ruta **absoluta** empieza con el carácter **/**, es decir especifica la ruta exacta empezando por la raíz del sistema de archivos. Una ruta **relativa** empieza con el nombre del directorio directamente accesible desde la ubicación actual.

3.2.1. Responda

- Ejecute el comando `$ cd`
 - ¿Qué ocurre si ejecuta `$ cd bin`? Explique
 - ¿Cómo se ubicaría en el directorio **bin** del sistema?

3.3. Creación y control de directorios

- **mkdir** (*Make Directory*): crea un directorio.
- **rmdir** (*Remove Directory*): elimina directorios vacíos.
- **rm** (*Remove*): elimina archivos.

3.3.1. Responda

- Ubíquese en su directorio personal ¿Qué comando usó?
- Agregue un directorio de nombre "abc" ¿Qué comando usó?
 - Ubíquese dentro del directorio "abc"
 - Agregue un nuevo directorio de nombre "xyz"
 - Regrese a su carpeta personal e intente eliminar el directorio "abc" con `$ rmdir abc` ¿Qué ha ocurrido? ¿Por qué?
 - ¿Cómo eliminaría el directorio "abc"?
- ¿Qué comando(s) utilizaría para buscar un archivo en un directorio?
- ¿Qué comando(s) utilizaría para renombrar un archivo o un directorio?

3.4. Enlaces

Los enlaces permiten otorgar a un archivo o directorio múltiples nombres. Pueden ser de dos tipos:

- **Enlaces fuertes**: apuntan a los mismos datos sobre el sistema de archivos.
- **Enlaces simbólicos**: contienen la ruta del archivo que enlazan.

El comando usado para crear enlaces (fuertes o simbólicos) es `$ ln`.

3.4.1. Responda

- Agregue un archivo de nombre "abc" con el comando `$ touch abc`.
 - Escriba su nombre dentro de "abc" con el comando `$ echo "Nombre" > abc`.
 - ¿Qué tipo de archivo es "abc"?
 - Agregue un enlace fuerte a "abc" con el comando `$ ln abc cba`
 - ¿Qué tipo de archivo es "cba"?
 - Agregue un enlace simbólico a "abc" con el comando `$ ln -s abc sym`
 - ¿Qué tipo de archivo es "sym"?
 - ¿Qué puede decir sobre los enlaces fuertes y simbólicos basado en los tipos de archivos de "cba" y "sym"? Discuta.
- Elimine todos los archivos creados. ¿Qué comando usó?
- Agregue un directorio de nombre "dir". ¿Qué comando usó?
 - Agregue un enlace fuerte para "dir", de nombre "fuerte" con el comando `$ ln dir fuerte`. ¿Qué ocurrió?
 - ¿Puede crear un enlace simbólico de nombre "sym" para el directorio "dir"? ¿Qué comando usó?
 - ¿A qué conclusión puede llegar con esto? Discuta.
- ¿Para qué cree que son útiles los enlaces? Dé ejemplos puntuales.

3.5. Montaje de directorios

Lea la sección 10.6.1 *Fundamental Concepts* del subcapítulo *THE LINUX FILE SYSTEM* del libro de Tanenbaum y Bos (2014).

3.5.1. Responda

- Discuta el proceso de montaje de directorios en su sistema operativo (cómo se lo programa para que sea automático, qué hacer para modificar el proceso de montaje, etc).
- ¿Cuáles son los comandos más importantes en esta tarea?

4. Control del sistema

4.1. Apagando y reiniciando el sistema

- **shutdown**: apaga/reinicia el sistema de forma segura, informando a los usuarios conectados e impidiendo el ingreso de usuarios.



Asegúrese de guardar todo documento sensible antes de usar el comando `shutdown`.

4.1.1. Responda

- ¿Qué ocurre al ejecutar el comando `shutdown` sin argumentos?
- ¿Qué argumento es necesario para reiniciar el sistema en lugar de apagarlo?
- ¿Qué ocurre al ejecutar el comando `$ shutdown -t 10`?
 - ¿Qué utilidad se le ocurre para esto?

4.1.2. Apagado de sistema mediante un script

Para automatizar el apagado del sistema se propone utilizar el *Método 2*:
Programar el apagado del ordenador mediante un script explicado en la entrada de blog [Programar el apagado del ordenador](#).

Responda

- ¿Qué es un script?
- Haga una discusión acerca de la primera línea que se encuentra en el archivo `(#!/bin/bash)`.
- ¿Qué significa el número a continuación del comando `exit`? ¿Qué valores pueden utilizarse?
- ¿Qué otras opciones se pueden utilizar luego del comando `chmod` (a parte de `+x`)?
- ¿Se podría ejecutar el script con `sh apagar.sh`? Discuta.

4.2. Conclusiones



Escriba los principales resultados y conclusiones de su práctica.

4.3. Referencias

- Tanenbaum, A.S., Bos, H., 2014. Modern Operating Systems, 4th ed. Pearson, Boston, USA.
- Wolf, G., Ruiz, E., Bergero, F., Meza, E., 2015. Fundamentos de Sistemas Operativos, 1st ed. Universidad Nacional Autónoma de México, México D.F., México.



Liste aquí todas las referencias de las citas que haya puesto en el documento. Se recomienda, mas no se obliga, usar [Zotero](#).