

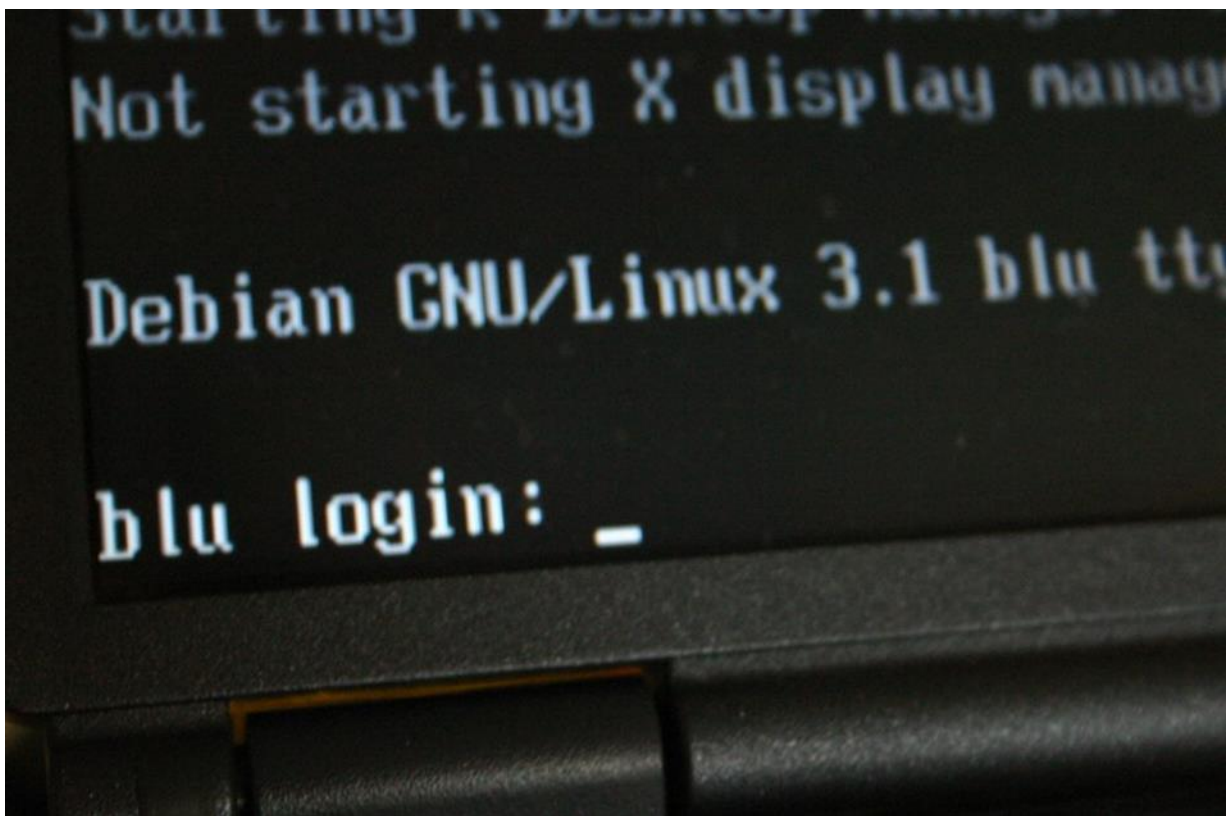


Qué es una terminal y los principales comandos de ella

Profesor: Omar Ruano

Alexia Miranda Limón

2b



-¿Qué es una terminal?

Terminal o también conocido como consola, es todo dispositivo electrónico que es parte del hardware de un ordenador, tiene la funcionalidad de ingresar y mostrar los datos que se encuentran dentro de una computadora en un determinado sistema de computación. Su finalidad es la de transformar estos datos en información que puede ser fácilmente percibida por los sentidos.

Las primeras terminales eran dispositivos económicos que buscaban dejar atrás las tarjetas perforadas pero eran demasiado lentos, la tecnología ha ido evolucionando y ha logrado mayor calidad y rendimiento.

Una terminal puede depender estrictamente del ordenador que le brinda su entorno y funcionamiento, siendo estos conocidos como clientes ligeros.

-COMANDOS

Cat

Cat (de concatenar), es una maravillosa utilidad que nos permite visualizar el contenido de un archivo de texto sin la necesidad de un editor. Para utilizarlo solo debemos mencionarlo junto al archivo que deseamos visualizar:

```
$ cat prueba.txt
```

Ls

Ls (de listar), permite listar el contenido de un directorio o fichero. La sintaxis es:

```
$ ls /home/directorio
```

El comando ls tiene varias opciones que permiten organizar la salida, lo que resulta particularmente útil cuando es muy grande. Por ejemplo, puedes usar `-a` para mostrar los archivos ocultos y `-l` para mostrar los usuarios, permisos y la fecha de los archivos. Así como para todos los comandos Linux, estas opciones pueden combinarse, terminando en algo como:

```
$ ls -la /home/directorio
```

Cd

Cd (de *change directory* o cambiar directorio), es como su nombre lo indica el comando que necesitarás para acceder a una ruta distinta de la que te encuentras. Por ejemplo, si estas en el directorio `/home` y deseas acceder a `/home/ejercicios`, seria:

```
$ cd /home/ejercicios
```

Si estás en `/home/ejercicios` y deseas subir un nivel (es decir ir al directorio `/home`), ejecutas:

```
$ cd ..
```

Touch

Touch crea un archivo vacío, si el archivo existe actualiza la hora de modificación. Para crear el archivo prueba1.txt en /home, sería:

```
$ touch /home/prueba1.txt
```

Mkdir

Mkdir (de *make directory* o crear directorio), crea un directorio nuevo tomando en cuenta la ubicación actual. Por ejemplo, si estas en /home y deseas crear el directorio ejercicios, sería:

```
$ mkdir /home/ejercicios
```

Mkdir tiene una opción bastante útil que permite crear un árbol de directorios completo que no existe. Para eso usamos la opción *-p*:

```
$ mkdir -p /home/ejercicios/prueba/uno/dos/tres
```

Cp

Cp (de *copy* o copiar), copia un archivo o directorio origen a un archivo o directorio destino. Por ejemplo, para copiar el archivo prueba.txt ubicado en /home a un directorio de respaldo, podemos usar:

```
$ cp /home/prueba.txt /home/respaldo/prueba.txt
```

En la sintaxis siempre se especifica primero el origen y luego el destino. Si indicamos un nombre de destino diferente, cp copiará el archivo o directorio con el nuevo nombre.

El comando también cuenta con la opción `-r` que copia no sólo el directorio especificado sino todos sus directorios internos de forma recursiva. Suponiendo que deseamos hacer una copia del directorio `/home/ejercicios` que a su vez tiene las carpetas `ejercicio1` y `ejercicio2` en su interior, en lugar de ejecutar un comando para cada carpeta, ejecutamos:

```
$ cp -r /home/ejercicios /home/respaldos/
```

Mv

`Mv` (de *move* o *mover*), mueve un archivo a una ruta específica, y a diferencia de `cp`, lo elimina del origen finalizada la operación. Por ejemplo:

```
$ mv /home/prueba.txt /home/respaldos/prueba2.txt
```

Al igual que `cp`, en la sintaxis se especifica primero el origen y luego el destino. Si indicamos un nombre de destino diferente, `mv` moverá el archivo o directorio con el nuevo nombre.

Rm

`Rm` (de *remove* o *remover*), es el comando necesario para borrar un archivo o directorio. Para borrar el archivo `prueba.txt` ubicado en `/home`, ejecutamos:

```
$ rm /home/prueba.txt
```

Este comando también presenta varias opciones. La opción `-r` borra todos los archivos y directorios de forma recursiva. Por otra parte, `-f` borra todo sin pedir confirmación. Estas opciones pueden combinarse causando un borrado recursivo y sin confirmación del directorio que se especifique. Para realizar esto en el directorio `respaldos` ubicado en el `/home`, usamos:

```
$ rm -fr /home/respaldos
```

Este comando es muy peligroso, por lo tanto es importante que nos documentemos bien acerca de los efectos de estas opciones en nuestro sistema para así evitar consecuencias nefastas.

Pwd

Pwd (de *print working directory* o imprimir directorio de trabajo), es un conveniente comando que imprime nuestra ruta o ubicación al momento de ejecutarlo, así evitamos perdernos si estamos trabajando con múltiples directorios y carpetas. Su sintaxis sería:

```
$ pwd
```

Clear

Clear (de limpiar), es un sencillo comando que limpiara nuestra terminal por completo dejándola como recién abierta. Para ello ejecutamos:

```
$ clear
```

Como *bonus* les recomiendo utilizar *man* que muestra una documentación completa de todos los comandos. Para *clear*, por ejemplo:

```
$ man clear
```