

# Lo elemental de la terminal de línea de comandos de GNU/Linux para usuarios de la UBMI

## Antes de comenzar

Esta no es una guía exhaustiva de uso de línea de comandos y hay muchas omisiones intencionales en cuanto a las opciones y uso de los programas, pero hemos recopilado aquí lo que nos parece es un buen punto de partida para hacer mas agradable el uso de la terminal de GNU/Linux

1. Entre [ ] se encuentran los argumentos que deben ser sustituidos por el usuario.
2. Los ejemplos en este documento van acompañados del símbolo \$ indicando que es texto en una terminal, pero este no es necesario copiarlo o escribirlo para reproducir los ejemplos.
3. No se muestran todas las opciones, sólo las que creemos son las más utilizadas.
4. En general, las opciones pueden combinarse por ejemplo: un programa con opciones -l -h -t -r puede ejecutarse cómo:

```
$ programa -l -h -t -r
```

ó como:

```
$ programa -lhtr
```

la excepción a esta regla ocurre cuando alguna de las opciones requiere un argumento adicional por el usuario, en el ejemplo anterior supongamos una opción -n que recibe un nombre, entonces se ejecutaría:

```
$ programa -lhtr -n [nombre]
```

5. Típicamente los programas contienen información que facilita su uso, en general si se utilizan las opciones -h ó --help se puede obtener información, adicionalmente se puede utilizar los comandos man

```
$ man [nombre del programa del que se quiere saber su uso]
```

ó tldr

```
$ tldr [programa]
```

6. Por último, en la terminal de línea de comandos de GNU/Linux suelen haber programas con opciones de una letra tal como: -h y opciones de mas de una letra cómo --help estas últimas las distinguimos por que tienen dos guiones en vez de uno, y a diferencia de las opciones de una letra estas no pueden ‘aplilarse’ como en el punto 4 de esta lista, pero pueden usarse juntas en combinación con otras opciones de una o mas letras.

## Contenido

- Uso de comodines ó *wildcards*
- Teclas amigas
- Uso de comandos útiles
  - Navegación
  - Manipulación de archivos
  - Sesiones locales remotas y red
  - Estado del equipo
    - \* Cockpit
    - \* Monitorix

## Comodines (*wildcards*)

En GNU/Linux podemos usar caracteres comodín para realizar operaciones sobre archivos o líneas de texto con parte del nombre en común, existen muchos tipos de comodín pero los que se utilizan con mas frecuencia son:

---

### Comodín

---

- \* Representa cualquier carácter, cualquier número de veces, por ejemplo \$ **rm \*txt** borraría todos los archivos cuyo nombre termine en “txt”, \$ **ls \*fotos\*** enumeraría los archivos y directorios que contengan la palabra “fotos”
  - ? Representa cualquier carácter **una** sola vez, en una carpeta con archivos llamados: “perros”, “gatos” “ratos” “platos” la operación \$ **ls ?atos** nos devolvería: **gatos ratos** pero no “platos”
  - ~ Un atajo que hace referencia a la carpeta “*home*” del usuario que la escribe, permite sustituir por ejemplo: \$ **cd /home/miusuario/Documentos** por \$ **cd ~/Documentos**
  - .. Otro atajo que hacer referencia a la carpeta anterior en el arbol de directorios; \$ **cd ..** evita que tengamos que escribir la dirección completa si solo queremos regresar un nivel y puede usarse en sucesión para regresar varios niveles: \$ **cd ../../..**
  - . Un atajo que representa el directorio actual
- 

## Teclas amigas

Existen varias teclas y combinaciones que nos facilitan la vida en la línea de comandos dentro de las más utiles están:

Nota: La tecla *Ctrl* también suele ser representada como  $\wedge$  en algunas documentaciones y archivos de ayuda

Tecla(s)	Función
Tab	Habilita el autocomple- tado al escribir comandos y paths (casi siempre)
Ctrl+ c	Interrumpe la ejecucion de un programa
Ctrl+ z	Manda a segundo plano la ejecución del programa, no debe confundirse con terminar el programa, es el equivalente de minimizar una pantalla
fg	Escribir \$ fg en la línea de comando revierte el efecto de Ctrl+z
Ctrl+a Ctrl+e	Nos llevan al inicio y al final de una linea de texto respectiva- mente
Teclas arriba y abajo	Permiten navegar entre los últimos comandos utilizados

Tecla(s)	Función
Ctrl+r	Nos permite buscar entre los comandos usados recién- temente tecleando una fracción del mismo

Tecla(s)	Función
>	<p>Permite escribir en un archivo de texto lo que de otra forma se representaría en pantalla como resultado de la ejecución de un programa, por ejemplo</p> <pre>\$ ls &gt; archivos.txt</pre> <p>creará un archivo dónde escribirá el resultado de la operación</p> <pre>\$ ls utilizar de nuevo \$ otro_comando &gt; archivos.txt</pre> <p>sobreescribirá el archivo, mientras que:</p> <pre>\$ otro_comando &gt;&gt; archivos.txt</pre> <p>agregará el resultado del programa “otro_comando” al final del archivo “archivos.txt”</p>

Tecla(s)	Función
\	El caracter pipe nos permite dar el resultado de un programa a otro programa por ejemplo: \$ ls -l \  wc -l primero ejecutará “ls -l” para listar los archivos de una carpeta y el texto resultante en vez de ponerlo en pantalla se lo dará al programa “wc -l” que sirve para contar líneas de texto y este imprimirá en pantalla un número que representará el numero de renglones de la primera operación, que en este caso se traduce en el número de archivos de la carpeta actual.

## Comandos útiles

Existen muchos comandos en GNU/Linux, aquí están los mas útiles para comenzar con lo que (creemos) son las mejores opciones para su uso.

### Navegación general

Comando	Descripción
<code>cd [/path/objetivo]</code>	Cambia de directorio a la dirección especificada
<code>cd ..</code>	Nos devuelve un nivel en el arbol de directorios
<code>cd -</code>	Nos regresa a la carpeta anterior en la que estábamos
<code>ls</code>	Enlista los contenidos de la carpeta actual
<code>ls [/path/objetivo/]</code>	Enlista los contenidos de la carpeta indicada
<code>ls -l</code>	ls en forma de lista
<code>ls -lh</code>	ls en forma de lista detallada y en formato mas legible
<code>ls -a</code>	ls enlistando también archivos ocultos
<code>ls *</code>	ls recursivo sobre los directorios de la carpeta actual

Comando	Descripción
<code>tree</code>	Funciona como un <code>ls *</code> pero con la recursividad se extiende hasta que no haya mas archivos que enlistar
<code>tree -fi</code>	Cambia el formato de “arbol” default por uno con el path completo de los archivos enlistados
<code>tree -fi \   grep [parte del nombre de un archivo]</code>	Permite buscar un archivo con parte del nombre y regresa el path completo
<code>find [path/objetivo] -name [parte del nombre de un archivo]</code>	Igual que el anterior
<code>pwd</code>	Nos dice en que directorio nos encontramos actualmente con un path absoluto
<code>history</code>	enlista los últimos n comandos que hemos utilizado (n varía en cada equipo)



Comando	Descripción
<code>history \  grep [parte de un comando]</code>	Nos permite buscar entre los últimos comandos usando una fracción del deseado, es mejor usar simplemente <code>Ctrl+r</code>
<code>tldr [nombre de un comando]</code>	Ayuda simplificada de uso de muchos comandos
<code>man [nombre de un comando]</code>	Ayuda en extenso de uso de muchos comandos

## Manipulación de archivos

Comando	Descripción
<code>mv [archivo] [/path/a/otra/ubicación]</code>	Mueve un archivo o carpeta de la carpeta actual a otro sitio
<code>mv [/path/a/un/archivo] [/path/a/otra/ubicación/]</code>	Mueve un archivo en otra carpeta diferente a la actual de un lugar a otro
<code>mv [archivo] [otro nombre]</code>	renombra un archivo de la carpeta actual, si la carpeta de origen es igual que la de destino y el nombre es distinto: <code>mv</code> renombra los archivos simplemente
<code>rm [archivo(s)]</code>	Elimina los archivos permanentemente <b>NO hay vuelta atrás</b>
<code>rm -rf [directorío]</code>	Elimina un directorio y todo su contenido recursivamente
<code>touch [archivo(s)]</code>	Crea archivos vacíos en la carpeta actual
<code>mkdir [directorío(s)]</code>	Crea directorios vacíos en la carpeta actual
<code>rmdir [directorío(s)]</code>	Elimina directorios vacíos
<code>grep "[patrón]" [archivo de texto]</code>	Devuelve las líneas que contienen un patrón de texto en un archivo
<code>grep "[patrón]" [archivo de texto] -i</code>	Devuelve las líneas que contienen un patrón de texto en un archivo ignorando mayúsculas y minúsculas

Comando	Descripción
grep "[patrón]" [archivo de texto] -v	Devuelve las líneas que <b>NO</b> contienen un patrón de texto en un archivo
cualquierComando \  grep "[patrón]"	Busca el patrón en el resultado de la ejecución de otro programa
cat [archivo]	Muestra en pantalla el contenido de cualquier archivo de texto
zcat	Muestra en pantalla el contenido de cualquier archivo de texto que esté comprimido
[archivo_comprimido]	
less [archivo]	Muestra en pantalla el contenido de un archivo pero permite desplazarse interactivamente y búsqueda
head [archivo] --lines [número]	Muestra las primeras líneas , especificadas en --lines en pantalla de archivos de texto
tail [archivo] --lines [número]	Muestra las últimas líneas , especificadas en --lines en pantalla de archivos de texto
nano [nombre de archivo existente o por crear]	Editor de texto para modificar archivos las opciones son interactivas y se presentan en la parte de abajo de la pantalla, la tecla Ctrl está representado con el carácter “^” en el menú

### Sesiones locales, remotas y red

Comando	Descripción
ssh [usuario]@[dirección IP]	Permite crear una sesión en una computadora remota a través de la línea de comandos
ssh [usuario]@[dirección IP] -p [número de puerto]	ssh especificando el puerto de red
sshfs	Monta una carpeta de un equipo en otro ,ver tutorial
scp [archivo local] [usuario]@[dirección IP]:[/path/remoto/]	Copia un archivo local en una computadora remota
scp [usuario]@[dirección IP]:[/path/remoto/] [/path/local]	Copia un archivo remoto en una carpeta local
wget [dirección web] [carpeta local]	Descarga un archivo desde la internet en la carpeta especificada
nohup [comando con sus opciones] &	Permite ‘dejar corriendo’ un programa sin riesgo de que se interrumpa por un fallo en la conexión

Comando	Descripción
byobu	Un modo interactivo de ‘dejar corriendo programas’ una vez dentro: F2 crea pantallas nuevas, F3 y F4 permite navegar entre pantallas, F6 permite salir <b>sin terminar los procesos</b> \$ <b>exit</b> en todas las pantallas es la manera de terminar con todos los procesos
exit	Termina la sesión local o remota y todos los procesos asociados

### Estado del equipo y procesos

Comando	Descripción
du -h [/path/objetivo]	Despliega el tamaño de todos los archivos y carpetas a partir del path objetivo
du -sh [path/objetivo]	Despliega sólo el total del tamaño que ocupa el path objetivo y su contenido
df -h	Informa sobre el uso de los discos duros las carpetas críticas en el caso de la UBMI son las identificadas como “/” y “/home”
free -h	Informa sobre el uso de RAM actual
nvidia-smi	En algunos equipos (workstations) informa del estado de las tarjetas gráficas
kill -9 [ID del proceso]	Para terminar un proceso, <b>se necesita el PID</b> (ver mas abajo)
ps ux	Enlista los procesos y PID del usuario que lo ejecuta
ps aux	Enlista todos los procesos y PID en el equipo
top	Visualización interactiva de procesos, uso de memoria y CPU
htop	Visualización interactiva ligeramente menos fea de procesos, uso de memoria y CPU
bpytop	Visualización interactiva significativamente menos fea de procesos, uso de memoria y CPU. (sólo workstations)

**Cockpit** Cockpit no es una aplicación de la línea de comandos es un servidor web de monitoreo de los equipos de la UBMI- IFC que permite conocer el estado actual de un equipo y consultar varios parámetros de funcionamiento de un equipo de manera gráfica.

Dentro de la red del instituto en un explorador web escribe: **[dirección ip del servidor]:9090**

- Este servicio no está disponible para UBMI-Portal
- Es altamente probable que el explorador web indique un riesgo de seguridad en acceder a Cockpit, simplemente hay que seleccionar que deseamos continuar al sitio web.
- Una vez ingresados los mismos usuario y contraseña que utilizas con ssh podrás acceder a la interfáz

Además del estado del equipo cockpit incluye un emulador de terminal con el cual podrás interactuar con el equipo.

**Monitorix** Otro servicio de monitoreo del estado de los equipos, permite conocer el estado actual y pasado de un equipo en cuestión, nuevamente se trata de un servidor web accesible sólo dentro de la red del IFC.

Dentro de la red del instituto en un explorador web escribe: **[dirección ip del servidor]:8080/monitorix**

- Este servicio no está disponible para UBMI-Portal
- No se requiere usuario ni contraseña para acceder

**NEMS** La UBMI cuenta con un servidor de monitoreo del estado de conexión de los servidores y workstations que administra, se pueden consultar las alertas y el estado de los servidores dando click aquí.

- Servicio disponible sólo dentro de la red del IFC
- No se requiere usuario ni contraseña para acceder