

Comandos Linux

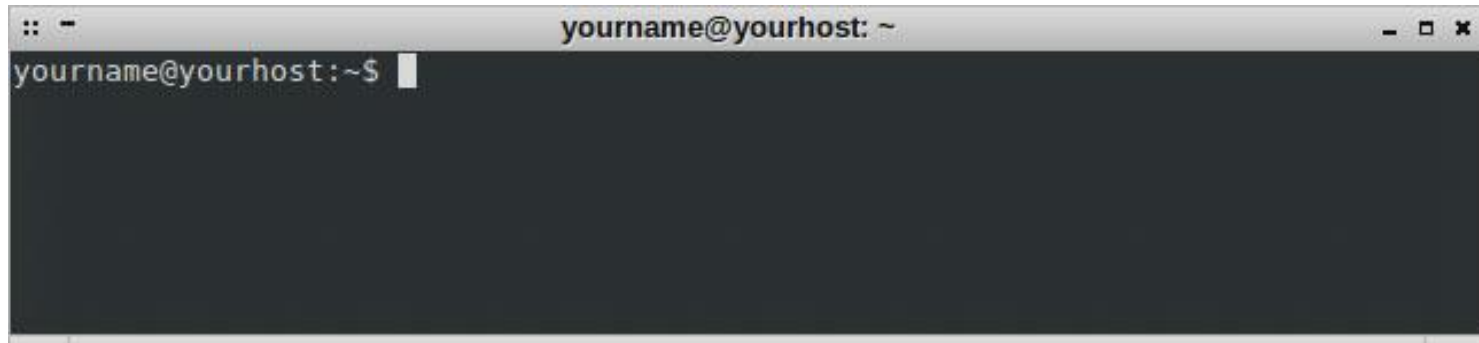
```

="hugo"
="$25 mai 2011 19:14:28$
rch(path,dir,i,taille): def search(path,dir,
ction principale. Paramètres : chemin du fichier, dossier de travail, iter
= path.replace(dir,"") def search(path,dir,taille):
g = name.replace(".avi","").replace(" ","").lower()
url = "http://www.alpouk.fr/recherche/?q={0}".format(string)
= urllib2.Request(the_url) string = name.replace(".avi","").replace(" ","")
.replace(".avi","").replace(" ","").lower()
dle = urllib2.urlopen(req)
cept IOError, e: string = name.replace(".avi","").replace(" ","").lower()
.avi,"").replace(" ","").lower()
i hasattr(e, 'reason'): echo "url: %s, error: %s" % (url, e.reason)
rint 'Nous avons échoué à joindre le serveur.'
rint 'Raison: %s' % e.reason
elif hasattr(e, 'code'):
print 'satisfaire la demande.' string = name.replace(".avi","").replace(" ","").lower()
print 'Code d' erreur: %s' % e.code
le read()

```

¿QUÉ ES PROMPT?

Es un carácter o conjunto de caracteres que se muestra en una línea de comandos para indicar que está a la espera de órdenes

A screenshot of a terminal window. The title bar at the top reads "yourname@yourhost: ~" and includes standard window control icons (minimize, maximize, close) on the right. The terminal content shows the prompt "yourname@yourhost:~\$" followed by a white cursor block, indicating it is ready for input.

```
yourname@yourhost:~$
```

Sintaxis de la línea de comandos

Cuando abres la terminal puedes ver un cursor en forma de texto al inicio de cada línea que muestra tu nombre de usuario y el nombre de tu equipo. En mi caso, se muestra lo siguiente:

```
Last login: Mon Sep 27 23:47:22 UTC 2021 from 192.168.168.61 on pts/0
You have new mail.
eocares@developers:~$ _
```

Esta estructura puede variar dependiendo que la distribución de Linux o de la terminal que utilices, pero será muy parecida. Esta es la sintaxis de lo que se muestra:

Usuario@Equipo:Directorio\$

Vamos a ver lo que es cada uno de estos elementos y a que se corresponden con lo que se muestra en vuestras pantallas:

Usuario: Es tu nombre de usuario en el equipo, que en mi caso es eocares.

@: Un símbolo separador que se suele utilizar para indicar que lo que viene tras el nombre de usuario es un equipo.

Equipo : Se trata del nombre que tiene tu equipo, que es el que le hayas dado al instalar Linux, si es que no lo has cambiado.

Directorio (~): Se trata del directorio en el que estás actualmente. Suele ser un nombre, pero es habitual utilizar el símbolo ~ para representar al directorio home.

Listo (\$): Un indicador para informar de que la línea de comandos está lista para aceptar nuevos comandos. En sistemas Unix se utiliza el símbolo \$, aunque en Windows se suele utilizar el símbolo >.

pwd

Significa Personal Working Directory te muestra la ruta del directorio en que estás ubicado actualmente.

```
1 $ pwd
2 /home/alejandro/EDteam
```

mkdir

Significa Make Directory y con él puedes crear una nueva carpeta con el nombre que indiques.

mkdir {nombre_de_la_carpeta}

```
1 $ mkdir mi_carpeta
```

touch

Con este comando puedes crear nuevos archivos en el directorio actual.

```
1 $ touch mi_archivo.txt
2 $ ls
3 mi_archivo.txt
```

ls

Este comando lista los archivos y directorios de la carpeta actual.

```
1 $ ls
2 mi_archivo.txt
3 mi_carpeta
```

Si quieres mostrar elementos ocultos, usa el flag -a.

```
1 $ ls -a
2 .soy_un_archivo_oculto
3 mi_archivo.txt
4 mi_carpeta
```

Para listar el contenido detallado usa el flag -l

cd

El comando cd (change directory) permite moverse entre directorios del sistema.

cd {ruta_absoluta_o_relativa}

Puedes cambiar de directorio especificando la ruta absoluta desde el directorio raíz o relativa desde tu ubicación actual, en Linux el directorio actual se indica con el signo .

```
1 $ pwd
2 /home/alejandro/EDteam
3
4 $ cd /home/alejandro/EDteam/mi_carpeta
5 $ cd ./mi_carpeta
6 $ cd mi_carpeta
```

mv

El comando mv (Move) mueve directorios o archivos de una ubicación a otra. Su sintaxis es:
`mv {ubicación_actual} {nueva_ubicación}`

```
1 $ mv mi_archivo.txt ./mi_carpeta/mi_archivo.txt
```

mv

También puedes renombrar archivos y carpetas con este mismo comando: `mv {nombre_actual} {nombre_nuevo}`

```
1 $ mv mi_archivo.txt usuarios.txt
2 $ mv mi_carpeta cursos
```

cp

El comando cp (Copy) copia archivos o directorios. Su sintaxis es cp {origen} {destino}

```
1 $ cp usuarios.txt usuarios_copia.txt
```

```
1 $ cp -r cursos/ cursos_copia/
```

rm

El comando rm (Remove) elimina archivos. Su sintaxis es rm {nombre_del_archivo}

```
1 $ rm -r cursos_copia/
```

cat

El comando cat te permite leer el contenido de archivos.

```
1 $ cat notas.txt
2
3 Estas son las notas de mi curso
4 Fin de las notas de mi curso
```

```
1 $ cat -n notas.txt
2
3 1 Estas son las notas de mi curso
4 2 Fin de las notas de mi curso
```

find

Con el comando `find` y la opción `-iname` puedes encontrar archivos. La sintaxis es: `find {donde_buscar} -iname {archivo_a_buscar}`

```
1 $ find . -iname "usuarios.txt"
2 ./cursos/usuarios.txt
```

Como habilitar su root por terminal

Se necesita los siguientes comandos:

su -i = me permite acceder como root

Sudo passwd root = me permite colocar una contraseña al root.

```
eocares@developers:~$ sudo -i
root@developers:~# exit
logout
eocares@developers:~$ sudo passwd root
New password:
Retype new password:
passwd: password updated successfully
eocares@developers:~$ su
Password:
root@developers:/home/eocares#
```