

Clase Linux - Copy

CONOZCAMOS EL SHELL DE LINUX

Temas a revisar

1. El Shell de UNIX.
 2. Archivos y directorios.(ls,pwd,mkdir, rmdir, rm cp, alias, cat. more)
 3. Permisos de archivos
 4. Combinando comandos y redireccionamiento de salida.
 5. Manipulacion de texto.
 6. Acceso remoto y transferencia de archivos
-

El shell de Linux

Ventajas y desventajas contra un ambiente gráfico.



GUI vs CLI

#1. Basic

GUI



This user interface enables user to interact with electronic device with the help of graphical icons and visual indicators.

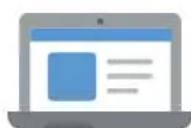
CLI



This user interface enables user to give command to interact with electronic device.

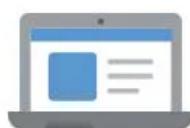
#2. Ease of understanding

GUI



Graphical user interface is visually intuitive. It is easy to understand for beginners.

CLI



Due to need of remembering commands, it is difficult to handle and requires expertise.

#3. Memory Requirement

GUI



CLI





It requires more memory as it consists of lot of graphical components.



It requires less memory as compared to GUI.

#4. Speed

GUI



It generally uses mouse to execute commands. The speed of GUI is Slower than CLI.

CLI



Because keyboard is used to execute the commands, the speed of the CLI is Faster than GUI.

#5. Appearance

GUI



One can change the appearance with customizable option.

CLI



It is not possible to change the appearance.

#6. Flexibility

GUI



More flexible than CLI.

CLI



Less flexible than GUI.

#7. Device used

GUI



Keyboard and mouse.

CLI



Keyboard.

#8. Precision

GUI



Low as compared to the CLI.

CLI



High as compared to the GUI.

www.educba.com

Ejecutando comandos

Cada vez que se arranca una terminal se presenta un prompt en pantalla, es el indicador que el shell esta esperando para recibir comandos a ser ejecutados.

```
jmanuel@conacyt:~$
```

Intentemos ejecutar nuestro primer comando, mostrando el directorio home:

```
jmanuel@conacyt:~$ls
```

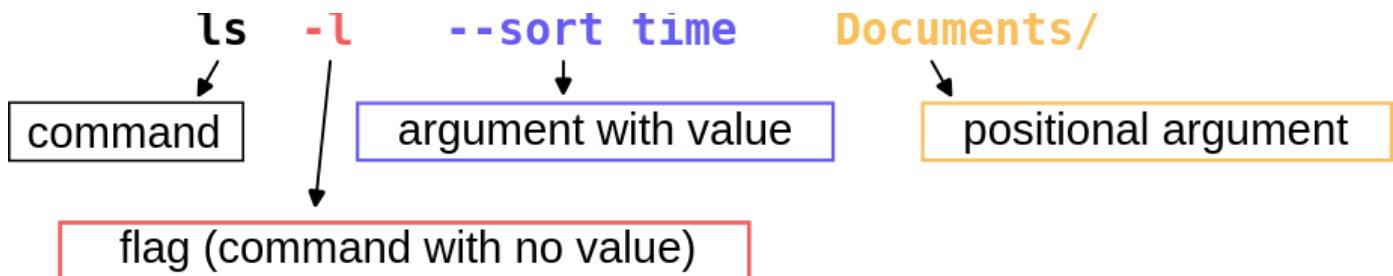
Salida:

```
jmanuel@conacyt:~$
```

La salida de este comando es:

```
jmanuel@conacyt:~$ ls
borra      Desktop    Downloads      Music        Public      test1
borraw     dir         eco.fna       nbc_hmms.hmm.gz R          test.py
cnamm      dirlelse   FastTree     nuevoarchivo.txt Recursos   UBCG_v3.zip
curso      docker     id_rsa.pub  otros_campylobacter Recursos.zip Videos
curjobash  docker2.tar Metagenomic_Course parallellexamples Pictures
curjobash.tar docker.tar.gz Metagenomic_Course_ori procesSeq.py  respaldo snap
DB.tar      Documents  midir        Templates    Templates  workshop xx.txt
jmanuel@conacyt:~$ |
```

Estructura de un comando.



Comandos más utilizados



Basic Linux Commands

ls
List files and dirs in current directory

cd
Change current directory

pwd
Display current directory path

man
Show manual page of a command

grep
Search for patterns in files or output string

find
Search for files in directory hierarchy

cat
Concatenate and display file contents

sudo
Run a command with superuser privilege

vi
Edit a text file from command line

ps
Display currently running processes

top
Display system processes dynamically

ip
Display and configure network interfaces

df
Report filesystem disk space usage

du
Show file and directory space usage

chmod
Change file permissions

chown
Change file owner and group

ssh
Remote login via secure shell

scp
Copy files between hosts over ssh



Created by
@dan_nanni
on Instagram



Ayuda

1. Podemos pasar el parametro `--help`
como

```
ls --help
```

2. Leer la ayuda con comando `man`

```
man ls
```

```
jmanuel@conacyt:~$ ls
borra      Desktop    Downloads       Music          Public        test1
borraw     dir        eco.fna        nbc_hmms.hmm.gz  R           test.py
cnamn      dirlefse   FastTree      nuevoarchivo.txt Recursos    UBCG_v3.zip
curso     docker     id_rsa.pub    otros_campylobacter Recursos.zip Videos
cursobash  docker2.tar Metagenomic_Course ori parallelexamples Pictures
cursobash.tar docker.tar.gz  Metagenomic_Course_ori midir      procesSeq.py  respaldo
DB.tar     Documents   snap          Templates      workshop
jmanuel@conacyt:~$ |
```

el manual presenta ayuda únicamente de los comandos del sistema, no presenta ayuda de programas instalados fuera del manejador de paquetes.

PRACTICA

ejecute los siguientes comandos y explique el resultado:

```
ls -l
ls -lt
lks -l
ls -ltr
ls -la
```

Trabajando con directorios y archivos.

Primero que nada es necesario conocer el arbol de directorios de los sistemas Linux



Linux Filesystem Hierarchy

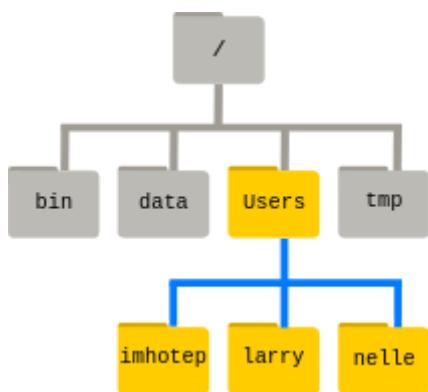


Created by @dan_nanni on Instagram



/	/bin	Essential commands needed for all users & sysadmins
	/boot	Files needed for boot (kernel, initrd, bootloader config)
	/dev	Device files representing hardware devices
	/etc	System-wide configs/scripts for the system and apps
	/home	Home directories for all users except the root
	/lib	Essential shared libraries and kernel modules
	/media	Mount point for removable devices (e.g. USB stick, CD)
	/mnt	Mount point for a temporarily mounted filesystem
	/opt	Third-party software and add-on software packages
	/proc	Virtual filesystem for kernel stats and process info
	/root	Home directory of the root user
	/run	Temporary filesystem (tmpfs) storing run-time data
	/sbin	Essential system binaries used for sysadmin tasks
	/srv	Data for system services (e.g. FTP, web, Git, Samba)
	/sys	Virtual filesystem for devices, drivers & kernel features
	/tmp	Temporary files created by apps
	/usr	Shareable read-only user utilities and applications
	/var	Dynamically generated files (e.g. logs, package cache)

Describiendo el árbol con la diagonal (slash)



/ es la raíz del sistema

/Users/larry es el directorio de inicio del usuario larry

/Users/nele es el directorio del usuario nele

Movernos en el árbol

El comando para movernos en el arbol de directorios será:

```
cd
```

Para mostrarnos en que directorio nos encontramos:

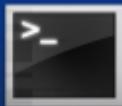
```
pwd
```

Con sus respectivos modificadores

```
cd nombredir  
cd rutaabsoluta  
cd rutarelativa  
cd ..
```

Paréntesis

El uso de ciertas combinaciones de teclas ayudan a acelerar el proceso de escribir los comandos más rápidamente.



Linux Terminal Shortcuts



Created by @dan_nanni on Instagram



Cursor Control	Ctrl+A	Move cursor to the beginning of the line
	Ctrl+E	Move cursor to the end of the line
	Alt+F	Move cursor forward by one word
	Alt+B	move cursor backward by one word
Text Control	Ctrl+U	Remove everything from the cursor to the beginning of the line
	Ctrl+K	Remove everything from the cursor to the end of the line
	Ctrl+W	Remove one word on the left of the cursor
	Alt+D	Remove one word on the right of the cursor
	Tab	Auto-complete the name of a command, a file or a directory
Job Control	Ctrl+C	Abort a foreground job by sending SIGINT signal
	Ctrl+Z	Stop a foreground job by sending SIGSTP signal
Session Control	Ctrl+L	Clear the current terminal session
	Ctrl+D	Exit the current terminal session
	Ctrl+Alt+T	Start a new terminal session
History Search	Ctrl+R	Reverse search through command history
	Ctrl+P	Go to the previous command in history
	Ctrl+N	Go to the next command in history
	Alt+. .	Use the last word of the previous command

PRACTICA

1. Encuentre el directorio `corrected` dentro de su home y describa el contenido, cuantos archivos comprimidos tiene, cuantos sin comprimir, cuantos directorios, cuantas veces encontró el directorio `corrected` en su arbol de directorios.

2. En cuentre el directorio `alkanes`

¿Qué otras formas existen para buscar los directorios en cuestión?

Crear y borrar directorios

El comando `mkdir` nos permite crear directorios mientras que `rmdir` nos permite directorios vacio.

```
mkdir nombredirectorio  
rmdir nombredirectorio
```

Trabajando con archivos

Copiando archivos:

El comando cp nos permite realizar copias de un archivo la sintaxis y ejemplo es el siguiente:

```
cp archivo destino
```

Ejemplo:

```
cp shell-lesson-data.zip resp.zip
```

Es posible utilizar rutas absolutas y relativas, por ejemplo:

```
cp /etc/passwd .  
cd cursobash/dircurso/shell-lesson-data  
cp Documents.tar ../curso/R/
```

Copiar un directorio completo con el modificador -R (verificar los modificadores posibles del comando cp con man)

```
cp -r shell-lesson-data/ backup-data
```

Eliminando archivos

El comando rm nos permite eliminar archivos como directorios que contienen archivos

```
rm archivo  
rm -i archivo  
rm -rf directorio
```

Se tiene que ser muy precavido y estar seguro con éste último comando porque es complicado recuperar archivos eliminados, y en un servidor en producción se tiene de detener todos los servicios para poder extraer el disco y recuperar los archivos.

Alias de comandos

En este punto es importante mencionar que es posible acortar la escritura de instrucciones mediante el comando alias

Para ver que alias tenemos definidos en el sistema ejecutar solo el comando:

```
alias
```

```
jmanuel@hplinux:~/Documents/BursoBash$ alias
alias alert='notify-send --urgency=low -i "$(([ $? = 0 ] && echo term;
; &|]\s*alert$//\''))"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias gal='ssh 132.248.32.20'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
jmanuel@hplinux:~/Documents/BursoBash$ |
```

el si

```
alias ls='ls -ltr --color=auto'
```

Donde definir los alias de comandos

Depende del shell utilizado por el sistema se puede definir que al inicio de la sesión, en este caso que nosotros estamos utilizando el shell bash, los comandos de inicio de sesión se ejecutan son los que estan en el archivo .bashrc

```
cat .bashrc
```

```
jmanuel@hplinux:~$ cat .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
```

Variables de entorno.

Las variables de entorno permiten cambiar el comportamiento de los programas, así como permiten proporcionarle datos al SHELL para la ejecución de programas.

Existen varias formas de conocer el contenido de las variables de entorno, si no sabe cuales variables estan definidas en el SHELL ejecutar el comando

```
env
```

```
jmanuel@hplinux:~/Documents$ env
SHELL=/bin/bash
SESSION_MANAGER=local/hplinux:@/tmp/.ICE-unix/2703,unix/hplinux:/tmp/.ICE-unix/2703
QT_ACCESSIBILITY=1
COLORTERM=truecolor
XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/etc/xdg
SSH_AGENT_LAUNCHER=gnome-keyring
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
GTK_IM_MODULE=ibus
LC_ADDRESS=es_MX.UTF-8
GNOME_SHELL_SESSION_MODE=ubuntu
LC_NAME=es_MX.UTF-8
SSH_AUTH_SOCK=/run/user/1000/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=ubuntu
LC_MONETARY=es_MX.UTF-8
GTK_MODULES=gail:atk-bridge
PWD=/home/jmanuel/Documents
LOGNAME=jmanuel
XDG_SESSION_DESKTOP=ubuntu
XDG_SESSION_TYPE=x11
GPG_AGENT_INFO=/run/user/1000/gnupg/S.gpg-agent:0:1
```

Si conoce cuales variables están definidas ejecutar el comando

```
echo NombreVariable
echo $PATH
```

```
~/usr/bin/env
jmanuel@hplinux:~/Documents$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/snap/bin
jmanuel@hplinux:~/Documents$ |
```

Definir y modificar variables de entorno

Para definir una variable de entorno solo escribir de la siguiente forma:

```
VAR="contenido variable"
```

Por convención se definen con mayúsculas los nombres de variables.

Conectando comandos y redireccionando las salidas.

Flujos de datos en unix.

Existen 3 flujos de datos cuando se ejecutan comandos en el SHELL

- STDIN Entrada estandar que suele ser el teclado

- SDTOUT salida estandar.
 - STDERR Es el flujo a donde se envían los errores





Redireccionamiento E/S a un archivo

Teclear los siguientes comandos en su terminal y explicar:
(Redireccionamiento <>)

```
ls  
ls 1> test  
lsx  
lsx 1> test  
lsx 2> test  
head /etc/passwd 1> cuentas
```

X

X

```
jmanuel@hplinux:~/Documents$ ls 1> test
jmanuel@hplinux:~/Documents$ lsx
Command 'lsx' not found, did you mean:
  command 'lsd' from snap lsd (0.16.0)
  command 'lex' from deb flex (2.6.4-8build2)
  command 'lex' from deb flex-old (2.5.4a-10.1)
  command 'lyx' from deb lyx (2.3.6-1)
  command 'sx' from deb lrzsz (0.12.21-10)
  command 'lsm' from deb lsm (1.0.4-2)
  command 'lsw' from deb suckless-tools (46-1)
  command 'ls' from deb coreutils (8.32-4.1ubuntu1)
  command 'lsc' from deb livescript (1.6.1+dfsg-2)
  command 'lsh' from deb lsh-client (2.1-13)
  command 'osx' from deb opensp (1.5.2-13ubuntu3)
See 'snap info <snapname>' for additional versions.
jmanuel@hplinux:~/Documents$ lsx 1> test
Command 'lsx' not found, did you mean:
  command 'lsd' from snap lsd (0.16.0)
  command 'sx' from deb lrzsz (0.12.21-10)
  command 'lex' from deb flex (2.6.4-8build2)
  command 'lex' from deb flex-old (2.5.4a-10.1)
  command 'lyx' from deb lyx (2.3.6-1)
  command 'lsw' from deb suckless-tools (46-1)
  command 'osx' from deb opensp (1.5.2-13ubuntu3)
  command 'lsc' from deb livescript (1.6.1+dfsg-2)
  command 'lsh' from deb lsh-client (2.1-13)
  command 'lsm' from deb lsm (1.0.4-2)
  command 'ls' from deb coreutils (8.32-4.1ubuntu1)
See 'snap info <snapname>' for additional versions.
jmanuel@hplinux:~/Documents$ lsx 2> test
```

Redireccionamiento a otro comando

Se puede redireccionar la salida de un comando hacia otro comando, para poder procesar la información que nos producen los comandos precedentes.

```
file * | grep directory | wc -l
```

Mirando el contenido de archivos.

Encuentre el directorio llamado alkanes

```
(base) root@conacyt:/home/jmanuel/cursobash/dircurs# ./shell-lesson-data/exercise-data/alkanes/propane.pdb  
./shell-lesson-data/exercise-data/alkanes/methane.pdb  
./shell-lesson-data/exercise-data/alkanes/pentane.pdb  
./shell-lesson-data/exercise-data/alkanes/octane.pdb  
./shell-lesson-data/exercise-data/alkanes/ethane.pdb  
./shell-lesson-data/exercise-data/alkanes/cubane.pdb
```

Caracteres comodines

```
cubane.pdb  
ethane.pdb  
methane.pdb  
octane.pdb  
pentane.pdb  
propane.pdb
```

Caracter * Permite buscar coincidencias de palabras completas ó partes de palabras

.pdb todos los archivos

p.pdb pentane.pdb y propane.pdb

?ethane.pdb coincide con methane.pdb

???ane.pdb coincide con cualquier palabra con 3 letras finalizadas por ane.pdb, por ejemplo

cubane.pdb ethane.pdb octane.pdb.

Mirando dentro de archivos

Entrar al directorio molecules

```
cd molecules
```

Muestra el contenido del archivo

```
cat cubane.pdb
```

Encabezado de archivo

```
head cubane.pdb
```

Lineas finales del archivo

```
tail -n 2 cubane.pdb
```

Muestra el contenido del archivo

```
less cubane.pdb
```

Cuenta las líneas de todos los archivos extensión pdb

```
wc *.pdb
```

Encontrando patrones:

```
grep "ATOM" cubane.pdb
```

EJERCICIO

1. Del directorio **dircurso/coronavirus/variants** concatenar todos los archivos en uno solo.
2. Crear un nuevo archivo llamado alpha.csv, que debe contener solo el patron Alpha
3. Contar cuantas ocurrencias existen de esa variante
4. Repita el mismo procedimiento para contar la variante Delta del coronavirus en todo esos paises
5. Ejecute los siguientes comandos y explique el resultado:

```
cat *.csv | cut -d "," -f 2 | sort  
cat *_variants.csv | cut -d "," -f 2 | sort | uniq
```

JULIA EVANS
@bork

sort & uniq

16

sort sorts
its input

```
$ sort names.txt
```

the default sort is
alphabetical.

sort -n numeric sort	
'sort' order	'sort -n' order
12	12
15000	48
48	96
6020	6020
96	15000

sort -h: human sort	
'sort -n' order	'sort -h' order
15 G	45 K
11 M	30 M
11 K	15 G
200 G	200 G

useful example:
du -sh * | sort -h

uniq removes duplicates

a a notice there
b b are still 2
b => b 'a's! uniq
a a only uniquifies
c c adjacent
c matching lines

sort + uniq = ❤
Pipe something to
'sort | uniq' and you'll
get a deduplicated list
of lines! **sort -u** does the
same thing.
b
b
b
a
or sort -u => a

uniq -c
counts each line it saw.
Recipe: get the top 10 most
common lines in a file:

```
$ sort foo.txt  
| uniq -c  
| sort -n  
| tail -n 10
```



Parte de la presentación fue tomada de:

<https://cambiotraining.github.io/unix-shell>