



Interacción con la shell

Shell vs Terminal vs Consola

Shell

Es el programa que realmente procesa comandos y genera una salida.

Ejemplos:

Bourne Shell (sh)
Korn Shell (ksh)
C Shell (csh)
Bash

Terminal

Se refiere a un programa (wrapper) que ejecuta una shell.

Ejemplos:

Terminal
Terminator
XMate

Consola

Usualmente significa el teclado y ratón físicamente conectados a la computadora.

Ejemplo

“Accedí al servidor a través de consola porque la red no estaba funcionando”



man

El mejor amigo del usuario linux



man
\$command

Despliega el manual de un
comando

A large, vibrant green brushstroke graphic that sweeps across the left side of the slide, partially enclosed by a white rectangular frame.

man
\$configfile

Despliega el manual de un archivo de configuración



**man
\$daemon**

Despliega el manual de un
proceso “demonio”



man -k

Muestra la lista de los man pages
que contienen una cadena



man sections

Sección	Descripción
1	Programas o comandos
2	Llamadas al sistema
3	Llamadas a bibliotecas
4	Archivos especiales
5	Formatos y convenciones
6	Juegos
7	Misceláneos
8	Rutinas del kernel



Man \$section \$file

Ejemplo:

man passwd

man 1 passwd

man 5 passwd



“

*Unfortunately, manual pages
do not have the answer to
everything...*



Los 10 comandos básicos



ls

- l** forma de lista
- a** todos los archivos
- h** formato legible para humanos



cd

`cd ~`

Accede al directorio home del usuario

`cd ..`

Accede al directorio padre

`cd -`

Accede al directorio anterior



mkdir

-p Crea los directorios padre que sean necesarios



cp

-r recursivo

-i interactivo

Múltiples archivos a un directorio



mv

-i interactivo



rm

-i interactivo

-r recursivo

-f forzar, prácticamente te deja borrar todo



echo

- n** No imprimir el carácter de salto de línea
- e** Reconoce caracteres especiales



pwd

-P se resuelven los enlaces
simbólicos



file

-s para archivos especiales

Utiliza el archivo `/usr/share/file/magic` que contiene patrones para reconocer los tipos de archivos



touch

-t cambia tiempos de acceso y modificación

[[CC]YY]MMDDhhmm[.SS]

-r copia los tiempos de acceso y modificación de un archivo a otro

-a cambia tiempos de acceso

-m cambia tiempos de modificación

[-][[hh]mm]SS

Ejercicios

Cambia a tu directorio home presionando solo 3 teclas

Cambia al directorio /boot/grup presionando solo 11 teclas

Desde tu directorio actual lista todos los archivos de tu directorio home en formato humano presionando solo 9 teclas

Crea un archivo vacío y luego cámbiale el nombre

Ejercicios

Crear un directorio llamado mexico en el home del usuario

Crear los archivos 'dia_bandera.txt' y 'dia_constitucion.txt'

Cambiar las fechas de estos archivos para que coincidan con las fechas del día de la bandera y el día de la constitución respectivamente.

Crear un respaldo del directorio mexico



1. Lectura de archivos



head

-n las primeras 'n' líneas de un archivo

Sin opciones despliega las primeras 10 líneas de un archivo



tail

-n las últimas 'n' líneas de un archivo

Sin opciones despliega las últimas 10 líneas de un archivo



cat

Copia entrada estandar a salida estandar. Es corto de concatenar.

También sirve para crear o copiar archivos.

Ej.

```
cat > texto.txt <<alto
```

```
cat /passwd > passwd.backup
```



tac

Concatenar de atrás para adelante



more

Útil para mostrar un archivo que no caben en una sola pantalla

ESPACE muestra la siguiente pantalla

ENTER muestra la siguiente línea



less

Similar a more, pero permite moverse hacia atrás

ESPACE muestra la siguiente pantalla

ENTER muestra la siguiente línea
b regresa una pantalla
y regresa una línea



strings

Muestra cadenas legibles en
archivos binarios

Ejercicios

Muestra la última línea del archivo `/etc/passwd`

Usa el comando `cat` para crear un archivo llamado 'basics' que contenga una lista de los nombres de los 10 comandos básicos

Crear un respaldo de este archivo usando el comando `cat`

Ejercicios

Abrir una segunda terminal y en ambas ir a su directorio home.

En la terminal 1

echo "Esta es la primera línea del archivo" > prueba.txt

En la terminal 2

tail -f prueba.txt

En la terminal 1

echo "Esta es la segunda línea" >> prueba.txt

¿Que hace tail -f?



¿Qué pasa cuando se escribe un comando y se presiona enter?



La shell escanea la línea que acabamos de ingresar y la separa en argumentos, en ocasiones hace algunos cambios y luego se ejecuta



2. Shell expansion



**Remueve
espacios
en blanco**

echo hello world

echo hello world



Comillas simples

echo 'hello world'

echo 'hello world'



Comillas dobles

echo “hello world”

echo “hello world”

Comandos externos o integrados

Externos

Son programas que tienen sus propios binarios y que se alojan en alguna parte del sistema de archivos

/bin

/sbin

¿Cómo identificarlos?

type cd

type cat

Integrados

Son una parte integral del programa de la shell en sí.



Prioridad

Un comando puede ser de ambos tipos, pero la versión integrada siempre tiene prioridad.

Para ejecutar la versión externa es necesario escribir el PATH del comando.

type -a echo



alias

Para los fans de windows

alias dir=ls

Para los descuidados

alias rm='rm -i'

Para los flojos

alias c=clear

* Sin argumentos muestra una lista de todos los alias

A large, vibrant green brushstroke graphic that sweeps across the left side of the slide, partially enclosed by a white rectangular frame.

unalias

Permite borrar un alias existente

unalias dir

unalias rm

unalias c



Espiar la shell

set -x muestra el proceso de shell expansion

set +x deja de mostrar el proceso de shell expansion

Ejercicios

1. ¿Es tac un comando integrado?

Crear un alias llamado 'ls' equivalente a ls -l

Crear un alias llamado 'ciudad' 'que muestre tu ciudad de nacimiento

Mostrar el proceso de shell expansion

Ejecutar los alias creados

Dejar de mostrar el proceso de shell expansion



3. **Operadores de control**



;

Se ejecutan varios comandos en una sola línea. Siempre uno después del otro

```
mkdir prueba ; cd prueba
```



&

Cuando una línea termina en &. La shell devuelve el prompt y ejecuta el comando en segundo plano.

Ejemplo:

sleep 5

...



\$?

Es un parámetro de la shell y almacena el valor de retorno de un comando



&&

AND lógico.

El segundo comando se ejecuta solo si el primero termina en 0

Ejemplo:

```
mkdir prueba && cd prueba
```

```
cd prueba && ls
```



OR lógico.

El segundo comando se ejecuta solo si el primero termina en algo diferente de 0

Ejemplo:

```
cd prueba || ls
```



#

Todo lo que sigue después del
caracter de número es ignorado



\

Si se antepone a un caracter especial, la shell lo considera un caracter normal

Ejemplo

echo hola \; mundo

Si se coloca al final de una línea, los comandos continuan en la siguiente línea

Ejercicios

Ejecutar `pwd` dos veces en una sola línea

Ejecutar `'ls'` solo si el comando `'cd /etc'` termina en 0

Ejecutar `cd` en caso de que el comando `cat ./bashrc` falle

Ejecutar un `'sleep 200'` (no esperen a que termine)

Combina los operadores `&&` y `||` para construir una estructura if-then-else en línea de comandos

EXTRA

Usando el comando echo mostrar exactamente la siguiente cadena (incluyendo todas las comillas)

"Hola mundo con caracteres' raros * [] ~ \\. "



4. Variables de la shell



\$

Es otro caracter especial de la shell
y permite leer el valor de una
variable

Las variables en la shell son
case-sensitive

A large, vibrant green brushstroke graphic that sweeps across the left side of the slide, partially framing the title.

¿Cómo crear variables?

Para crear/asignar valor a una variable no es necesario el signo '\$'.

Ejemplo:

```
nombre='JuanEscutia'
```



Comillas

echo \$nombre

echo “\$nombre”

echo ‘\$nombre’



set

Se puede utilizar para mostrar una lista de todas las variables de entorno

set | more



unset

Puede remover una variable del entorno de la shell

```
unset nombre_variable
```



env

Sin opciones muestra una lista de las variables exportadas.

No confundir con el comando set
set lista todas las variables
incluyendo las no que no están
exportadas



env -i

También se utiliza para iniciar una shell sin variables heredadas.

```
[miguel@mapq ~] $ bash -c 'echo $SHELL $HOME $USER'
/bin/bash /home/miguel miguel
[miguel@mapq ~] $ env -i bash -c echo 'echo $SHELL $HOME $USER'
```



env **\$VARIABLE**

Se utiliza para establecer variables de entorno para un solo comando

```
[miguel@mapq ~]$ env LANG=en_US.UTF-8 bash -c 'echo $LANG'  
en_US.UTF-8  
[miguel@mapq ~]$ echo $LANG  
es_MX.UTF-8
```




export

Se utiliza para exportar variables a otras shells.

Solo exporta a las shells hijas



Delinear variables

Ejemplo

```
variable=345
```

```
echo "012${variable}6789"
```



Variables no definidas

Por defecto una shell despliega nada, si una variable no existe

`set -u`

La opción de la shell `nounset` genera un error cuando la variable no existe

Ejercicios

Usa el comando echo para mostrar un 'Hola' seguido de tu nombre de usuario

Copia el valor de la variable \$LANG en \$MyLANG

Lista todas las variables de la shell

Lista todas las variables exportadas de la shell

¿Se muestra la variable \$MyLANG?



5. Variables de entorno



\$HOME

Contiene el directorio home del usuario loggeado

A large, vibrant green brushstroke graphic that sweeps across the left side of the slide, partially enclosed by a white rectangular frame.

\$HOSTNAME

Contiene el nombre del host



\$LANG

Contiene el lenguaje



\$PATH

Aquí se guardan la dirección
donde la shell busca los comandos
que se van a ejecutar



\$PS1

Esta variable determina el valor del prompt



\$SHELL

Contiene el directorio de la shell
que se está utilizando

A large, vibrant green brushstroke graphic that sweeps across the left side of the slide, partially enclosed by a white rectangular frame.

\$UID

Contiene el User_ID del usuario
loggeado



\$USER

Contiene el nombre del usuario
logueado

Tarea

Cambia tu prompt para que se vea de la siguiente manera

[<Nombre_host>:<Directorio_Actual> <Hora_actual>](#/\$)

La hora debe estar en formato de 24 horas HH:MM

Usar color verde para (#/\$)

Buscar en el manual de bash la sección de PROMPTING



6. Shell implantada



Las shells pueden ser incrustadas en línea de comandos.



Sub shell

```
echo $var1
```

```
echo $(var1=1; echo $var1)
```

```
echo $var1
```

Nested Embedding

Se puede incrustar una shell en una shell incrustada y en repetidas ocasiones.

```
[miguel@mapq home]$ A=shell  
[miguel@mapq home]$ echo $C$B$A $(B=sub; echo $C$B$A; echo $(C=sub; echo $C$B$A))  
shell subshell subsubshell
```



Comillas invertidas `

Se pueden utilizar para crear una sub shell

```
echo 'cd $HOME; ls -l'
```

```
echo $(cd $HOME; ls -l)
```

No funciona múltiples veces

Opciones de la shell

-x

-u

-c

Para listar las opciones activadas de la shell se puede ejecutar el siguiente comando

echo \$-

Para ver todas las opciones buscar en el manual de bash la cadena 'option-name'



7. Historial



!!

(bang bang)

Repite el último comando



!
(bang)

La shell repetirá el último comando que comenzó con los caracteres después del bang
`!<indicio_de_un_comando>`



history [n]

Permite mirar el historial completo de comandos.

n: permite ver los últimos n comandos



!<n>

Imprime el comando 'n' del historial y luego lo ejecuta



Ctrl-r

Permite buscar un comando por coincidencias con los caracteres ingresados



\$HISTSIZE

Determina el número de comandos que serán recordados en el historial



\$HISTFILE

Contiene la ubicación del archivo donde se almacena el historial.

Escribir exit para almacenar el historial

Cerrar la terminal con el mouse o dar reboot no guarda el historial



\$HISTFILES IZE

Se puede utilizar para definir el número de comandos a recordar en el historial



8. File Globbing



*

Es un comodín que puede ser
sustituido por cualquier
combinación o repetición de
caracteres

ls *.txt



?

Es otro comodín que puede ser
sustituido por cualquier caracter,
pero solo uno.

ls practica?.txt



Es otro comodín que puede ser sustituido por cualquiera de los caracteres especificados entre los corchetes.

Cada par de corchetes es sustituido por un solo caracter



Rangos

Se pueden abreviar los rangos de caracteres utilizando la siguiente sintaxis

[a-z]

[A-Z]

[0-9]



!
(negación)

Los caracteres después del bang
se excluyen de las coincidencias

`ls *[^.]`

Ejercicios

Crea un directorio de prueba y acceder a este

Crear los siguientes archivos

practica1.sh

practica_2.txt

practica 3.pdf

tarea1.txt

tarea02.png

Ejercicios

Lista todos las prácticas

Lista todos los archivos que tenga extensión '.txt'

Lista la tarea1 y la practica1

Lista los archivos que no tengan espacios

Lista los archivos que no tengan guiones

Lista los archivos que no tengan extensión .sh



Now you can use any emoji as an icon!

And of course it resizes without losing quality and you can change the color.

How? Follow Google instructions

<https://twitter.com/googledocs/status/730087240156643328>

