Terminal y Línea de Comandos

¿Qué es la Terminal?

Interfaz gráfica que nos permite ejecutar línea de comandos (Shell)

¿Qué es un comando?

Es un programa que se puede ejecutar desde la terminal.

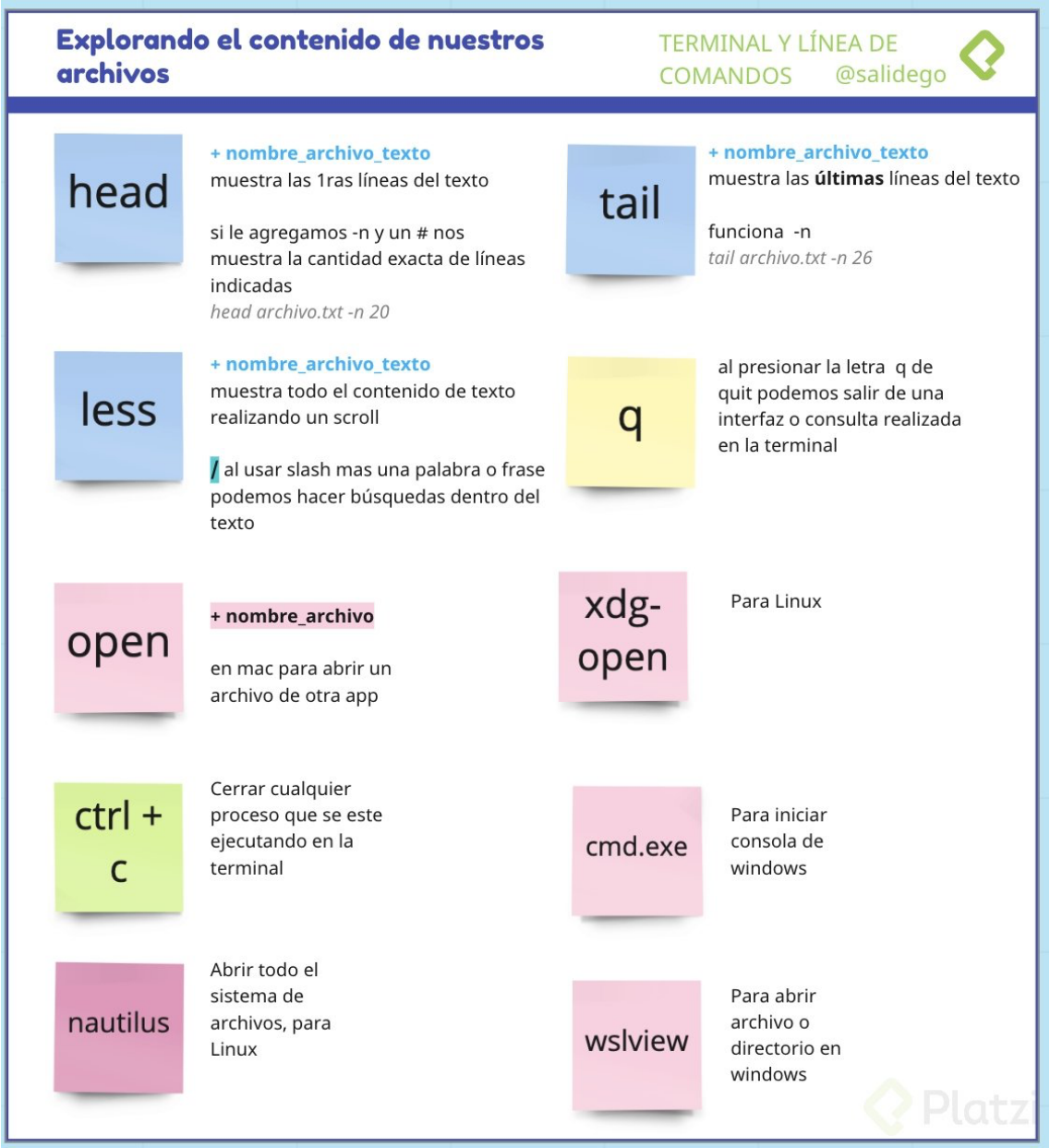
Comandos básicos 1

Ls : muestra el directorio home
Cd: entra a una carpeta
Cmd + l = limpia la terminal es igual al comando Clear
Ls -l: muestra información sobre los archivos listados dentro de una carpeta
Ls -lh : lectura humana, especifica los tamaños de los archivos en mb, b, kb, etc.
cd.. : regresa un directorio atrás
ruta absoluta: ruta completa /docs/etc/
ruta relativa: cd. Cd..
file: ver información de un archivo
pwd: identificar la ruta donde estamos

Comandos básicos 2

ls: Algunas variantes:
ls -la: Nos muestra todo, incluso los ocultos.
ls -lS: Los ordena por tamaño.
ls -lr: Los muestra en reversa.
tree: Nos muestra todo, lo que hay dentro de cada carpeta, en forma de arbol 🎄.
Podemos elegir el nivel de profundidad con tree -L <numero de niveles>
mkdir : crea directorios, se pueden crear varios en un 1 solo comando
touch: crea archivos, se pueden crear varios en un solo comando
cp: copia archivos cp file 1 file_nuevo
mv: mueve los archivos a otro directorio mv file.. (lo mueve hacia atrás)
mv file file1 : Renombra archivos
rm: comando peligroso borra archivos
rm -r: borra un directorio
rm -i : borra de forma interactiva preguntando si estas seguro de borras
se reponde con y de yes

Explorando archivos



¿Qué es un comando?

¿Qué puede ser un comando?
Un programa ejecutable.
Un comando de utilidad de la Shell.
Una función de la Shell.
Un alias.

¿En dónde se guardan los programas ejecutables en Linux?
En la ruta /usr/bin

¿A que nos referimos cuando hablamos de un comando de utilidad de la Shell?
Nos referimos a los comandos que ya vienen por defecto en nuestra Shell.

¿A que nos referimos cuando hablamos de una función de la Shell?
Nos referimos a los comandos que no vienen por defecto en nuestra Shell.

¿Qué es un alias?
Es un comando que nos sirve para almacenar otro comando y poder ejecutarlo solamente con el alias que le asignemos.

¿Para qué nos sirve el comando type comando?
Para saber qué tipo de comando es el que escribimos posterior al comando type:
type comando-a-indagar

¿Qué tipo de comando es el comando cd?
Comando de utilidad de la Shell.

¿Qué tipo de comando es el comando mkdir?
Un programa ejecutable.

¿Qué tipo de comando es el comando ls?
Un alias.

¿el comando ls es alias de qué comando?
Del comando ls --color=auto

¿podemos escribir funciones de comandos con palabras completas?
Si.

¿Los comandos con palabras completas pueden llevar parámetros?
Si.

¿Cómo podemos escribir un comando con una opción de palabras completas?
Escribiendo al principio de la opción dos guiones (--).

¿Cómo podemos crear alias?
Con el comando alias. Seguido del comando escribimos el alias que vamos a usar, seguido escribimos un símbolo de igual (=) y entre comillas escribimos el comando al que le queremos asignar el alias:
alias nombre-alias=comando-al-que-apuntamos

¿Los alias son permanentes?
No.

¿Para qué nos sirve el comando help?
Este comando nos permite saber cómo funciona otro comando. Para usar este debemos escribir seguido del comando help, el comando que queremos indagar:
help comando-a-indagar

¿Como podemos saber cómo funciona un comando sin usar el comando help?
Escribiéndole al comando que queremos investigar la opción de palabra --help.

¿El comando help puede funcionar en todas las Shell?
No.

¿Para qué nos sirve el comando man?
Para desplegar el manual de usuario de algún comando:
man comando-a-indagar

¿Todos los comandos tienen un manual de usuario?
No.

¿Para qué nos sirve el comando info?
Para desplegar un manual de usuario de algún comando, pero de manera más resumida y con una interfaz un poco diferente a la del comando man.

¿El comando info funciona con todos los comandos?
No.

¿Para qué nos sirve el comando whatis?
Para saber de manera super resumida que hace un comando.

WILDCARDS

Caracteres especiales que nos permite hacer búsquedas mas avanzadas.

Ls *.txt : filtra y muestra solo los archivos txt
Ls datos*: muestra todos los archivos que digan datos y tengas varios caracteres después.
Ls datos?: muestra todos los archivos que digan datos y tengan 1 solo carácter al final.
Ls [[:upper:]]*: muestra todos los archivos que inicien con mayúsculas
ls -d [[:upper:]]*: busca solo directorios y archivos con mayúsculas iniciales

[:alnum:] Coincide con cualquier carácter alfanumérico

[:alpha:] Coincide con cualquier carácter alfabético

[:digit:] Coincide con cualquier número

[:lower:] Coincide con cualquier letra minúscula

[:upper:] Coincide con cualquier letra mayúscula

Redirecciones: cómo funciona la Shell

7. Redirecciones: cómo funciona la shell
El stdin es 0, el stdout es 1, y el stderror 2

Les comparto los comandos aprendidos en esta clase:

ls dirTest1 > misDirTest1.txt : En vez de listar en pantalla, lo que hace es mandar el resultado en texto al archivo "misDirTest1.txt", y si no existe ese archivo lo crea. Si vuelvo a usar el mismo archivo, siempre crea nuevamente el archivo y lo sobreescribe

ls downloads >> misDirTest1.txt : En vez de listar en pantalla, lo que hace es mandar el resultado en texto al archivo "misDirTest1.txt", agregando el output al final del archivo, si este ya tiene contenido, es decir concatenando el texto.

ls f2q3fdfsd 2> error.txt : Además de mostar el error en pantalla, este comando envía el texto del error a un archivo, si el archivo no existe lo crea. Si vuelvo a usar el mismo archivo, siempre crea nuevamente el archivo y lo sobreescribe

Redirecciones: pipe operator

Es uno de los operadores mas útiles que existen, ya que nos permite poner varios comandos, tales que la salida de uno es la entrada del siguiente 🔼.
echo <texto> genera un stdout con el texto que tenemos.
cat <archivo1> <archivo2> muestra los dos archivos concatenados 💩.
El pipe operator ┃ hace que el stdout de un comando sea el stdin de otro comando.
Por ejemplo ls -lh | less
tee hace algo parecido a >, pero dentro de los pipe´s, por ejemplo ls -lh | tee output.txt |less . Se puede poner en medio, pero se ignora porque se sigue pasando.
cowsay "Texto" es un comando que imprime una vaca que dice algo JAJAJAJAJ 🐮.

Encadenando comandos: operadores de control

Los operadores de control son símbolos reservados por la terminal que nos permiten encadenar comandos.

Si usas constantemente la tecla enter para ejecutar varios comandos, puedes evitarlo si usas el operador ; que separa los comandos que estamos ejecutando.

Operador
; (Nos permite ejecutar varios comandos en 1 misma linea) ls; cal; mkdir hola.txt
& (ejecuta en paralelo los procesos de varios comandos en linea) y crea los hilos de ejecución.
&& : Se ejecutan solo si el comando anterior se haya ejecutado exitosamente. Suponemos que A, B y C son comando: A && B && C El B solo se va ejecutar si el A se ejecuta exitosamente, y el C solo se va ejecutar si el B si ejecuta exitosamente. Si el B no se ejecuta exitosamenta el C no se ejecuta. Si el A no se ejecuta exitomante el B y el C no se ejecutan. –
|| or: : Solo se ejecuta uno. Sea cuantos comandos tienes separados por || solo ejecuta o toma en cuenta el primer que se ejecuta exitosamente (bajo la redundacia),

y descarta automaticamente los demas. Cuando uno de los comandos se ejecuta exitosamente, descarta los demas comandos.

Cómo se manejan los permisos

Notas :smile: Cómo se manejan los permisos.

Cuando listamos con ls -l se muestran varias cosas. Los tipos de archivos:
- archivo normal.
d directorio.
l link simbólico.
b archivo de bloque especial.
Tipos de modos: rwx corresponde con read, write y execute. Se representan con 3 bits, y los podemos manejar a través de un modo octal, esto es, pasar de binario a número.
rwx (1,1,1) dueño. En modo octal es 7.
r-x (1,1,1) grupo. En modo octal es 5.
r-x (1,0,1) world. Octal 5.
Modo simbólico: Esto es para asignar los permisos a los diferentes posibles usuarios.
u Solo para el usuario.
g Solo para el grupo.
o Solo para otros (world).
a Aplica para todos.

Modificando permisos en la terminal

Quitar o dar permisos
Chmod u-r nombredearchivo.txt (quita permiso)
Chmod u+r nombredearchivo.txt (da permiso)
Ls -l (verifica permisos)
cat nombredearchivo.txt (para visualizar el texto si tenemos permiso o ver acceso denegado si no tenemos)

Cómo configurar variables de entorno

Variables de entorno: Las variables de entorno son útiles cuando necesitamos que cierta información prevalezca para poder trabajar más rápido o necesitamos guardar información para no tener que recordarla constantemente.

Echo $PATH
Echo $HOME

Comandos de búsqueda

Which (busca la ruta de los binarios dentro del sistema)
Find: nos permite encontrar un archivo
find ./ -name *.png (busca todos los archivos con extension .png) aquí se aplica un wildcard *.
find ./ -type f -name "f*"

Comando Grep

Aca subo los apuntes hechos en formato pregunta respuesta para que sea mas sencillo el repaso, si usan anki es bastante mas util

que hace grep

nos permite encontrar encontrar coincidencias en un archivo
que hace el parámetro -i en grep

ignorá si es mayúscula o minúscula
que hace el parámetro -v

me trae las coincidencias que no tengan el parámetro dado
que hace wc

es el comando que nos permite contar la cantidad de palabras, lineas y bits
como son los parámetros de wc

-l : el numero de líneas
-w el numero de palabras
-c el numero de bits

grep [ExpresiónRegular] [archivoDondeBuscar]

Utilidades de Red

Ifconfig (comando que muestra toda información relacionada a nuestra red)
Ping www.google.com (descubrir si una conexión de red funciona correctamente)
Curl (trae un archivo de manera de texto atravez de la red)
Wget (trae desde internet)
nos sirve para ver por cuales computadoras tenemos que ir pasando para llegar por ejemplo a una pagina web. Ejemplo, nos saldrán las ip que tenemos que pasar para llegar a la pagina que queremos.
$ netstat –i // nos muestra los dispositivos de red.

Comprimiendo archivos tar y zip

tar -cvf toCompress.tar toCompress: Con este comando lo que hacemos es comprimir un archivo, primero debemos decir el nombre del archivo final y luego la carpeta u archivo que deseamos comprimir.
tar -cvzf toCompress.tar.gz toCompress: Comprime de la misma manera, pero el output va a ser un archivo con extesion gz.
tar -xzvf toCompress.tar.gz: Con este comando descomprimimos un archivo comprimido en gzip.
zip -r toCompressInZip.zip toCompress: Con este comando lo que hacemos es comprimir un archivo en zip.
unzip toCompressInZip.zip: Esto es para descomprimir un zip en la terminal.

Manejo de procesos

ps ax, para ver los procesos del sistema.
top muestra los proceso en tiempo real
kill -9 pid, para terminar el proceso inmediatamente.
killall -9 proceso, para terminar el proceso.

Procesos de foreground y background.

Los procesos que están corriendo pero no se muestran en terminal se dice que están en background. Los que si se muestran están en foregroung. 🏠
Para mover un proceso al background, usamos Ctrl+z. Esto lo suspende, pero sigue corriendo (como con Cat). Para matar un proceso se usa Ctrl+c
fg <numero de trabajo> nos permite traer un proceso al foreground. Es importante notar que el número de trabajo no es lo mismo que el PID.
bg <numero de trabajo> nos permite llevar un proceso al background, pero sin suspender el proceso.⭐

Editores de texto en la terminal

Vim

Personalizar la terminal de comandos