Interfaz de LÍNEA DE COMANDOS (CLI)

Segundo Parcial - Introducción a la Informática

GRUPO 6

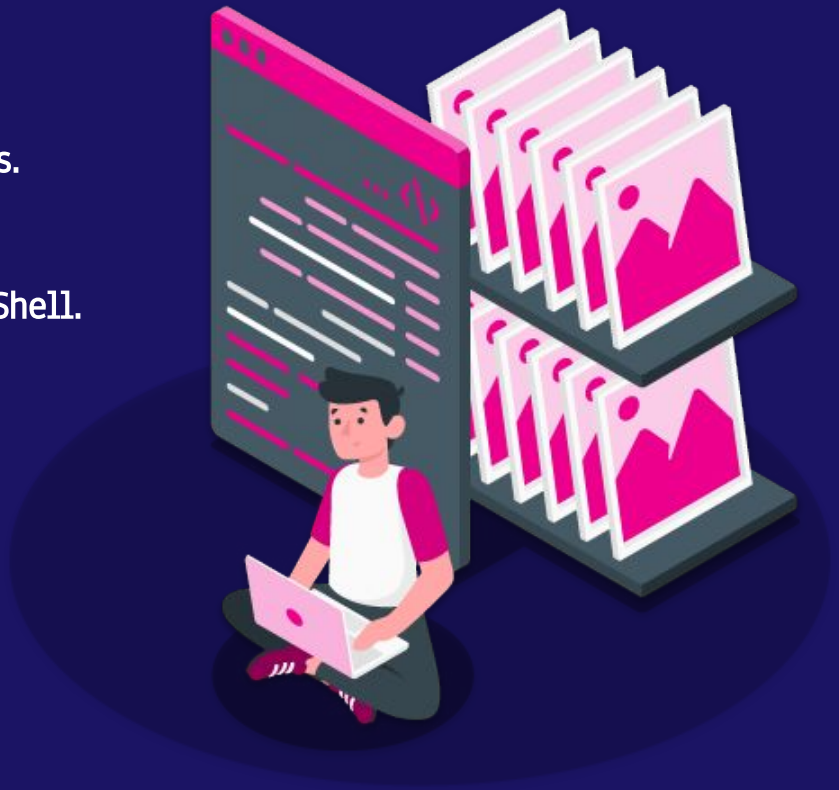
Docente:

- Erika Tovar

Equipo:

- Sorany Villa
- Luis Fernando Rodriguez
- Johan Galvis
- Luis Torrellas
- Marcos Escobar

ÍNDICE DE CONTENIDOS

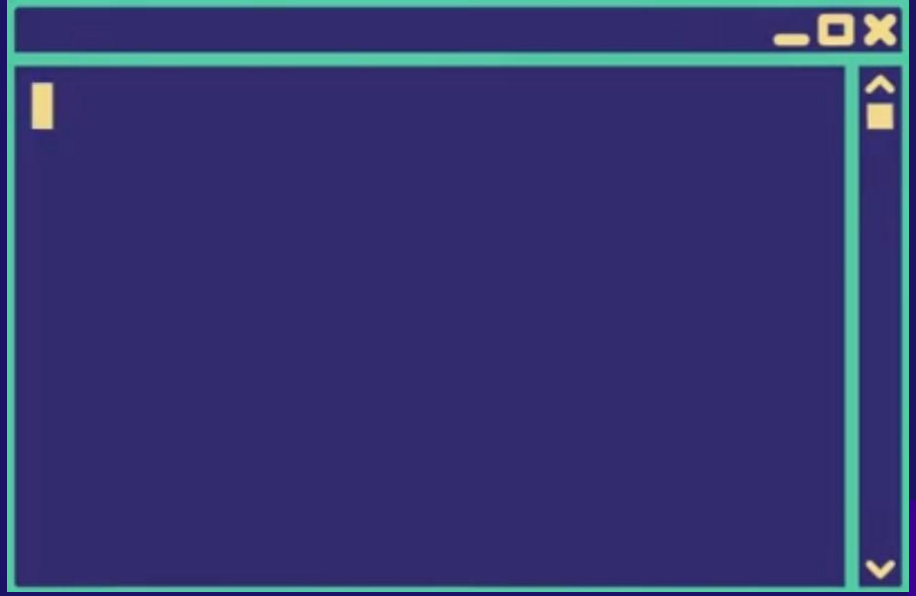


¿Qué es la terminal y para qué sirve?

Sorany Villa

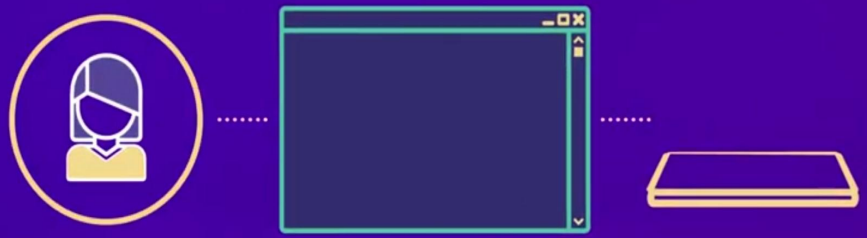
¡Un poco de historia !

Antes de los años 70 no existían interfaz graficas de usuario



Que es la terminal

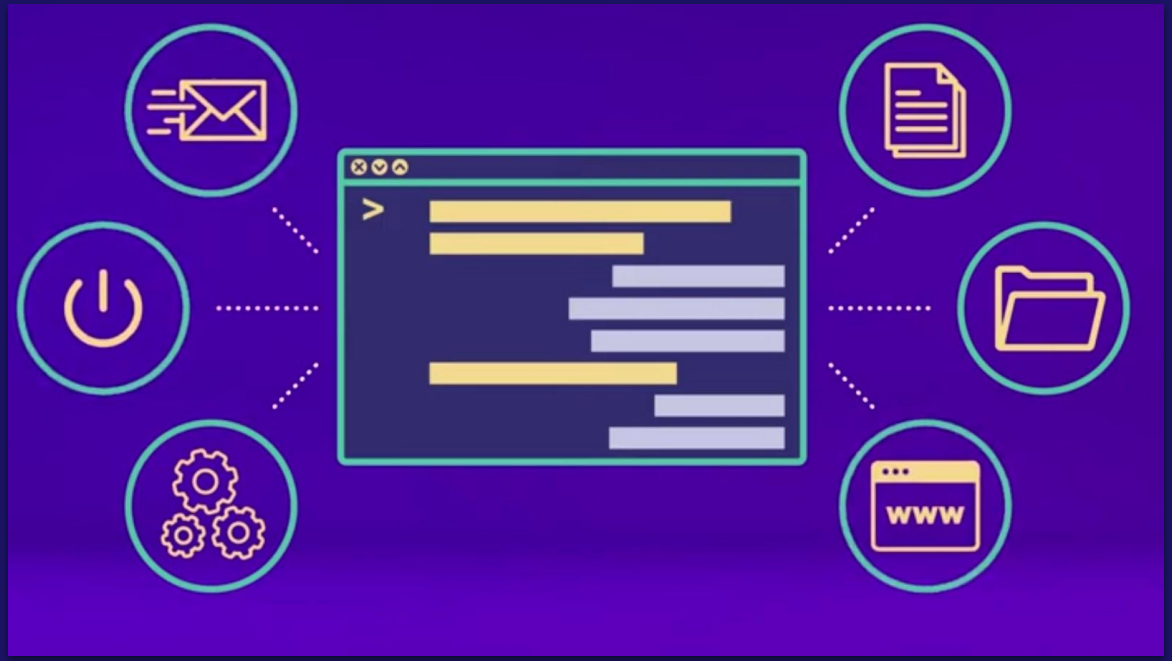
Función básica



Abrir la terminal

CLI

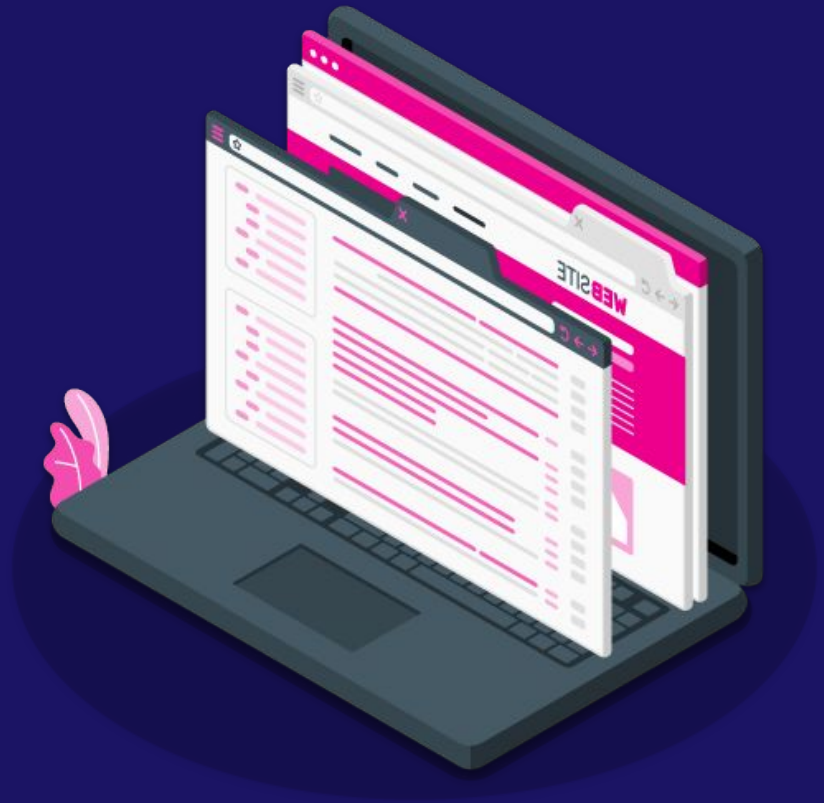
Que tipo de órdenes



Diferencias entre

CLI y GUI

Luis Fernando Rodriguez



Command Line Interface (CLI)

CLI

- Teniendo un buen conocimiento puede dar mayores posibilidades y funcionalidades
- Requiere menos periféricos de entrada para navegar sobre el terminal (solo teclado).
- Recuerda los comandos o acciones en la misma sesión.
- Es directo.
- Requiere alto nivel de conocimiento informático y de comandos
- Es ligero, por tal, ejecuta las acciones rápidamente.

Graphical User Interface (GUI)

GUI

- Ofrece facilidades de acceso y visualización de la información.
- Es intuitivo.
- Requiere bajo nivel de conocimientos.
- Consume mayor cantidad de recursos de la computadora.
- Amigable con el usuario.

Ventajas de la terminal en comparación con la interfaz Gráfica de usuario

- Consume pocos recursos de la computadora.
- Devuelve los resultados solicitados o hacer las modificaciones en menor tiempo.
- Dependiendo de los conocimientos, puede tener más funcionalidades.
- Permite estar enfocado en la tarea a desarrollar.
- Su compatibilidad en todas las versiones del sistema operativo permite que sus funcionalidades y uso no cambien.

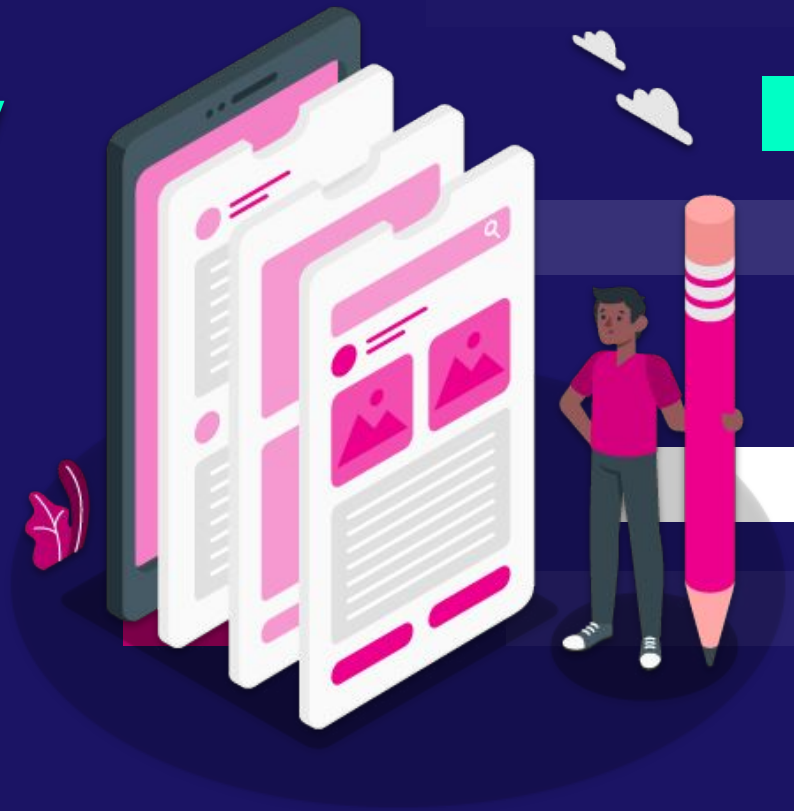
Ventajas de la interfaz Gráfica de usuario en comparación con la terminal

- Es intuitiva.
- Permite visualizar la información con mayor facilidad y ubicarse dentro de los archivos y carpetas.
- No requiere muchos conocimientos para usarlo.

  Facilita hacer diferentes tareas al mismo tiempo.
- Su uso e interacción no varía mucho, sin importar el sistema operativo. Contrario de la terminal, que sus comandos varían según el SO.
- Dan usabilidad a programas que se instalen en el sistema operativo.

CLI, Consola y Shell.

Johan Galvis



CONSOLA

- Hardware
- Puerto
- Conexión Digital
- Conexión a Unidad Central, (Mainframe).

TERMINAL

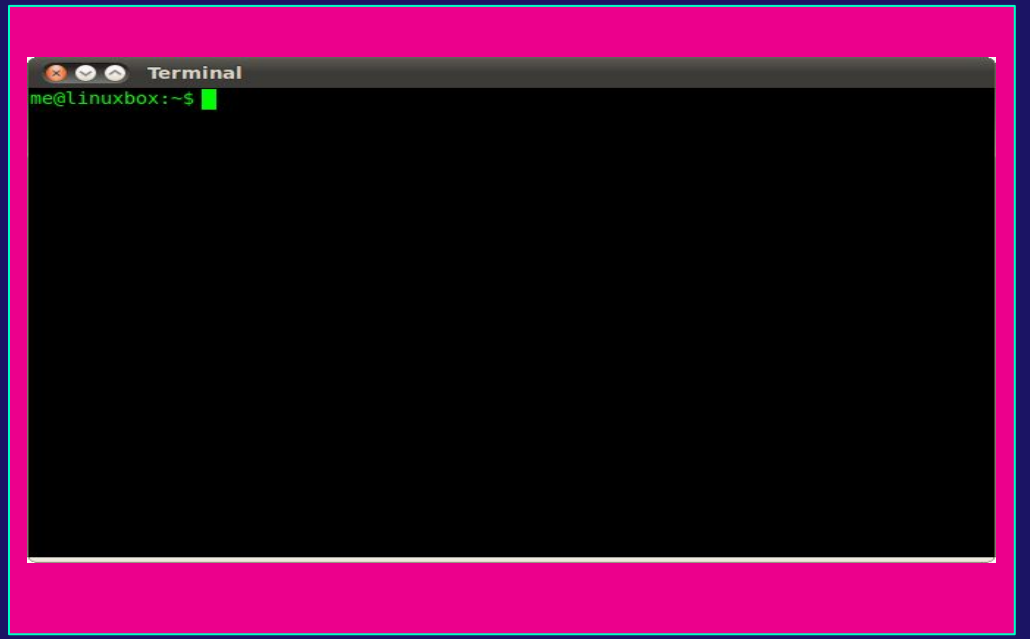
- Entorno de entrada y salida de texto.
- Es un programa contenedor
- Ejecución de SHELL.
- Emulador de terminal.

SHELL

- Intérprete de CLI.
- Comunicación directa con el SO.
- Realiza las solicitudes y permite la ejecución.

CLI (Command line interface)

- Ubicación donde se permite la escritura de comandos.
- Permite la interactividad del usuario.
- Acceso al shell.



Comandos y funciones básicos de la CLI

Luis Torrellas

Comandos y funciones
Mostrar, crear y eliminar

- ls: Lista el contenido de un directorio
- ls -a: Lista todos los archivos, incluyendo los archivos ocultos.
- pwd: Muestra la carpeta en la que se está trabajando.
- mkdir: Crea un nuevo directorio o carpeta.
- touch: crea un nuevo archivo.
- rm: Elimina un archivo.
- rmdir: Elimina una carpeta vacía.
- rm -r: Elimina una carpeta y su contenido.

Comandos y funciones
Mover, renombrar y otros

- mv: Mueve o renombra archivos y directorios.
- cp: Copia un archivo o carpeta.
- cd (nombre de carpeta): nos lleva a la carpeta seleccionada.
- cd .. : Nos lleva a la carpeta padre de la actual.
- clear: limpia la pantalla de la terminal

Comandos y funciones Repositorio local.

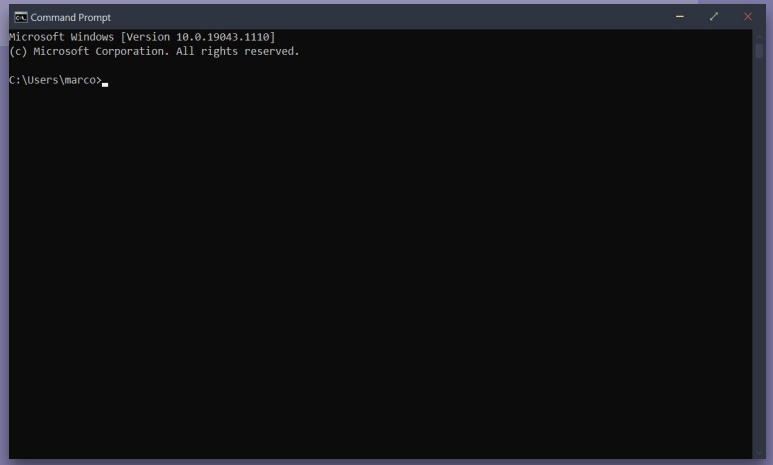
- ➔ git init: Crea un repositorio.
- ➔ git config user.name: Agrega nuestra identidad.
- ➔ git config user.email: Agrega nuestro email.
- ➔ git add . : Agrega todos los archivos al repositorio.
- ➔ git commit -m: Comitea los cambios hechos.
- ➔ git log: Historial de nuestros commits.

CLI más usados y sus principales características.

Marcos Escobar

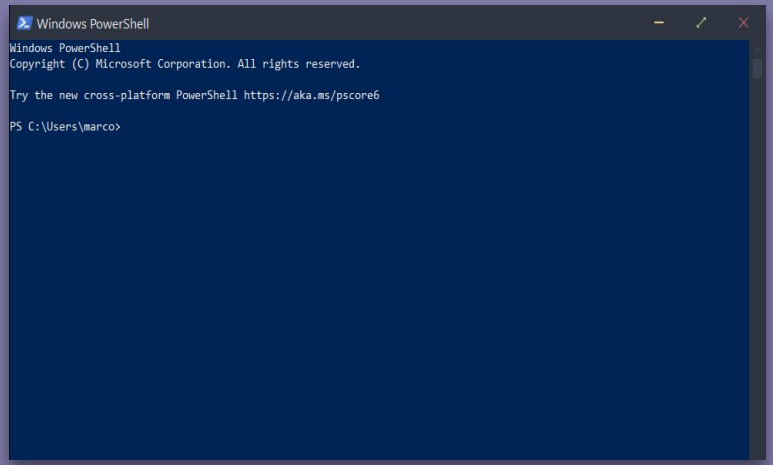
CLI MÁS UTILIZADOS

CLI	Windows	Apple	Linux / Unix / GNU
Ms-Dos	✔️		
Windows Terminal	✔️		✔️
PowerShell	✔️	✔️	✔️
Bash	✔️	✔️	✔️
TMux		✔️	✔️
Zsh		✔️	✔️

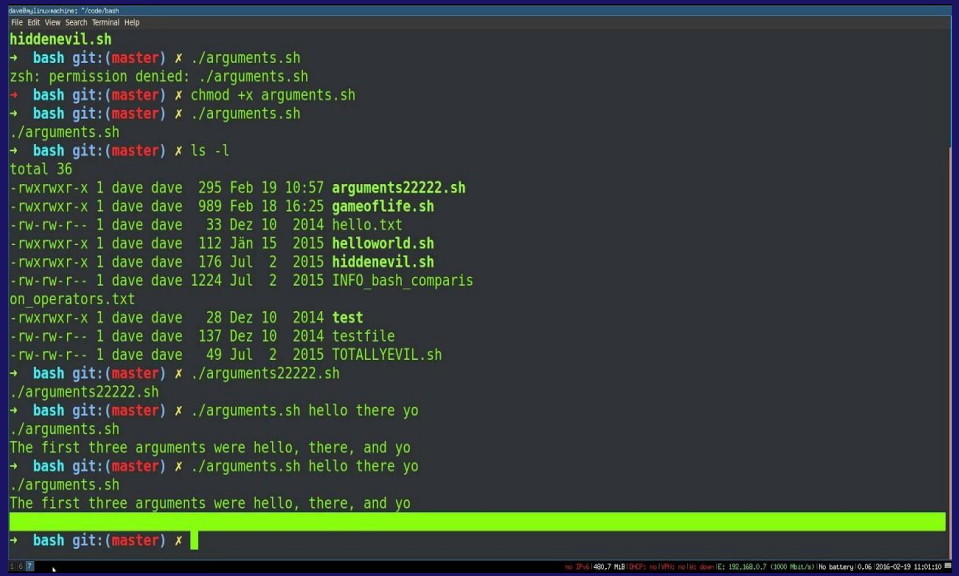


PowerShell

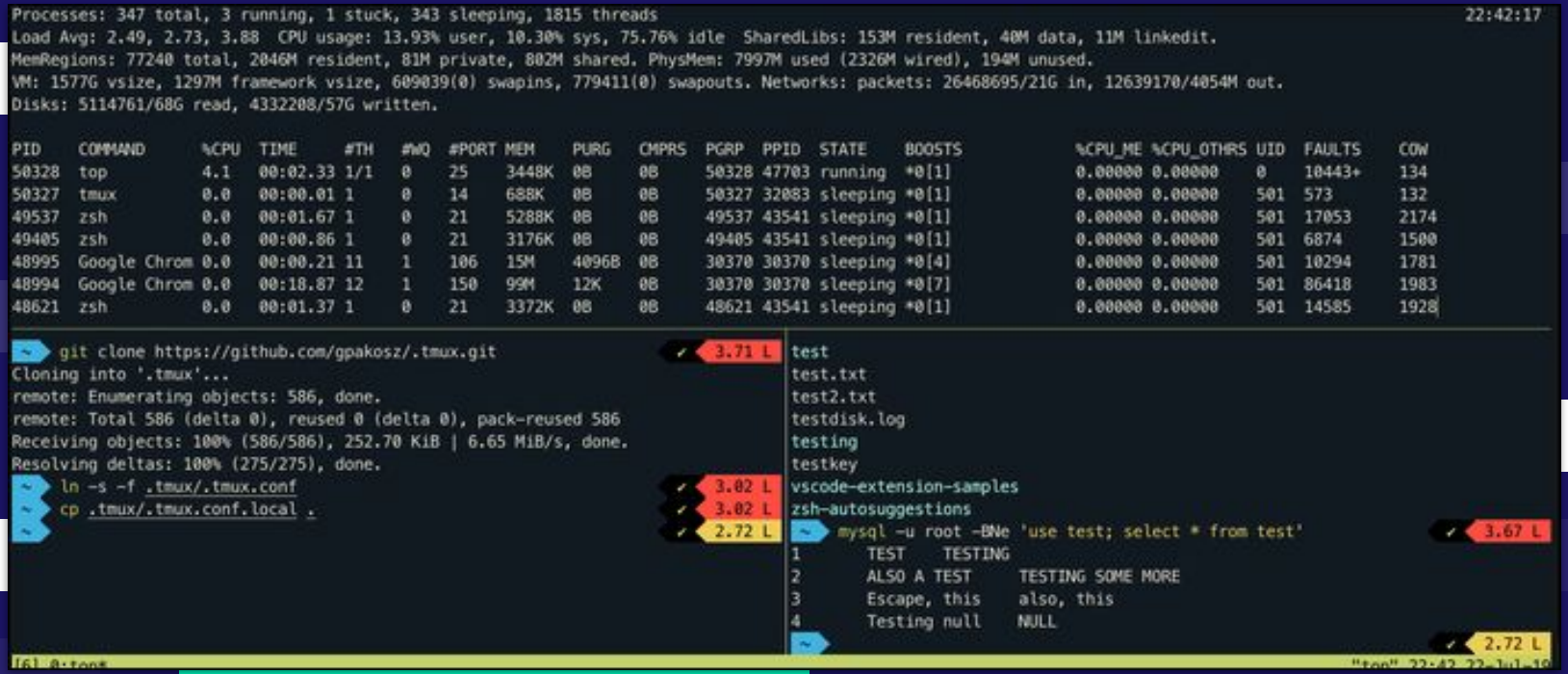
Windows Terminal

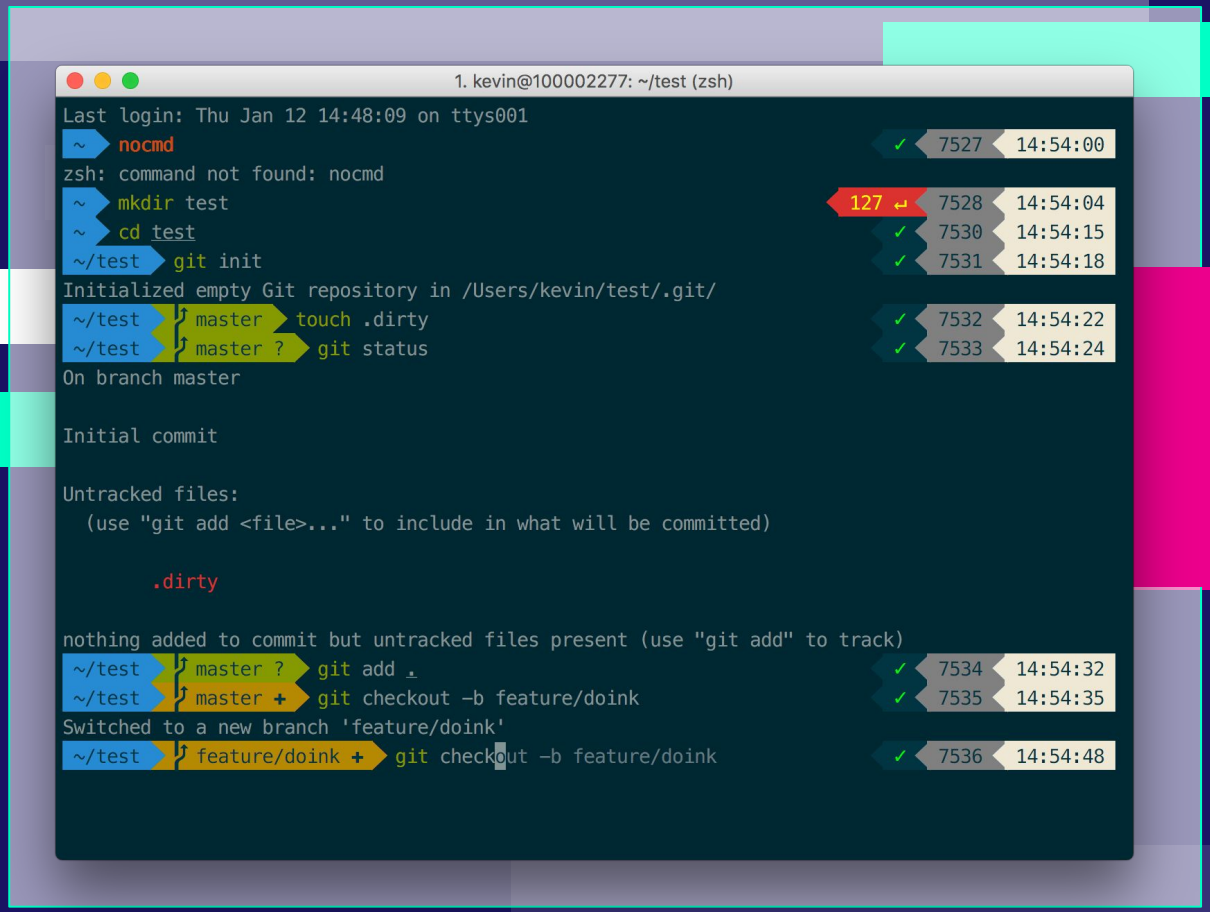


Bash


File Edit View Search Terminal Help
hiddenevil.sh
→ bash git:(master) ✗ ./arguments.sh
zsh: permission denied: ./arguments.sh
→ bash git:(master) ✗ chmod +x arguments.sh
→ bash git:(master) ✗ ./arguments.sh
./arguments.sh
→ bash git:(master) ✗ ls -l
total 36
-rwxrwxr-x 1 dave dave 295 Feb 19 10:57 arguments22222.sh
-rwxrwxr-x 1 dave dave 989 Feb 18 16:25 gameoflife.sh
-rw-rw-r-- 1 dave dave 33 Dez 10 2014 hello.txt
-rwxrwxr-x 1 dave dave 112 Jän 15 2015 helloworld.sh
-rwxrwxr-x 1 dave dave 176 Jul 2 2015 hiddenevil.sh
-rw-rw-r-- 1 dave dave 1224 Jul 2 2015 INFO_bash_comparis
on_operators.txt
-rwxrwxr-x 1 dave dave 28 Dez 10 2014 test
-rw-rw-r-- 1 dave dave 137 Dez 10 2014 testfile
-rw-rw-r-- 1 dave dave 49 Jul 2 2015 TOTALLYEVIL.sh
→ bash git:(master) ✗ ./arguments22222.sh
./arguments22222.sh
→ bash git:(master) ✗ ./arguments.sh hello there yo
./arguments.sh
The first three arguments were hello, there, and yo
→ bash git:(master) ✗ ./arguments.sh hello there yo
./arguments.sh
The first three arguments were hello, there, and yo
→ bash git:(master) ✗

TMux


Processes: 347 total, 3 running, 1 stuck, 343 sleeping, 1815 threads 22:42:17
Load Avg: 2.49, 2.73, 3.88 CPU usage: 13.93% user, 10.30% sys, 75.76% idle SharedLibs: 153M resident, 40M data, 11M linkedit.
MemRegions: 77240 total, 2046M resident, 81M private, 802M shared. PhysMem: 7997M used (2326M wired), 194M unused.
VM: 1577G vsize, 1297M framework vsize, 609039(0) swapins, 779411(0) swapouts. Networks: packets: 26468695/21G in, 12639170/4054M out.
Disks: 5114761/68G read, 4332208/57G written.
PID COMMAND %CPU TIME #TH #WQ #PORT MEM PURG CMPRS PGRP PPID STATE BOOSTS %CPU_ME %CPU_OTHRS UID FAULTS COW
50328 top 4.1 00:02.33 1/1 0 25 3448K 0B 0B 50328 47703 running *0[1] 0.00000 0.00000 0 10443+ 134
50327 tmux 0.0 00:00.01 1 0 14 688K 0B 0B 50327 32083 sleeping *0[1] 0.00000 0.00000 501 573 132
49537 zsh 0.0 00:01.67 1 0 21 5288K 0B 0B 49537 43541 sleeping *0[1] 0.00000 0.00000 501 17053 2174
49405 zsh 0.0 00:00.86 1 0 21 3176K 0B 0B 49405 43541 sleeping *0[1] 0.00000 0.00000 501 6874 1500
48995 Google Chrom 0.0 00:00.21 11 1 106 15M 4096B 0B 30370 30370 sleeping *0[4] 0.00000 0.00000 501 10294 1781
48994 Google Chrom 0.0 00:18.87 12 1 150 99M 12K 0B 30370 30370 sleeping *0[7] 0.00000 0.00000 501 86418 1983
48621 zsh 0.0 00:01.37 1 0 21 3372K 0B 0B 48621 43541 sleeping *0[1] 0.00000 0.00000 501 14585 1928
~ git clone https://github.com/gpakosz/.tmux.git
3.71 L
Cloning into '.tmux'...
remote: Enumerating objects: 586, done.
remote: Total 586 (delta 0), reused 0 (delta 0), pack-reused 586
Receiving objects: 100% (586/586), 252.70 KiB | 6.65 MiB/s, done.
Resolving deltas: 100% (275/275), done.
~ ln -s -f .tmux/.tmux.conf
3.02 L
~ cp .tmux/.tmux.conf.local .
3.02 L
~
2.72 L
test
test.txt
test2.txt
testdisk.log
testing
testkey
vscode-extension-samples
zsh-autosuggestions
~ mysql -u root -BNe 'use test; select * from test'
3.67 L
1 TEST TESTING
2 ALSO A TEST TESTING SOME MORE
3 Escape, this also, this
4 Testing null NULL
~
2.72 L


1. kevin@100002277: ~/test (zsh)
Last login: Thu Jan 12 14:48:09 on ttys001
~ nocmd ✓ 7527 14:54:00
zsh: command not found: nocmd
~ mkdir test 127 ↵ 7528 14:54:04
~ cd test ✓ 7530 14:54:15
~/test git init ✓ 7531 14:54:18
Initialized empty Git repository in /Users/kevin/test/.git/
~/test master touch .dirty ✓ 7532 14:54:22
~/test master ? git status ✓ 7533 14:54:24
On branch master
Initial commit
Untracked files:
(use "git add <file>..." to include in what will be committed)
.dirty
nothing added to commit but untracked files present (use "git add" to track)
~/test master ? git add . ✓ 7534 14:54:32
~/test master + git checkout -b feature/doink ✓ 7535 14:54:35
Switched to a new branch 'feature/doink'
~/test feature/doink + git checkout -b feature/doink ✓ 7536 14:54:48

zsh

¡MUCHAS GRACIAS!

Docente:

- Erika Tovar

Equipo:

- Sorany Villa
- Luis Fernando Rodriguez
- Johan Galvis
- Luis Torrellas
- Marcos Escobar

Ilustraciones:

https://storyset.com/