



School of
DevOps

Operating Systems **Linux**

Mauricio Camayo – Federico Cabrera





Mauricio Camayo

Development Lead



Systems and Computer Engineer

XP: Developer 20+, Security 7+, Management 5+

CEH, LINUX Geek, MMA



Federico Cabrera

DevOps Engineer



Systems and Computer Engineer
XP: Sysadmin, Cybersecurity, DevOps
Operating Systems, Formula 1, Music

Agenda

01

Unix – GNU/Linux

02

Kernel

03

Open source

04

Sudo

05

File tree, links and devices

06

Bash

Linux is NOT an operating system

Linux is a Kernel, the actual OS is GNU/Linux



Richard Stallman



Richard Stallman created a movement in 1983 to create free software inside a unix-like operating system, but it was only until **Linus Torvalds** that created the Kernel in 1991 that the whole system could be run without any other license.



Linus Torvalds

Linux is the kernel that allowed GNU to become a fully functional open operating system, so it should be called GNU/Linux

(but nobody does)

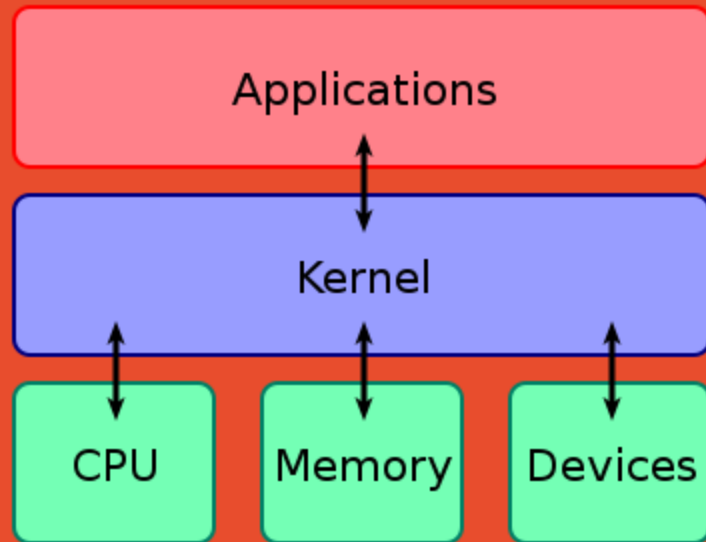


Ok, understood but **What is a kernel?**



The kernel is a computer program at the core of a computer's operating system and generally has complete control over everything in the system





A kernel connects the application software to the hardware of a computer

The kernel's interface is a low-level abstraction layer. When a process requests a service from the kernel, it must invoke a system call, usually through a wrapper function.

There are different kernel architecture designs. Monolithic kernels run entirely in a single address space with the CPU executing in supervisor mode, mainly for speed. Microkernels run most but not all of their services in user space like user processes do, mainly for resilience and modularity. MINIX 3 is a notable example of microkernel design. Instead, the Linux kernel is monolithic, although it is also modular, for it can insert and remove loadable kernel modules at runtime.

This central component of a computer system is responsible for executing programs. The kernel takes responsibility for deciding at any time which of the many running programs should be allocated to the processor or processors.

Then is **free**?

How do they pay for the people that works on it?

**If no one is paying
for that, It should be
completely full of
(nasty) bugs, Right?**

—

Actually it is not, it has less than others because of that

Ok, let's get in. What do I do first?

—

Actually, let's review what NOT to do first

What is **SUDO** and why is it important?

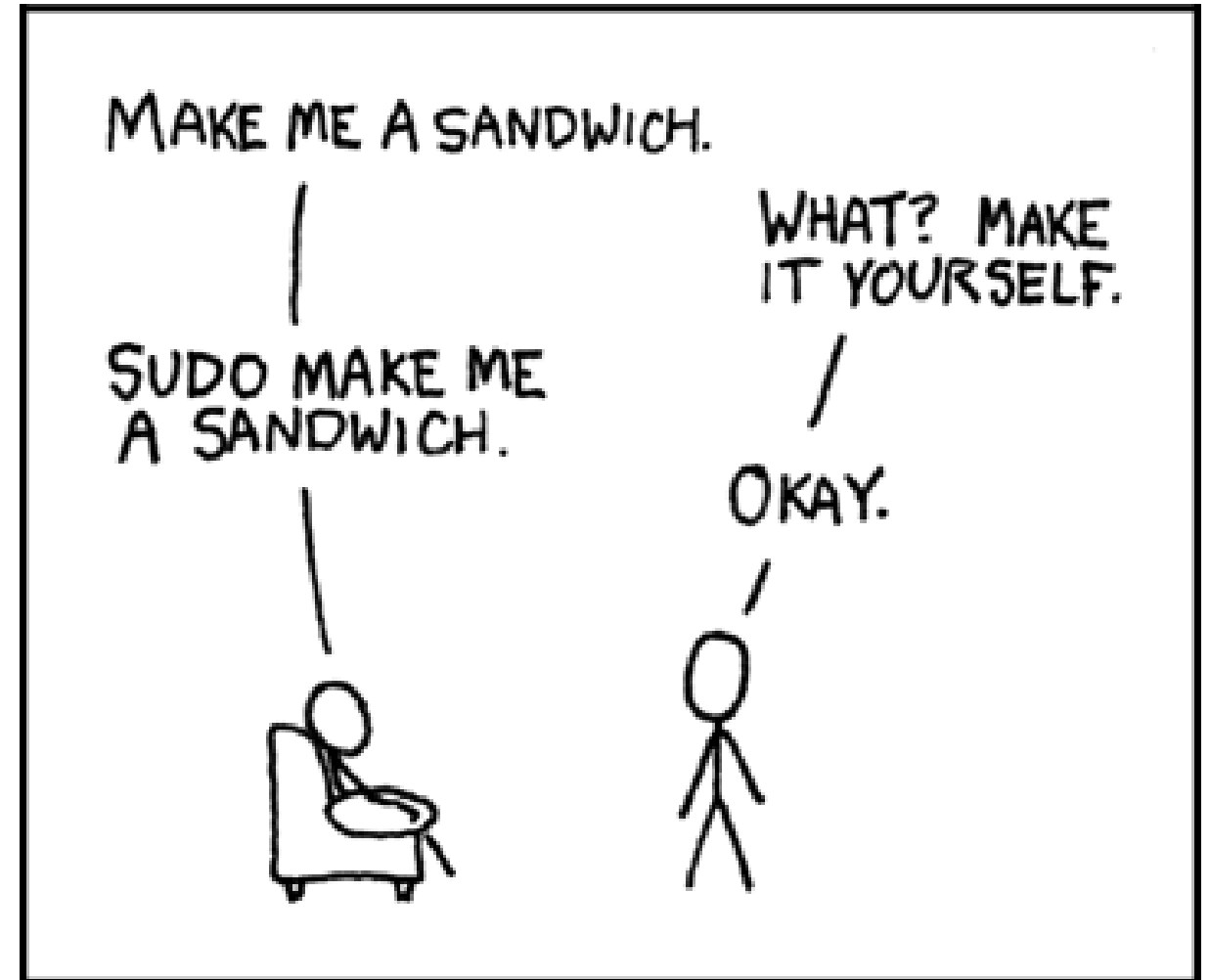


SUDO (SuperUser DO) enables users to run programs with the security privileges of another user, by default the superuser

In order to be able to run sudo, the user must be first in the /etc/sudoers file, and specific permissions and commands can be set for it to be run as superuser

sudo is required for example for installing programs or to modify configurations (typically found in /etc)

sudo can be used as well to switch to another user, as in: sudo john (will log in as john and load his environment)



Things to avoid as a new Linux User

General Ideas

- Do NOT use “sudo su” and only use sudo if you really need it (i.e. for installing programs)
- Always check where you are in the filesystem before doing any changes (you may use “pwd” or modify the prompt to show it)
- Avoid copy/paste if you do not understand what's going on
- Always review that the example/tutorial that you've found, is for the correct distribution (Redhat, Debian, Fedora, etc)

Specific commands

- `dd if=/dev/random of=/dev/sda`
- `wget http://example.com/something -O - | sh —`
- `sudo rm -rf /`
- `chmod 777`
- `:(){ :|:& };: (fork bomb)`

Linux is free!

Linux will let you do anything that you want to do, even if you don't know!

You can go as far as modify the kernel itself!

So instead of looking for an straight answer, first try to understand what's going on and how what you're about to do will affect the OS

..

The Linux philosophy is 'Laugh in the face of danger'. Oops. Wrong One. 'Do it yourself'.

Yes, that's it. ..

Linus Torvalds.

Everything is a file

File Tree Links and Devices

—

Filesystems

—

01

- Everything in Linux is a file
- We can create, edit, delete and view a file (among many other operations)
- Even a folder is a file
- A group of files can be grouped in a folder
- Many folders can create a hierarchy

Filesystems



02

A partition is a logical part of the disk, whereas a filesystem is a method of storing/finding files on a hard disk (usually in a partition).

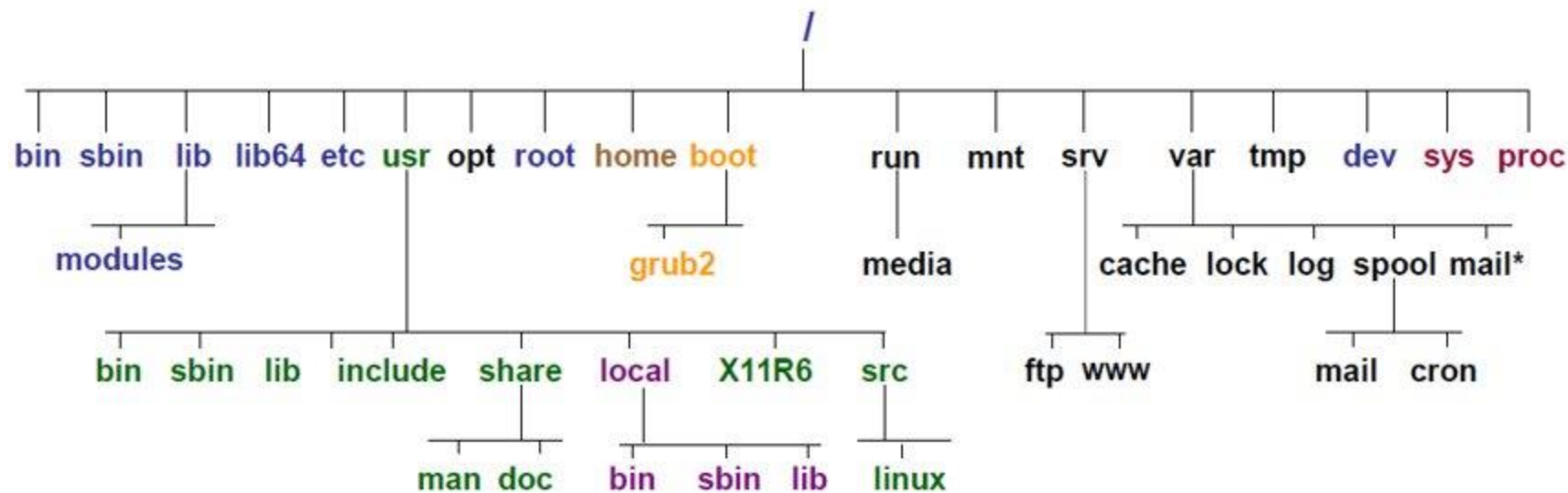
Linux uses FHS (Filesystem Hierarchy Standard), defining a two-layer hierarchy to store files in it.

Filesystems

—

03

Operating system	Associated file system
DOS	FAT16
Windows XP	NTFS
Windows 98	FAT32
Windows 95	FAT16 - FAT32 (for version OSR2)
Windows NT	NTFS
OS/2	HPFS
Linux	Linux Ext2, Linux Ext3



Filesystems



04

Directory	Description	Example Files
/bin	Essential binaries for all users	/bin/bash, /bin/rm
/sbin	System administration binaries.	/sbin/fdisk
/etc	System and services configuration files.	/etc/fstab, /etc/passwd, /etc/group, /etc/hosts
/usr	Unix System Resources. Secondary hierarchy for read-only shared data	/usr/local/* /usr/share/doc

Directory	Description	Example Files
/root	The root user's home directory.	
/home	Local user's home directory.	~/.bashrc, ~/.profile
/var	Contains files that can be modified while the system is running.	/var/log, /var/spool, /var/lock, /var/run
/proc	Virtual FS that exists in memory and display the current state of processes running.	/proc/cpuinfo, /proc/mounts, /proc/filesystems
/tmp	Where programs create temporary files while they are running	

Your interface to the operating system.

Bash

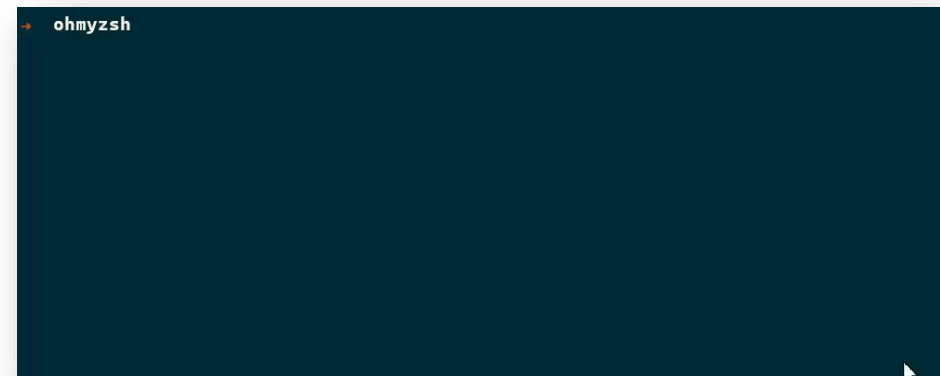
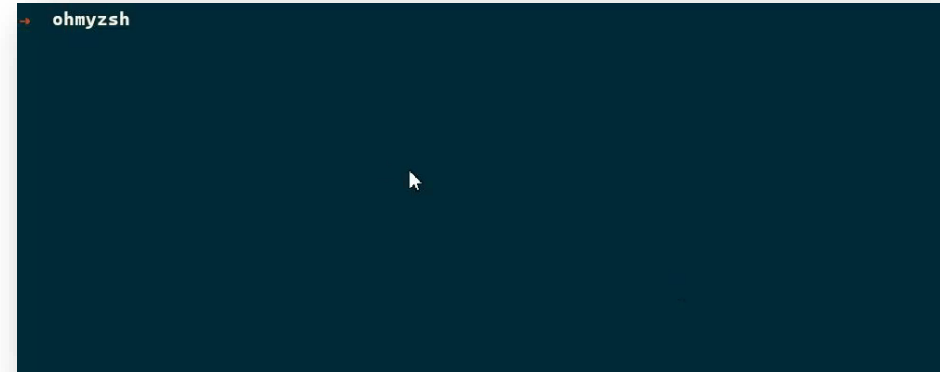
Bourne-again shell



Empower My Shell

I dare you to change your shell.

- Step 1 → Install ZSH
- Step 2 → From github download OhMyZsh framework
- Step 3 → Add zsh-autosuggestion plugin



Invoking the Shell

Special Files

- /etc/profile. Executed automatically at login.
- The first file found from this list: ~/.bash_profile, ~/.bash_login, or ~/.profile
- ~/.bashrc is read by every shell, after the login files.

Arguments

Are assigned in order to the positional parameters **\$1**, **\$2**, **\$3**, etc.

If the first argument is an executable script, commands are read from it, and the remaining arguments are assigned to **\$1**, **\$2**, **\$3**, etc.

The name of the script is available as **\$0**.

If you want to know the number of command-line arguments use **\$#**.

Command Forms

cmd &	bash script.sh &
cmd1; cmd2	cat file1; echo; grep "multiple cmds on the same line"
cmd1 cmd2	cat file2 head -1 tail -3 cut -F"=" -f1
cmd \$((expr))	echo \$((256/4))
cm1 && cmd2	apt-get update && apt-get upgrade
cmd1 cmd2	grep "something" echo "something not found"

Redirection Forms

File descriptor	Name	Abbreviation	Example
0	Standard input	Stdin	cmd < file
1	Standard output	stdout	cmd > file
2	Standard error	stderr	cmd 2> file or 2>&1

Variables

Variable substitution Built-in shell variables Other shell variables Arrays

Variable Substitution

No spaces should be used in the following expressions.

var=value u=up r=right blank=

\$var echo \$var echo "\$var"

\${var} echo \${var} echo "\${var}"

\${#var} echo \${#var} echo "\${#var}"

Declare Variable

declare -a arr=(this is powerfull)

declare -i sum

declare -u upper

declare -r readvar

Built-in Shell Variables

\$#	Number of command-line arguments
\$?	Exit value of last executed command
\$\$	Process number of current process
\$0	First word; that is, command name
\$* , \$@	All arguments on command line (\$1 \$2 ...)

Other Shell Variables

\$PWD	Current working directory (set by cd)
\$OLDPWD	Previous working directory (set by cd)
\$RANDOM	Generate a new random number with each reference
\$SECONDS	Number of seconds since the shell was started
\$PPID	Process number of this shell's parent
\$HOSTNAME	The name of the current host
\$SHELL	Name of default shell

Built-in Commands

Optimizations

Most shell scripts are **quick 'n dirty solutions** to non-complex problems.

Consider the case, though, where a script carries out an important task, does it well, but **runs too slowly**.

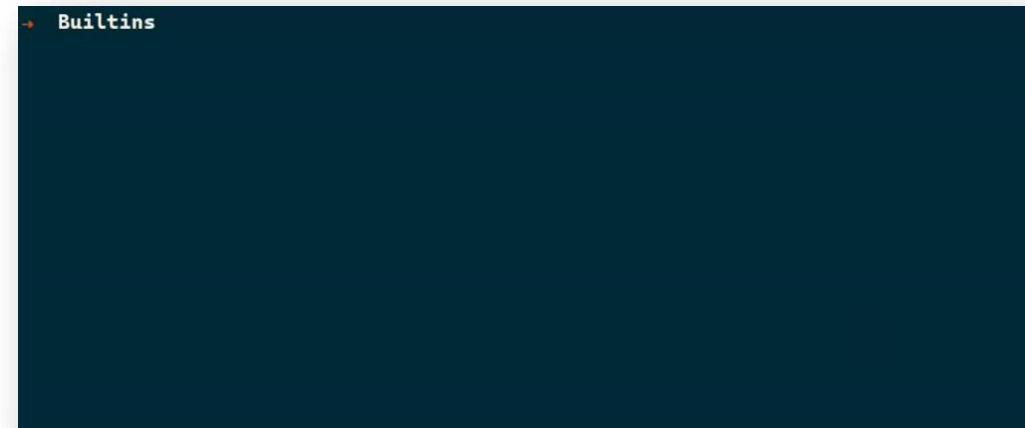
The simplest fix would be to **rewrite the parts** of the script that slow it down.

Check the **loops** in the script. Time consumed by repetitive operations adds up quickly. If possible, **remove time-consuming** operations from within loops.

Builtins execute faster and usually do not launch a subshell when invoked.

Use **builtin** commands in preference to system commands.

Avoid **unnecessary** commands, particularly in a pipe.



man is the system's manual pager



```
man [man options] [[section] page ...] ...
```

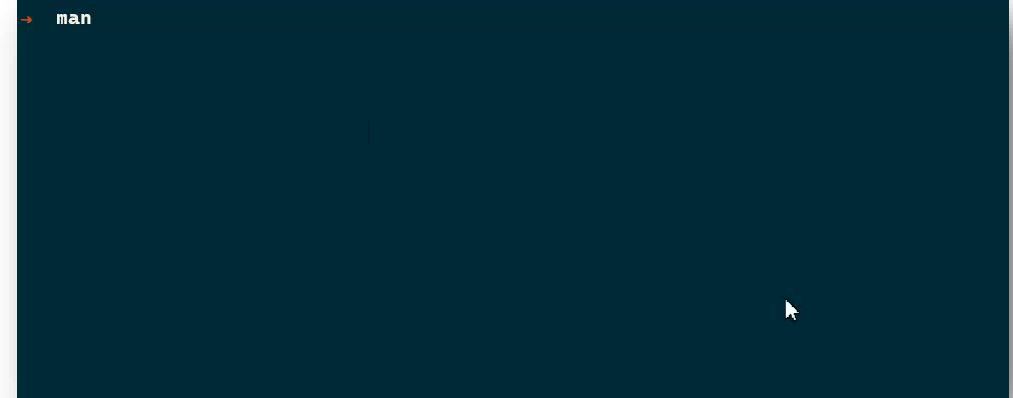
```
man -k [apropos options] regex ...
```

```
man -K [man options] [section] term ...
```

```
man -f [whatis options] page ...
```

```
man -l [man options] file ...
```

```
man -w|-W [man options] page ...
```



curl

is a tool to transfer data from or to a server

`curl -X / --request <command> ...`

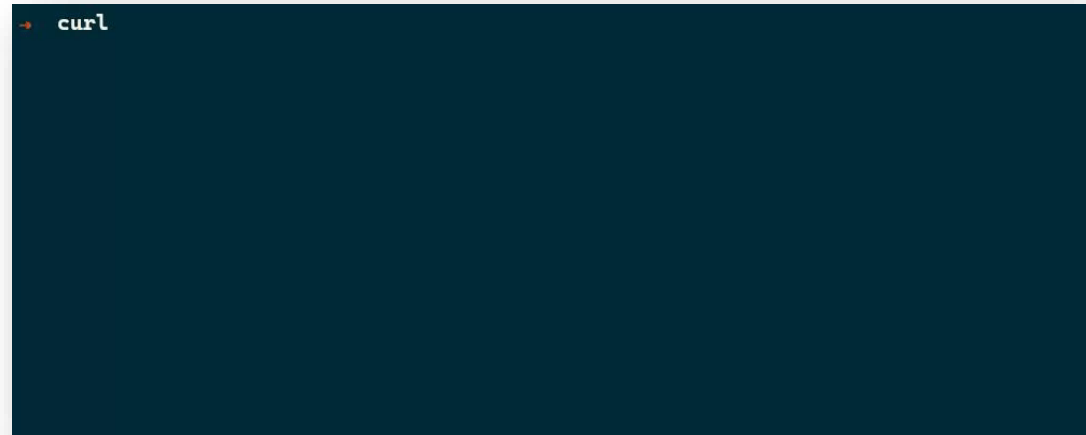
`curl -d / --data <data> ...`

`curl -s / --silent ...`

`curl -o / --output <file> ...`

`curl -H / --header <header> ...`

`curl -F / --form <name=content> ...`



sed is a stream editor used to perform basic text transformation

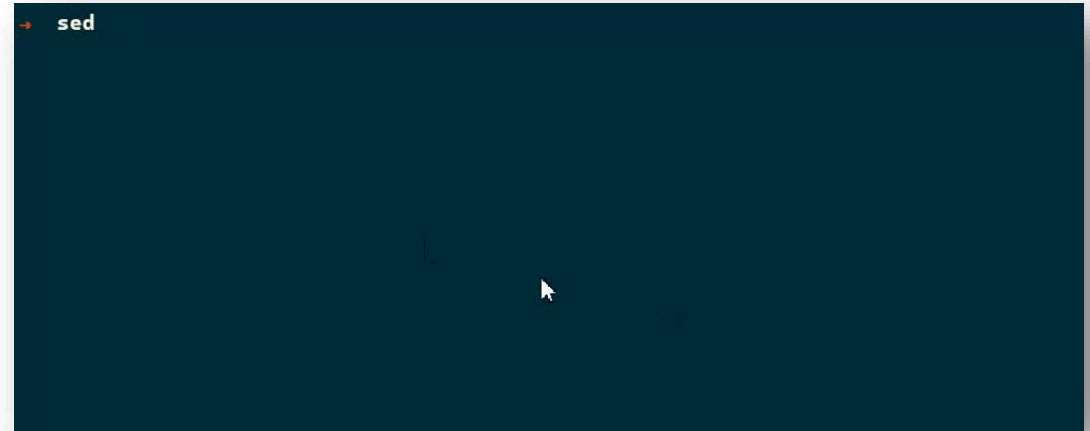
```
sed -e script / --expression=script ...
```

```
sed -f script-file / --file=script-file ...
```

```
sed -i[SUFFIX] / --in-place=[SUFFIX] ...
```

```
sed -s / --separate ...
```

```
sed -n / --quiet / --silent ...
```





grep searches for patterns in each file

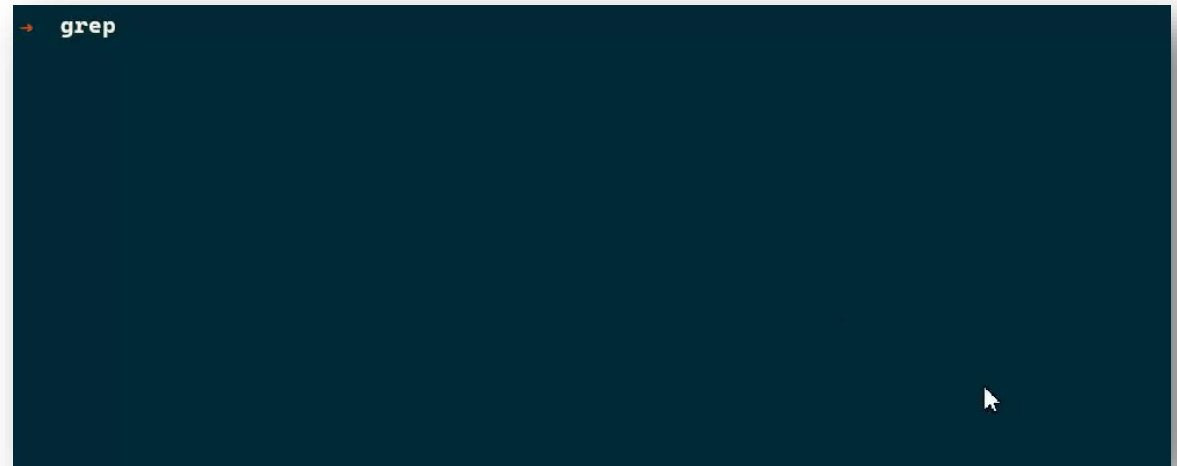
`grep -i / --ignore-case ...`

`grep -v / --invert-match ...`

`grep -c / --count ...`

`grep -l / --files-with-matches ...`

`grep -o / --only-matching ...`



find search for files in a directory hierarchy

```
find -maxdepth / -mindepth ...
```

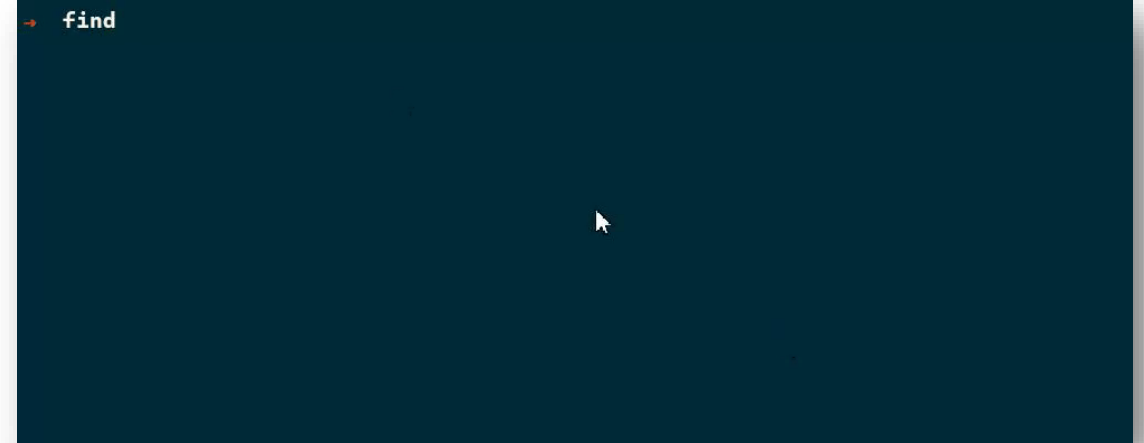
```
find -name / -iname ...
```

```
find -perm mode, -mode, /mode ...
```

```
find -regex pattern ...
```

```
find -size n[cwbkMG]...
```

```
find -type c ...
```



awk pattern scanning and **processing** **language**

```
awk -F / --field-separator fs ...
```

```
awk -v var=val / --assign var=val ...
```

```
awk -f program-file / --file program-file ...
```

```
awk -O / --optimize ...
```

```
awk -mf NNN / -mr NNN ...
```



Homework!



Select a linux distribution

Research the different linux distributions and choose one to be installed in your personal computer, write down the reasons why you have chosen it.



Show off

Show a picture of your Linux distribution installed (may be running from an USB stick) and a screenshot of your home directory from the CLI



This task may be done with virtual machines, but virtualization will be taught later

Enlaces de Interés:

- [Virtual x86](#)
- <https://www.xmind.app/m/WwtB/> (listado de comandos linux)
- <https://overthewire.org/wargames/bandit/>
- <https://www.hackthissite.org/>

Llenar el feedback:

https://forms.office.com/pages/responsepage.aspx?id=eME_CzC3i06YQ-gSWSN7d3Xoyx1ey21lp4rb9yVxOoRURVo0NUwwRUM0VIJBS0g3MlcwTTk1RExOQSQIQCN0PWcu



Thank you!