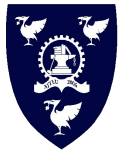


Xi'an Jiaotong-Liverpool University

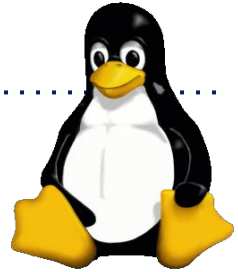
西交利物浦大學

CS108 Operating System Concepts

Lab 5 (2)

Linux 1

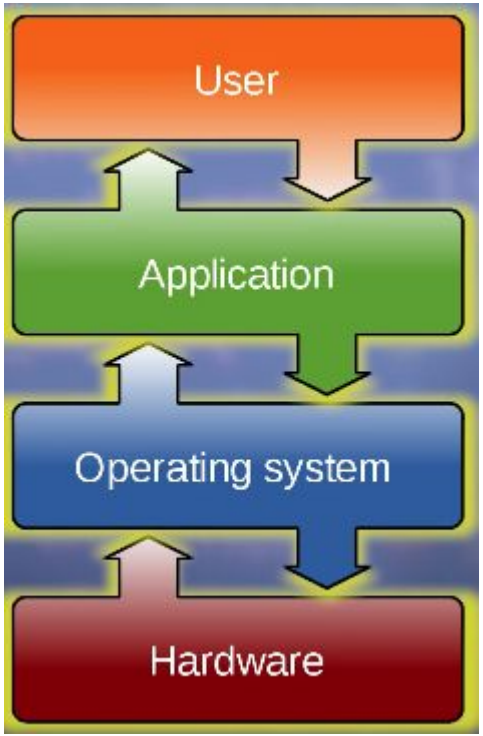
Linux



- Linux is a Unix-like operating system that is open source
  - created by Linus Torvalds
  - enhanced by thousands of programmers
  - available to the world for free

- Let us first review OS and Unix in this lab

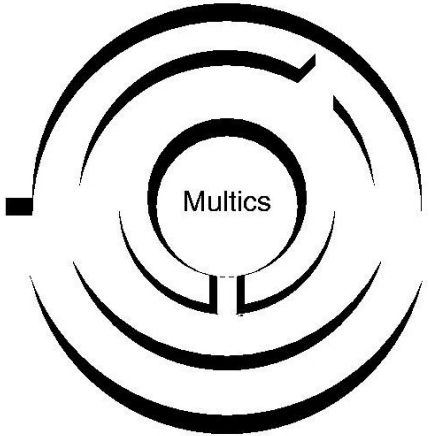
Operating System



- An operating system is an intermediary between the hardware and the applications that we use
- Users use the services rendered by applications
- Applications use the services provided by operating systems
- An operating system exploits the hardware resources that are at its disposal to render these services to the applications
    - example of services: file management, memory management

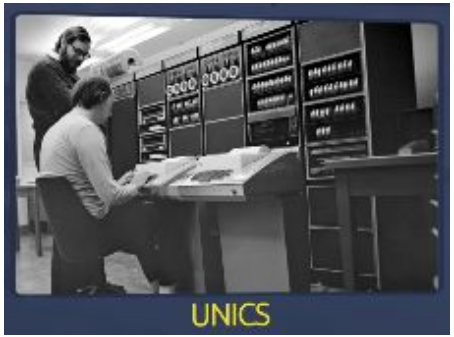
Unix Genesis (1)

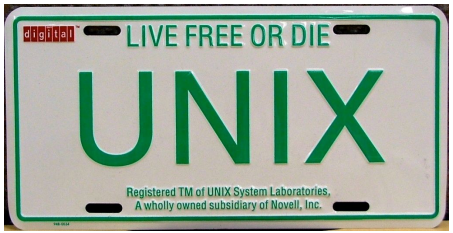
Project MAC



- The first ideas relating to Unix were started with *Project MAC* — project on mathematics and computation, founded by MIT, ARPA and NSF.
- Its main goal was to realize a timesharing system that would allow a wide community of users to simultaneously access the hardware and software resources of a single computer from multiple locations.
- Six years later, Project MAC, Bell Laboratories and General Electric, developed *MULTICS*, the Multiplexed Information and Computing Service, which service also incorporates features such as file sharing, file management, and system security.

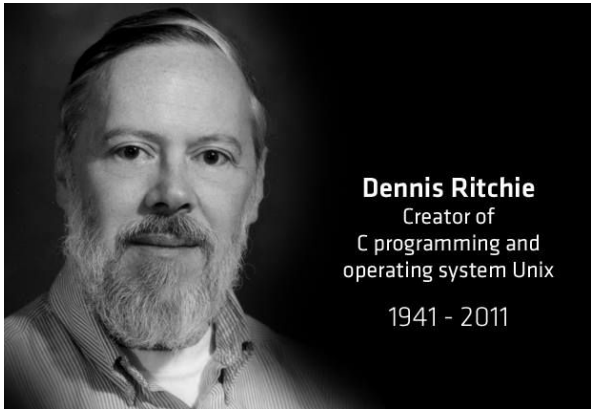
Unix Genesis (2)





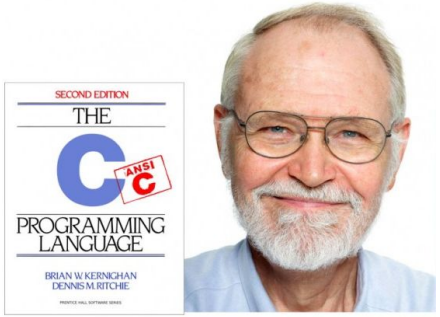
- Bell Labs contribution to MULTICS was on the design and writing of the operating system code.
- Bell Labs pulled out of the project at 1969.
- Some Bell Labs engineers working on the project, led by **Ken Thompson** and **Dennis Ritchie**, decided to pursue an alternative approach to the system, which they considered to be complex.
- In 1969, they began the development of a single-user operating system on their own account and without any support from Bell Labs.
- As a pun on MULTICS, they called the system *UNICS*.
- In 1970, the system changed from single user to multiple users, and the spelling changed to **UNIX**.

Unix Genesis (3)



- At the time, the system was written in the B programming language invented by Ken Thompson.
- In 1971, Dennis Ritchie improved his colleague Ken's language and called it the new B.
- By 1972, the changes to the B language became so significant that Dennis Ritchie renamed his new language **the C language**.
- Ken jumped on the opportunity and rewrote all of the code making up the Unix operating system in this improved C programming language.
- After two to three years of work, Ken sent the C source code of the Unix operating system to universities and research centers for educational purposes all around the world.

Unix Genesis (4)

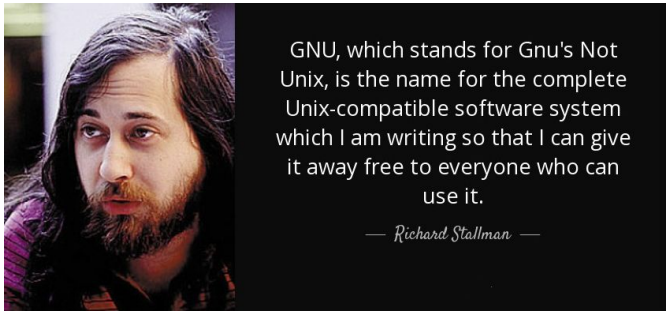


- In 1978, Dennis Ritchie — jointly with his colleague, **Brian Kernighan** — wrote the book The C Programming Language.
- In 1983, Ken and Dennis received the highest distinction in the computer science field for this invention, **the Turing Award**.
- The basics of Unix are ubiquitous today.
  - The *Mac OS* installed on Apple computers is a derivative of Unix, as is the *iOS* operating system for iPhone, iPad, the *Android* system for Google phones, and even the *Linux* system installed on the vast majority of today's service and connected objects.

Linux Genesis (1)

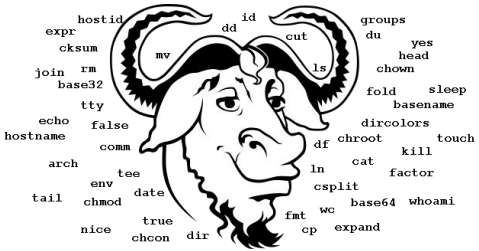
- Also in 1983, **Richard Stallman** of MIT launched the **GNU Project**.
- GNU is a recursive acronym, referring to itself, that stands for *GNU is Not Unix*.
- The GNU Project is an open and free, collaborative project to develop and provide software compatible with Unix.
- The big difference between the GNU Project and Unix is that Unix is owned by Bell Labs, which sells licenses for use or modification,
  while the GNU Project is *free*, and anyone can use or modify the entire project.

Linux Genesis (2)



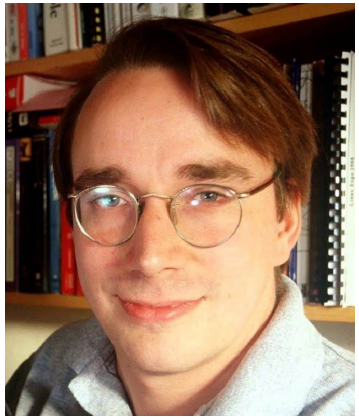
GPL Free Software

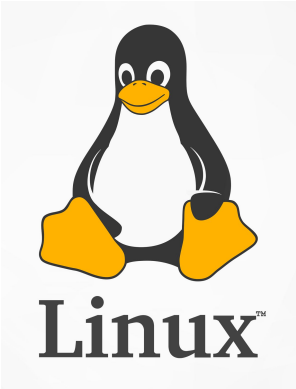
Free as in Freedom



- Richard Stallman is a strong advocate for *free* and *open source* software.
- In 1989, he conceived of the **GNU General Public License**, which aims to preserve the freedom to use, study, modify, and distribute software and its derivative versions.
- By 1990, the GNU Project included a lot of software already, text editors, a graphical user interface similar to Windows, libraries, *the GNU C Compiler* — **GCC** — a C language compiler, et cetera.
- The problem was that one had to use this free software on the Unix operating system that itself is proprietary and not open or free.

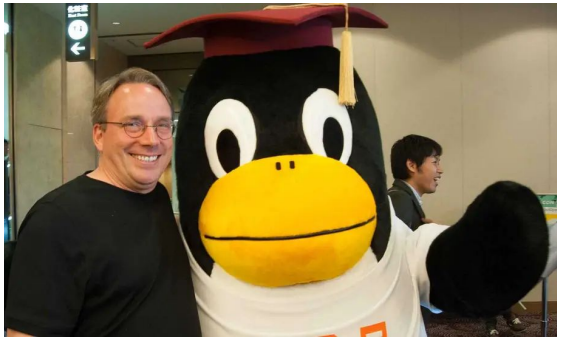
Linux Genesis (3)





- This is where **Linus Torvalds**, a 21-year-old Finnish CS student at the University of Helsinki, intervened.
- He was frustrated by these proprietary licenses that limited the use of the operating system.
- On August 25, 1991, he sent a message to the community that started like this, *"Hi, everyone. I am doing a free operating system. It's just a hobby. It will not be as big and as professional as the GNU Project."*
- This project, which later became the **Linux kernel**, was written specifically to be independent of an operating system, and was meant to run on Linus Torvalds' PC, with an 83.86 processor, developed using the GNU C Compiler, which to this day is still the main choice for compiling Linux.

Linux Genesis (4)



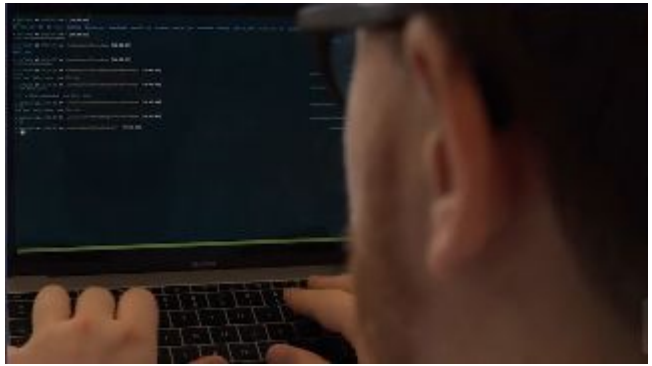
- Just as in the case of the GNU Project, the community formed, and developers of the GNU Project quickly integrated free software to run on the Linux kernel, which itself is also free.
- In 1992, the first Linux distributions were released and consisted of the free Linux operating system and a collection of free GNU Project software tools, such as editors, libraries, compilers, et cetera.
- By 1993, there were already more than 100 developers from around the world who worked on modifying and improving Linux.
- 1993 was also the year of some very popular Linux distributions, for example *Debian*.

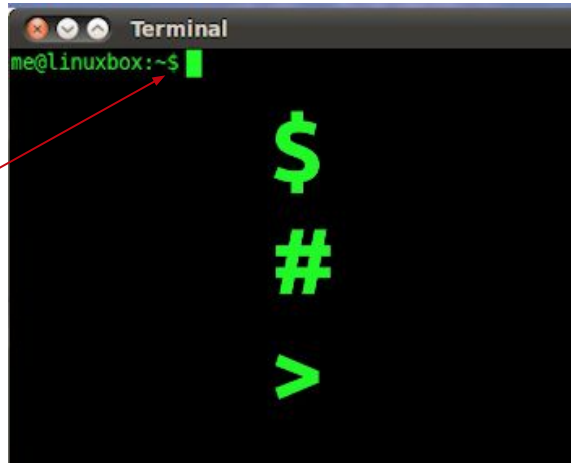
Linux Genesis (5)

- By the end of the '90s, Dell, IBM, and HP announced compatibility of their hardware with Linux.
- During the 2000s, Linux was deployed more and more as the operating system that runs *web servers*.
- In the 2010s, Linux and Unix-based operating systems became the most widely used systems on *internet servers*, taking about 70% of the market share, and *smartphones*, around 90%, since the Android OS is based on Linux and iOS is based on Unix.
- Linux and Unix-based operating systems can further be found in *game consoles*, for example Playstation, Internet Boxes, Wi-Fi routers.
- The Linux OS or other Unix derivatives have become a part of many aspects of our everyday life.

Command Line Interface

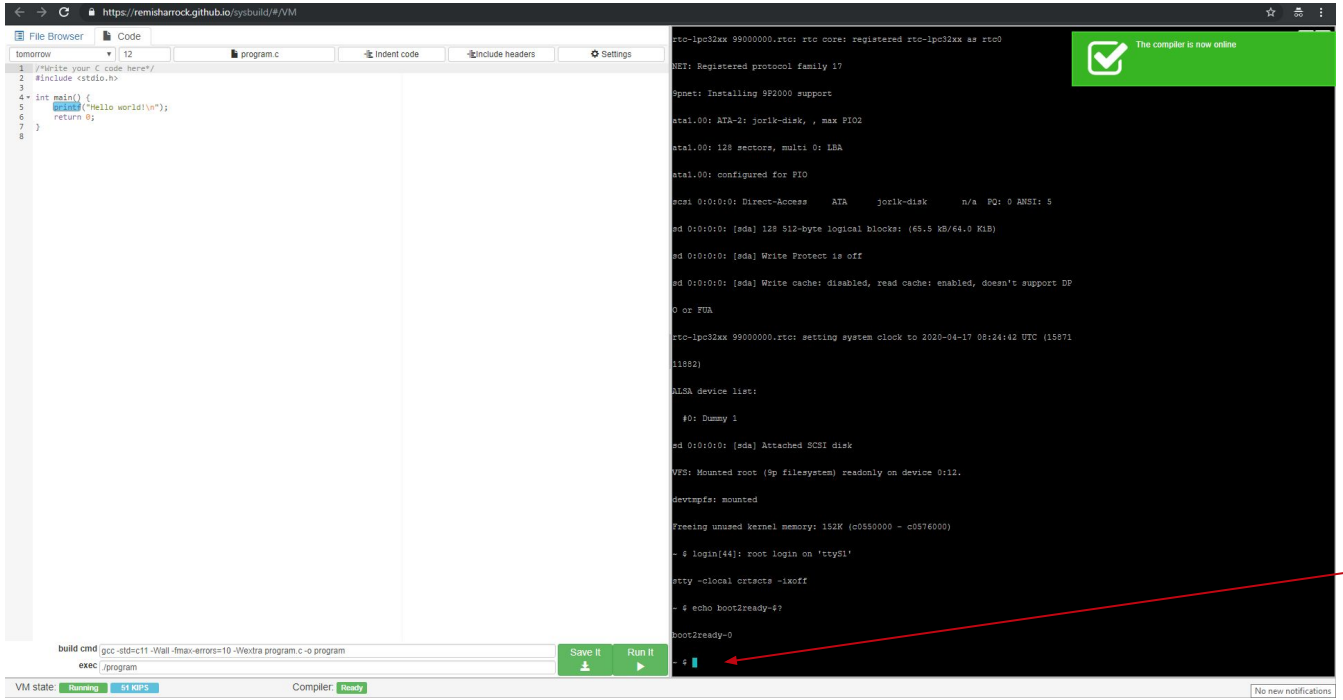
- **A command line interface (CLI)**, is a human machine interface in which communication between the user and the computer takes place in ***text mode***.
- The user ***types a command*** to ask the computer to perform an operation; the computer then displays text which is the result of the execution of the command.
- When a command line interface is ready to receive a command from the user, it indicates this by displaying **a command prompt**.
- This prompt consists of information at the beginning of the line: the user's account name, or the computer name, or the current working directory, or the date; and it ends with a character, often ***the dollar***, or ***the pound sign*** or ***the greater than sign***, to show that the system is ready to receive a command.





WebLinux

- In this lab, we are going to use WebLinux to learn about **Linux commands**.
- Now, go to https://remisharrock.github.io/sysbuild/#/VM



type your commands here in the command prompt

Commands in Linux (1)

- The Linux OS more than 100 software applications *usable from the command line*.
- The elementary commands are of the form: command at the beginning, then space, options, then space, then files_or_data.
  - the options and the files or data are *optional*.

```
command options files_or_data
```

- The ***command*** at the beginning of the line is the name of a software application.
  - This can be an operating system command or another software application written by a user, often in the C programming language.
- Command line ***options*** modify the execution of a command.
  - The effect of the options depends on the command.
  - Generally, the options immediately follow the name of the command on the command line and they are separated by spaces.

Commands in Linux (2)

command options files_or_data

- Finally, the ***files_or_data*** are the program entries.
- For example, if you want to run the application that *displays a calendar* in text mode, you type the command **cal** and press the Enter key.
  - If we add the -j (minus j) option to it, it displays the calendar in Julian days (that is, the number of days elapsed since January 1st).

cal

```
~ $ cal
     July 2018
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

cal -j

```
~ $ cal -j
          July 2018
 Su  Mo  Tu  We  Th  Fr  Sa
182 183 184 185 186 187 188
189 190 191 192 193 194 195
196 197 198 199 200 201 202
203 204 205 206 207 208 209
210 211 212
```

try now in WebLinux !

whoami

- *whoami* will display the username that is logged with this system
  - on WebLinux, you don't have to type username and a password because it is already configured to automatically logged in with a specific user: user, so the username is user and there is actually no password to boot WebLinux
  - on a Linux system, you will have to log first with a username and password
    - recall that Linux is multi-user so you can have different users with different passwords, and each user will have a home directory with different files

--help option, logname, echo

- add an option --*help* to show a little help of commands
  - Try: whoami --help
  - Try: logname --help

- *logname* works like whoami
- *echo* prints what you type after
  - Try: echo hello

echo $0, shell

- A special echo, *echo $0* will display the name of the interpreter of all the commands that you are using for the system
  - its name is sh, stands for shell
- There are different shells, and the shell is interpreting the command on Linux
  - it's an application/software that will get all the lines when you type Enter
  - it will interpret them and do some stuff or ask the system to do something
- Here we can see that the shell WebLinux uses is sh

Other Basic Commands

- *hostname* displays the name of the computer that is registered in the system
  - on web Linux, that name is openrisk, so openrisk is the name of the computer that is running WebLinux
- *uname* displays the name of the system: Linux
- *history* displays all the history of the commands that you just typed
  - the first two are ones that are automatically done when WebLinux is booting
- To go back to an old command, you can use the *up arrow*
  - after that, you can use *down arrow* to go down that list of commands
- *clear* to clear the screen (duh?)
- *uptime* displays how many minutes since the system was booted
- *date* displays the date, check out the help to format the date
- to show help, on the real Linux system, use *man* which is the manual of command
  - it was disabled in WebLinux because its big size is too slow to download

top

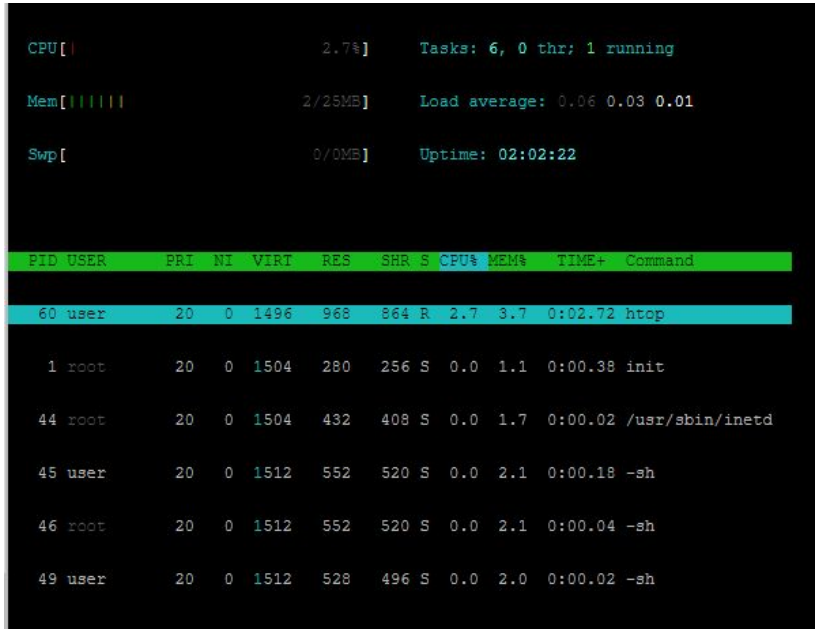
- So far, the commands we've tried go back directly to the command prompt
- There are commands that don't, and we will have to find a way to exit these programs to go back to the command line prompt
- Type top --help
  - you will see that to exit top, keys are: Q, ^C
- Now try to execute top and then exit using either of those keys
- *top* displays the programs/processes currently running, including top itself, and the state of the system

read as carrot C
type by Ctrl + C

```
Mem: 3224K used, 22720K free, 0K shrd, 0K buff, 377164K cached
CPU:  0.3% usr  0.7% sys  0.0% nic 98.8% idle  0.0% io  0.0% irq  0.0% sirq
Load average: 0.00 0.00 0.00 1/28 59
  PID  PPID USER     STAT   VSZ %VSZ CPU %CPU COMMAND
   59    45 user     R     1504  5.7   0  1.1 top
   45     1 user     S     1512  5.8   0  0.0 -sh
   46     1 root     S     1512  5.8   0  0.0 -sh
   49    46 user     S     1512  5.8   0  0.0 -sh
    1     0 root     S     1504  5.7   0  0.0 init
```

htop

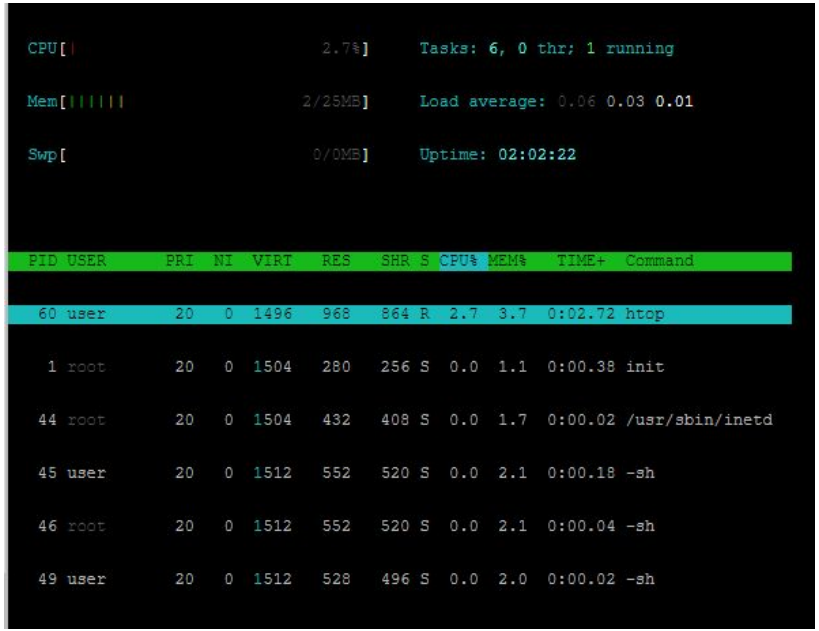
- *htop* displays information similar to top, but in a more graphical way



can you find out how to exit and go back to the command prompt?

htop

- *htop* displays information similar to top, but in a more graphical way

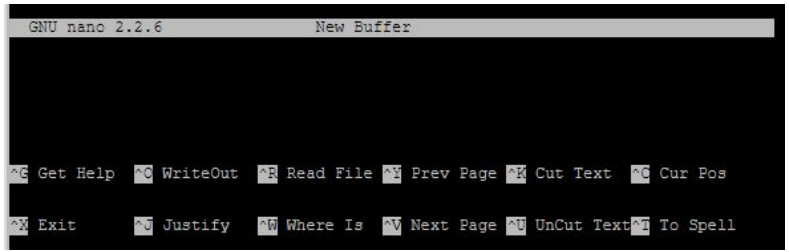


can you find out how to exit and go back to the command prompt?

on bottom right corner: F10 Quit
(may also need press Fn key)
Ctrl + C and Q work as well

nano

- *nano* is a text editor



- type Ctrl + X to exit
- go back to nano, and try Ctrl + C
  - it did something, but it did not exit the program,
    you may need to type Ctrl + X multiple times to exit

vim

- *vim* is another text editor
- vim is notoriously difficult to exit from, you may have seen some internet jokes about it
  - try exit by

- Let's try again, but now type the i key, it goes to insert mode and the menu is gone, but the program is still running, and typing :q does not work!
- Now try Esc, then :q Enter, you will be ask to add exclamation mark to override and exit

Another Commands

- Try the following commands and see what happens:
  - *hello*
  - *worm*
  - *firework*
  - *rain*
  - *hanoi* (type a number 1, 2, or 3)
  - *knight* (type ? for the rules)

  and try to exit using Q or Ctrl + C

- We will continue about Linux next week