

# Introduction to Linux

Francisco Fuster Tormo

13 - 03 - 18

[ffuster@carrerasresearch.org](mailto:ffuster@carrerasresearch.org)

# What is Linux?

Operating System (OS) based on UNIX

- Multiuser → Allows more than one user at the same time
- Multitasking → Allows more than one program executed at the same time
- Networking → Allows network access
- Various user interfaces → Different “desktops” allowed

Other UNIX-based OS?

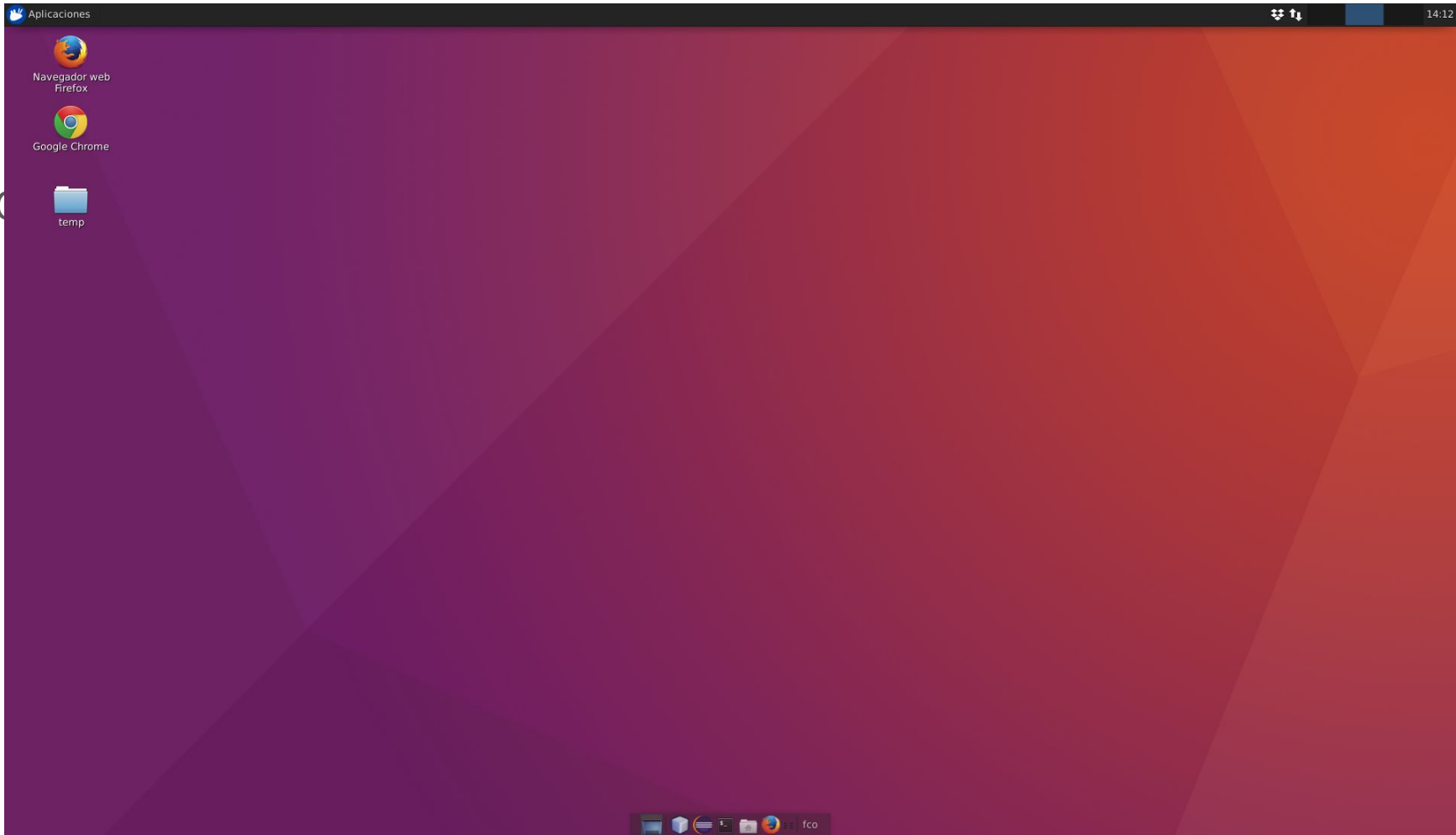
- Mac OS
- Android

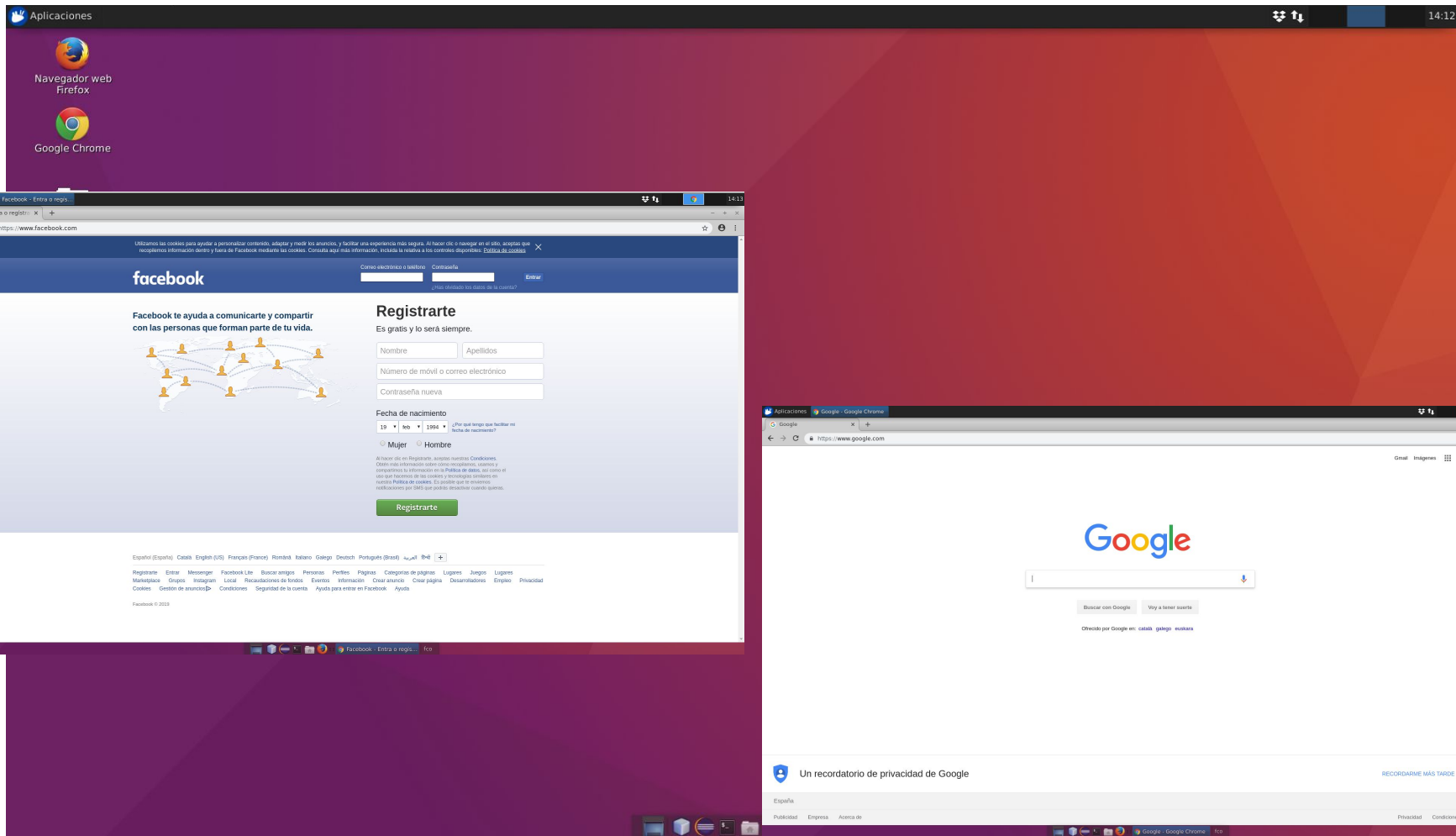
# How many Linux are?

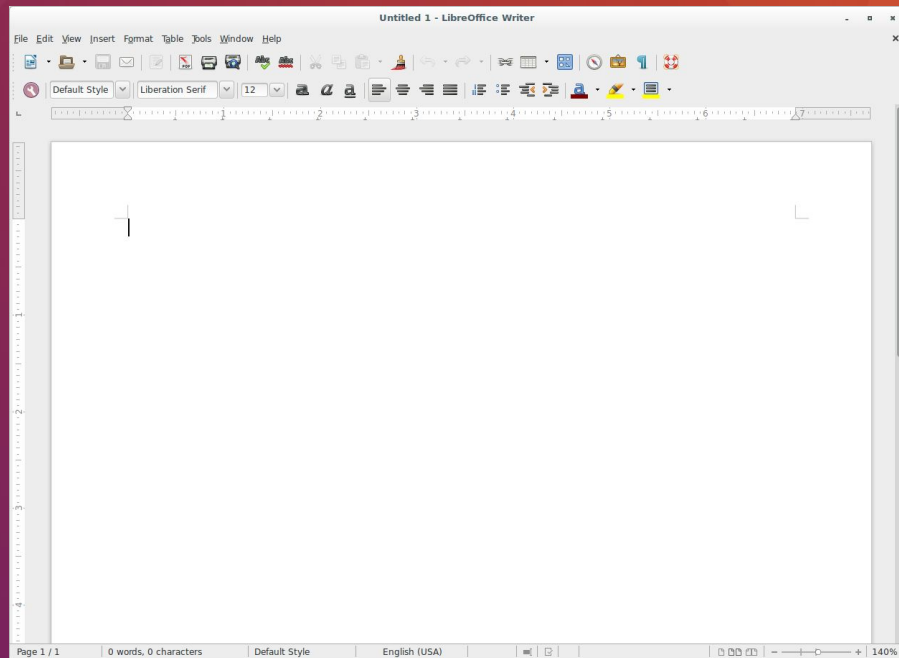
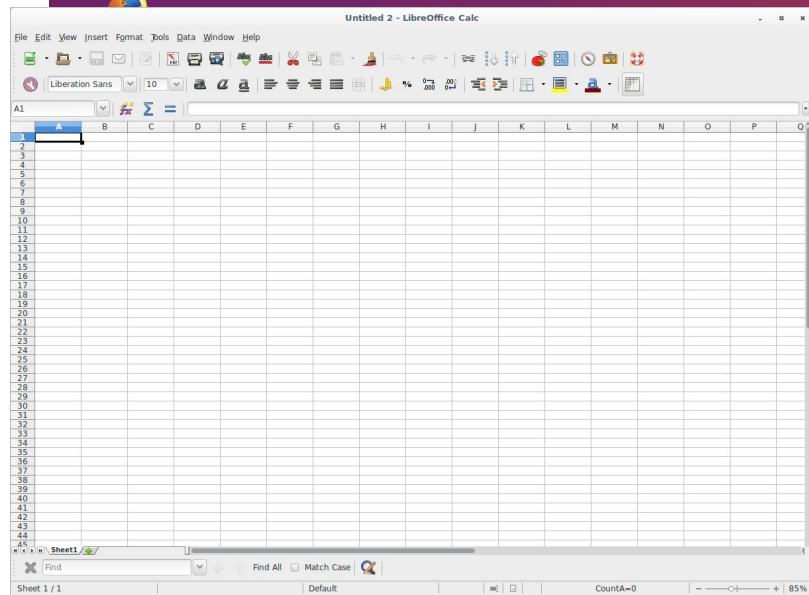
- Debian
- Ubuntu
- Linux Mint
- Red Hat
- Fedora
- OpenSuse
- CentOS
- Gentoo
- ...

# Is Linux difficult to use?

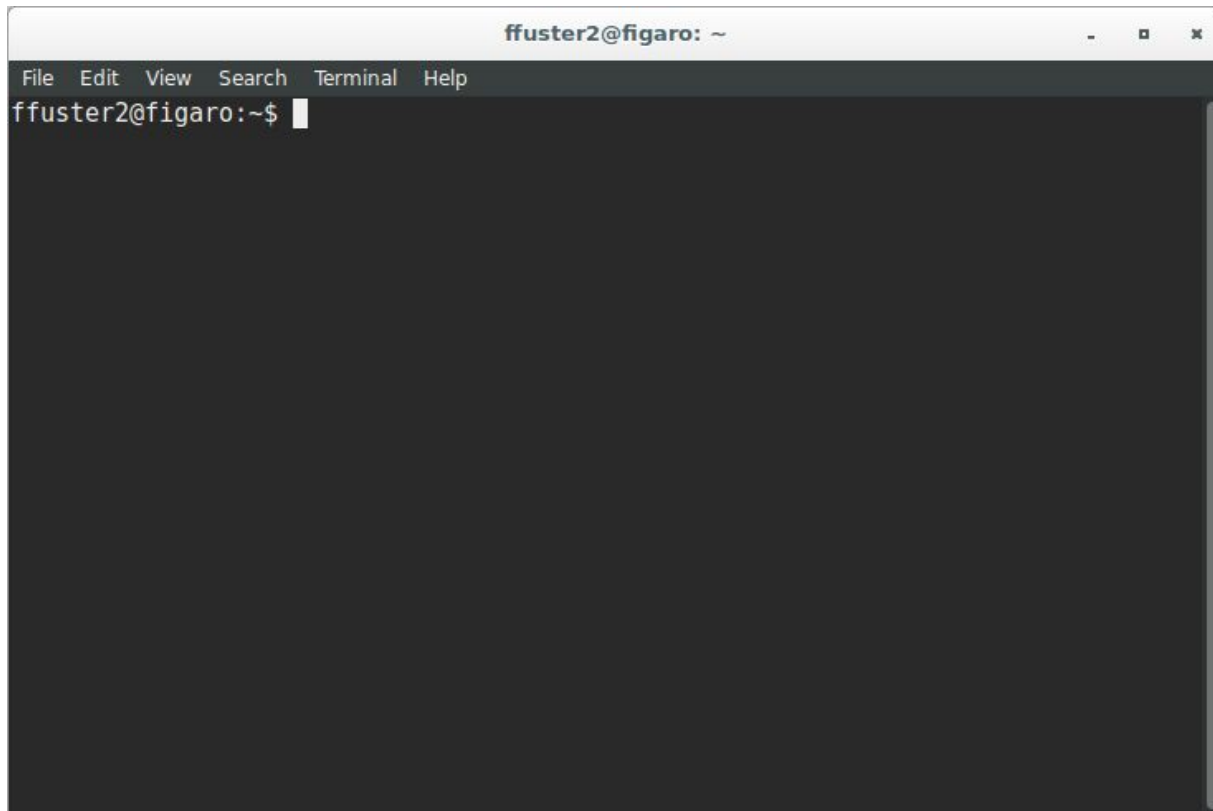
Graphic User Interface (Desktop)





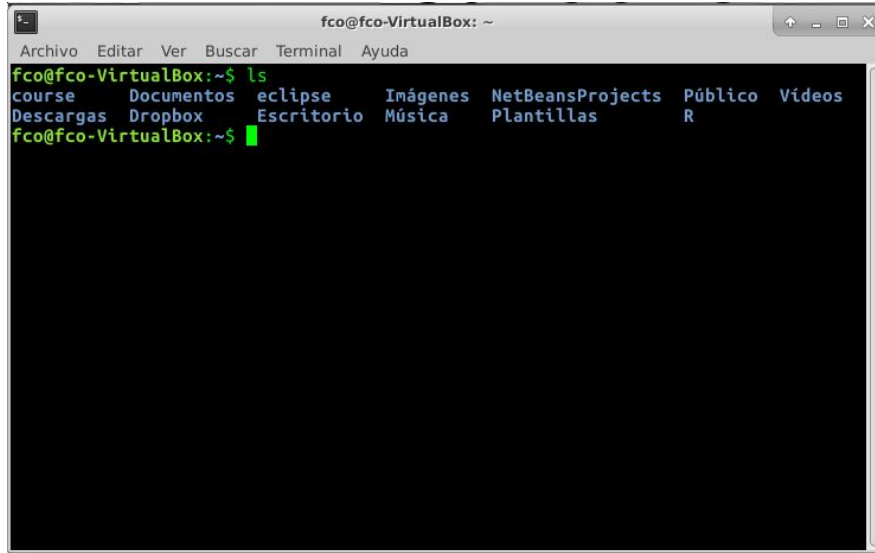


# But...



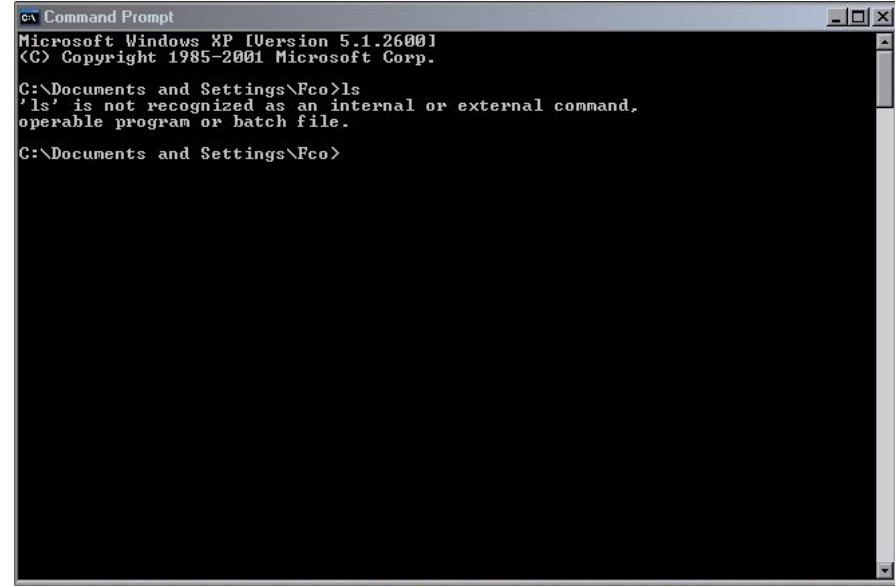


# Differences between Windows & UNIX



A terminal window titled "fco@fco-VirtualBox: ~" with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The prompt is "fco@fco-VirtualBox:~\$". The command "ls" has been entered, and the output is a list of files and directories: "course", "Documentos", "eclipse", "Imágenes", "NetBeansProjects", "Público", "Videos", "Descargas", "Dropbox", "Escritorio", "Música", and "Plantillas". The prompt is now "fco@fco-VirtualBox:~\$" with a green cursor.

```
fco@fco-VirtualBox:~$ ls
course      Documentos  eclipse     Imágenes   NetBeansProjects  Público  Videos
Descargas   Dropbox    Escritorio Música      Plantillas        R
fco@fco-VirtualBox:~$
```



A Windows Command Prompt window titled "C:\ Command Prompt". It shows the Microsoft Windows XP [Version 5.1.2600] copyright notice. The user is in the directory "C:\Documents and Settings\Fco" and has entered the command "ls". The output is an error message: "'ls' is not recognized as an internal or external command, operable program or batch file." The prompt is now "C:\Documents and Settings\Fco>".

```
C:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Fco>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Documents and Settings\Fco>
```

# Special considerations

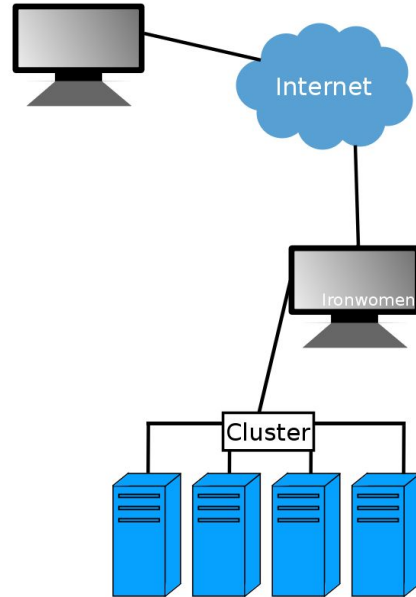
**command** -x -y --option [other\_parameters]

↑  
Command is in bold


↖  
Command options start by '-' or '--'

↖  
Other parameters are surrounded by '[' ].  
Square brackets are written just to  
distinguish between parameters

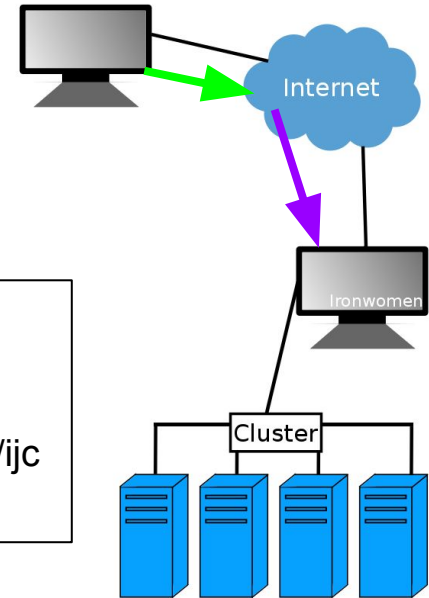
# Course connection schema



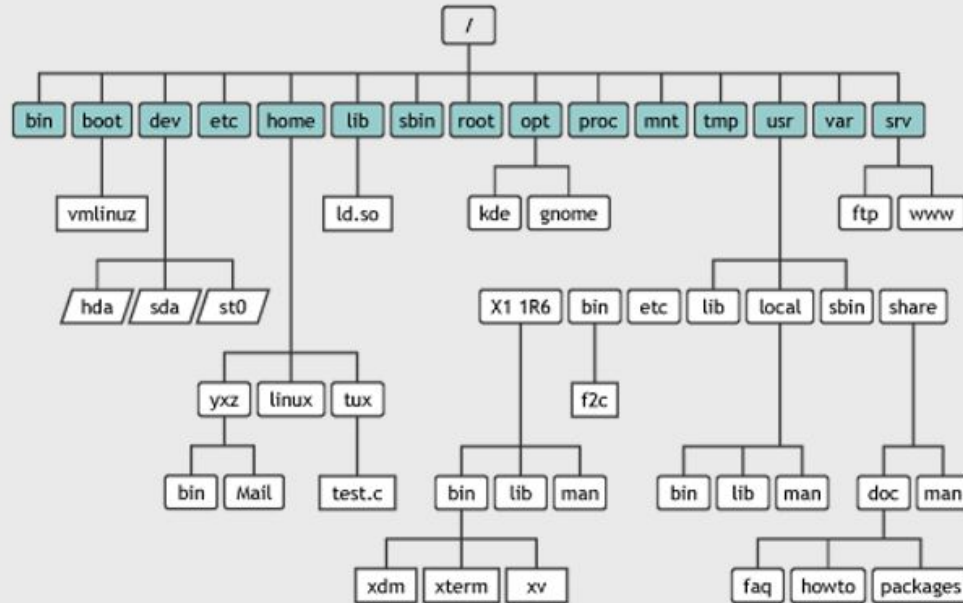
# Linux commands (0). Connecting to ironwomen

1. Open virtual box
2. Start EpiBioinfoVPN virtual machine
3. User: → vpnuser
4. Password: → vpnuser
5. Open “network connect” 
6. Go to profiles → new
7. Type this data → 

Name this profile: Epi\_student  
Username: student  
Password: ljc2019  
Server/URL: https://172.27.2.252/ljc  
Realm: Temp
8. Click “OK”
9. Click on “Connect”
10. Open terminal and write: **ssh** msuser@ironwomen
11. Password: → nZM0M8S7



# The Linux filesystem



File Edit View Search Terminal Help

```
ffuster2@figaro:~$ cd /home/yxz/Mail
```

```
ffuster2@figaro:/home$ cd yxz/Mail
```

# Linux basic commands (1). Folder navigation

- **cd** [folder] → Navigation in directories
- **ls** → List the files/directories in a path
- **pwd** → Get absolute path of current directory
- **mkdir** [folder\_name] → creates the folder [folder\_name] in the current path
- **find** [path] -name [pattern] → Searches files in the specified position

```
fco@fco-VirtualBox:~$ ls
Descargas  Dropbox  Escritorio  Música          Plantillas  R
Documentos eclipse  Imágenes    NetBeansProjects  Público     Vídeos
fco@fco-VirtualBox:~$ cd Escritorio/
fco@fco-VirtualBox:~/Escritorio$ ls
firefox.desktop  google-chrome.desktop  temp
```

```
fco@fco-VirtualBox:~/Escritorio$ pwd
/home/fco/Escritorio
```

```
fco@fco-VirtualBox:~/Escritorio$ find . -name new*
./new_dir
```

```
fco@fco-VirtualBox:~/Escritorio$ mkdir new_dir
fco@fco-VirtualBox:~/Escritorio$ ls
firefox.desktop  google-chrome.desktop  new_dir  temp
```

# Exercise (1)

Connect to ironwomen

Create the directory 'your\_name' in the folder you are

List all the elements in that home folder

Get absolute path of your folder

# Exercise (1)

Connect to ironwomen

```
fco@vpn:~$ ssh msuser@ironwomen
msuser@ironwomen's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar  9 15:08:11 2019 from 10.19.18.1
```

Create the directory 'your\_name' in the folder you are

List all the elements in that home folder

Get absolute path of your folder

```
msuser@ironwomen:~$ mkdir francisco_fuster
msuser@ironwomen:~$ ls
francisco_fuster
msuser@ironwomen:~$ pwd
/home/labs/mslab/msuser
msuser@ironwomen:~$
```



## Linux basic commands (2). Copy, move, remove files

- **touch** [file\_name] → creates an empty file called [file\_name]
- **cp** [file\_origin] [file\_destination] → copy [file\_origin] in [file\_destination]
- **scp** [user]@[server]:[path\_file] [destination] → copy files from remote server
- **rsync** [origin\_folder] [destination\_folder] → synchronize both folders
- **rsync** [user]@[server]:[path\_file] [destination] → synchronize from remote server
- **mv** [file\_origin] [file\_destination] → move [file\_origin] to [file\_destination]
- **rm** [file] → removes [file]

## Linux basic commands (2). Copy, move, remove files

- **touch** [file\_name] → creates an empty file called [file\_name]
- **cp** [file\_origin] [file\_destination] → copy [file\_origin] in [file\_destination]
- **rsync** [origin\_folder] [destination folder] → Synchronize both folders
- **mv** [file\_origin] [file\_destination] → move [file\_origin] to [file\_destination]
- **rm** [file] → removes [file]

```
fco@fco-VirtualBox:~$ cd course/
fco@fco-VirtualBox:~/course$ ls
fco@fco-VirtualBox:~/course$ touch file_name.txt
fco@fco-VirtualBox:~/course$ ls
file_name.txt
fco@fco-VirtualBox:~/course$ cp file_name.txt file_name_copy.txt
fco@fco-VirtualBox:~/course$ ls
file_name_copy.txt  file_name.txt
fco@fco-VirtualBox:~/course$ mv file_name_copy.txt file_name_new.txt
fco@fco-VirtualBox:~/course$ ls
file_name_new.txt  file_name.txt
fco@fco-VirtualBox:~/course$ rm file_name.txt
fco@fco-VirtualBox:~/course$ ls
file_name_new.txt
fco@fco-VirtualBox:~/course$
```

## Exercise (2)

Disconnect from ironwomen (type logout).

Create a local folder 'course' in virtual machine

Create a new file called ex2.sh in your home folder

Copy this file to your recently created folder

Remove the file from your home folder

## Exercise (2)

Disconnect from ironwomen (type logout).

Create a local folder 'course' in virtual machine

```
msuser@ironwomen:~$ logout  
Connection to ironwomen closed.  
fco@vpn:~$ mkdir course
```

Create a new file called ex2.sh in your home folder `fco@fco-VirtualBox:~$ touch ex2.sh`

Copy this file to your recently created folder `fco@fco-VirtualBox:~$ cp ex2.sh course/`

Remove the file from your home folder `fco@fco-VirtualBox:~$ rm ex2.sh`

## Exercise (2)

Disconnect from ironwomen (type logout).

Create a local folder 'course' in virtual machine

```
msuser@ironwomen:~$ logout  
Connection to ironwomen closed.  
fco@vpn:~$ mkdir course
```

Create a new file called ex2.sh in your home folder `fco@fco-VirtualBox:~$ touch ex2.sh`

Copy this file to your recently created folder

```
fco@fco-VirtualBox:~$ cp ex2.sh course/
```

Remove the file from your home folder

```
fco@fco-VirtualBox:~$ rm ex2.sh
```

```
fco@fco-VirtualBox:~$ mv ex2.sh course/
```



## Linux commands (3). Working with files

- **head** -n [number\_of\_lines] [file] → Show first [number\_of\_lines] of the [file]
- **tail** -n [number\_of\_lines] [file] → Show last [number\_of\_lines] of the [file]
- **cat** [file] → Print all the file
- **less** [file] → Print all the file
- **grep** [word] [file] → Find [word] in [file]
- **cut** -f [columns] [file] → Show only [columns] of [file]
- **sort** -k [column] [file] → Sort the lines of a file by [column]
- **diff** [file1] [file2] → Compare [file1] and [file2] line by line
- **uniq** [file] → Report or omit repeated lines in file
- **tr** → Translate or delete characters

## Examples (2)

### cat & less

```
fco@fco-VirtualBox:~$ less example.txt
fco@fco-VirtualBox:~$ cat example.txt
Hi humans!
This is just a text example
With a lot of different lines
I'm planning to have, at least 20
But I think I'm not going to have enough ideas
I will try
No, sure I haven't
How many lines do we have? 8
Bullshit, not enough. We need at least 10 more
So...
What I'm going to write?
Idea! just count last lines and that's it
Line 13
Line 14
Line 15
Line 16
Line 17
Almost done. This is 18
Second last: 19
And line 20!! Got it!
Congratulations!
fco@fco-VirtualBox:~$
```

```
Hi humans!
This is just a text example
With a lot of different lines
I'm planning to have, at least 20
But I think I'm not going to have enough ideas
I will try
No, sure I haven't
How many lines do we have? 8
Bullshit, not enough. We need at least 10 more
So...
What I'm going to write?
Idea! just count last lines and that's it
Line 13
Line 14
Line 15
Line 16
Line 17
Almost done. This is 18
Second last: 19
And line 20!! Got it!
Congratulations!
~
~
(END)
```

# Examples

## head & tail

```
fco@fco-VirtualBox:~$ head -n 5 example.txt
Hi humans!
This is just a text example
With a lot of different lines
I'm planning to have, at least 20
But I think I'm not going to have enough ideas
fco@fco-VirtualBox:~$
```

```
fco@fco-VirtualBox:~$ tail -n 8 example.txt
Line 14
Line 15
Line 16
Line 17
Almost done. This is 18
Second last: 19
And line 20!! Got it!
Congratulations!
fco@fco-VirtualBox:~$
```



## Examples (3)

### diff

```
fco@fco-VirtualBox:~$ cp example.txt example2.txt
fco@fco-VirtualBox:~$ echo "new line attached" >> example2.txt
fco@fco-VirtualBox:~$ tail example2.txt
Line 13
Line 14
Line 15
Line 16
Line 17
Almost done. This is 18
Second last: 19
And line 20!! Got it!
Congratulations!
new line attached
fco@fco-VirtualBox:~$ diff example.txt example2.txt
21a22
> new line attached
fco@fco-VirtualBox:~$ diff example2.txt example.txt
22d21
< new line attached
fco@fco-VirtualBox:~$
```

## Examples (4)

### sort, uniq & cut

```
fco@fco-VirtualBox:~$ cat example2.txt
cell1-1 cell1-2
cell2-1 cell2-2
cell3-1 cell3-2
cell3-1 cell3-2
cell3-1 cell3-2
cell4-1 cell4-2
fco@fco-VirtualBox:~$ cut -f 1 example2.txt
cell1-1
cell2-1
cell3-1
cell3-1
cell3-1
cell4-1
fco@fco-VirtualBox:~$ uniq example2.txt
cell1-1 cell1-2
cell2-1 cell2-2
cell3-1 cell3-2
cell4-1 cell4-2
fco@fco-VirtualBox:~$
```

```
fco@fco-VirtualBox:~$ sort example.txt
Almost done. This is 18
And line 20!! Got it!
Bullshit, not enough. We need at least 10 more
But I think I'm not going to have enough ideas
Congratulations!
Hi humans!
How many lines do we have? 8
Idea! just count last lines and that's it
I'm planning to have, at least 20
I will try
Line 13
Line 14
Line 15
Line 16
Line 17
No, sure I haven't
Second last: 19
So...
This is just a text example
What I'm going to write?
With a lot of different lines
fco@fco-VirtualBox:~$
```

## Examples (5)

tr

```
fco@fco-VirtualBox:~$ cat example2.txt
cell1-1 cell1-2
cell2-1 cell2-2
cell3-1 cell3-2
cell1-1 cell1-2
cell5-1 cell5-2
cell4-1 cell4-2
fco@fco-VirtualBox:~$ tr '\t' ',' < example2.txt
cell1-1,cell1-2
cell2-1,cell2-2
cell3-1,cell3-2
cell1-1,cell1-2
cell5-1,cell5-2
cell4-1,cell4-2
fco@fco-VirtualBox:~$
```

## Exercise (3)

Copy exercise.tsv

(msuser@ironwomen:/home/labs/mslab/msuser/fco/exercise.tsv) in course folder

Print last 10 lines of the file

Print 6th column of the file

Convert all the ':' to '-' in the file

## Exercise (3)

Copy exercise.tsv

(msuser@ironwomen:/home/labs/mslab/msuser/fco/exercise.tsv) in course folder

```
fco@vpn:~$ cd course
fco@vpn:~/course$ scp msuser@ironwomen:/home/labs/mslab/msuser/fco/exercise.tsv
.
msuser@ironwomen's password:
exercise.tsv                               100%   63KB   62.6KB/s   00:00
fco@vpn:~/course$ ls
exercise.tsv
fco@vpn:~/course$
```

## Exercise (3)

Print last 10 lines of the file

```
ffuster2@figaro:~/course$ tail -n 10 exercise.tsv
chr3 128199429 T 649 568 T:555:2
chr3 128199662 G 797 786 G:422:2
chr3 128204951 C 562 540 C:261:2
chr3 128205860 G 302 297 G:0:0:0
chr3 128206618 C 251 240 C:123:2
chr3 128206710 T 173 167 T:0:0:0
chr3 136056728 G 723 719 G:697:2
chr3 136056861 T 775 771 T:1:1:6
chr3 136191272 C 447 434 C:430:2
chr3 168801495 C 743 734 C:1:1:5
```

Print 6th column of the file

```
ffuster2@figaro:~/course$ cut -f 6 exercise.tsv
base:reads:strands:avg_qual:map_qual:plus_reads:minus_reads
T:451:2:42:1:308:143:183
T:635:2:44:1:409:226:0
G:650:2:45:1:472:178:13
A:307:2:37:1:154:153:0
A:0:0:0:0:0:0:0
T:567:2:36:1:518:49:0
A:545:2:32:1:509:36:2
```

Convert all the ':' to '-' in the file

```
ffuster2@figaro:~/course$ tr ':' '-' < exercise.tsv
chrom position ref_base depth q20_depth base-reads-
chr1 35649754 T 647 634 T-451-2-42-1-308-143-183
-3-AAA-1-1-13-1-1-0 INS-1-A-54-2-41-1-32-22 INS-2-AA-7-2-32-1-5-2
chr1 35649768 T 669 645 T-635-2-44-1-409-226-0 T
chr1 35649862 G 692 663 G-650-2-45-1-472-178-13 GA
chr1 35658369 A 347 308 A-307-2-37-1-154-153-0 A
```

# Special characters

'>' [file] → Store the output in [file]

'>>' [file] → Append output to [file]

2> [file] → Store error output to [file]

2>> [file] → Append output error to [file]

'<' [file] → Use [file] as input

',' → Separator between commands

'|' → Send the output to other program (pipe)

'\*' → All characters

'.' → Current folder

'..' → Parent folder

# Examples

```
fco@fco-VirtualBox:~$ cat example2.txt
cell1-1 cell1-2
cell2-1 cell2-2
cell3-1 cell3-2
cell1-1 cell1-2
cell5-1 cell5-2
cell4-1 cell4-2
fco@fco-VirtualBox:~$ tr '\t' ',' < example2.txt
cell1-1,cell1-2
cell2-1,cell2-2
cell3-1,cell3-2
cell1-1,cell1-2
cell5-1,cell5-2
cell4-1,cell4-2
fco@fco-VirtualBox:~$ tr '\t' ',' < example2.txt > example3.txt
fco@fco-VirtualBox:~$ cat example3.txt
cell1-1,cell1-2
cell2-1,cell2-2
cell3-1,cell3-2
cell1-1,cell1-2
cell5-1,cell5-2
cell4-1,cell4-2
fco@fco-VirtualBox:~$
```



# Examples

```
fco@fco-VirtualBox:~$ cat example2.txt
cell1-1 cell1-2
cell2-1 cell2-2
cell3-1 cell3-2
cell1-1 cell1-2
cell5-1 cell5-2
cell4-1 cell4-2
fco@fco-VirtualBox:~$ cat example2.txt >> example3.txt
fco@fco-VirtualBox:~$ cat example3.txt
cell1-1,cell1-2
cell2-1,cell2-2
cell3-1,cell3-2
cell1-1,cell1-2
cell5-1,cell5-2
cell4-1,cell4-2
cell1-1 cell1-2
cell2-1 cell2-2
cell3-1 cell3-2
cell1-1 cell1-2
cell5-1 cell5-2
cell4-1 cell4-2
fco@fco-VirtualBox:~$ █
```

# Examples

```
fco@fco-VirtualBox:~/course$ ls
ex2.sh      example3.txt  file_name_new.txt
example2.txt example.txt   not_example.txt
fco@fco-VirtualBox:~/course$ ls example*
example2.txt example3.txt example.txt
fco@fco-VirtualBox:~/course$ ls *example*
example2.txt example3.txt example.txt not_example.txt
fco@fco-VirtualBox:~/course$
```

# Examples

```
fco@fco-VirtualBox:~/course$ grep "!" example.txt
Hi humans!
Idea! just count last lines and that's it
And line 20!! Got it!
Congratulations!
fco@fco-VirtualBox:~/course$ grep "!" example.txt | tr "!" "?"
Hi humans?
Idea? just count last lines and that's it
And line 20?? Got it?
Congratulations?
fco@fco-VirtualBox:~/course$ grep "!" example.txt | tr "!" "?" | sort
And line 20?? Got it?
Congratulations?
Hi humans?
Idea? just count last lines and that's it
fco@fco-VirtualBox:~/course$ grep "!" example.txt | tr "!" "?" | sort | tr "?" "."
And line 20.. Got it.
Congratulations.
Hi humans.
Idea. just count last lines and that's it
fco@fco-VirtualBox:~/course$ █
```

## Exercise (4)

Print 1st column of exercise.tsv and store the output in a new file called exercise2.tsv

Add to exercise2.tsv, the first 5 lines of exercise.tsv

Find ':' in exercise2.tsv. Convert ':' to ','. Store the output in new file called exercise3.tsv

# Exercise (4)

Print 1st column of exercise.tsv and store the output in a new file called exercise2.tsv

```
fco@fco-VirtualBox:~/course$ cut -f 1 exercise.tsv > exercise2.tsv
```

Add to exercise2.tsv, the first 5 lines of exercise.tsv

```
fco@fco-VirtualBox:~/course$ head -n 5 exercise.tsv >> exercise2.tsv
fco@fco-VirtualBox:~/course$ tail -n 10 exercise2.tsv
chr3
chr3
chr3
chr3
chr3
chr3
chrom    position    ref_base    depth    q20_depth    base:reads:strands:avg_qual:map_qual:plus_read
chr1     35649754    T           647      634          T:451:2:42:1:308:143:183    TA    0    0    0
-3-AAA:1:1:13:1:1:0    INS-1-A:54:2:41:1:32:22    INS-2-AA:7:2:32:1:5:2
chr1     35649768    T           669      645          T:635:2:44:1:409:226:0    T    635    2    44    1
chr1     35649862    G           692      663          G:650:2:45:1:472:178:13    GA    0    0    0    0
chr1     35658369    A           347      308          A:307:2:37:1:154:153:0    A    307    2    37    1
fco@fco-VirtualBox:~/course$
```

Find ':' in exercise2.tsv. Convert ':' to ','. Store the output in new file called

exercise3.tsv

```
fco@fco-VirtualBox:~/course$ grep ':' exercise2.tsv | tr ':' ',' > exercise3.tsv
fco@fco-VirtualBox:~/course$ cat exercise3.tsv
chrom    position    ref_base    depth    q20_depth    base:reads:strands:avg_qual:map_qual:plus_reads,minus_reads
chr1     35649754    T           647      634          T,451,2,42,1,308,143,183    TA    0    0    0    0    0    0    A,1,1,34,1,0,1    DEL-1-A,105,2,44,1,80,25    DEL-2-AA,19,2,43,1,11,8    DEL
-3-AAA,1,1,13,1,1,0    INS-1-A,54,2,41,1,32,22    INS-2-AA,7,2,32,1,5,2
chr1     35649768    T           669      645          T,635,2,44,1,409,226,0    T    635    2    44    1    409    226    0    A,8,2,46,1,6,2
chr1     35649862    G           692      663          G,650,2,45,1,472,178,13    GA    0    0    0    0    0    0    DEL-1-A,7,2,38,1,3,4    INS-1-A,7,2,46,1,3,4
chr1     35658369    A           347      308          A,307,2,37,1,154,153,0    A    307    2    37    1    154    153    0
fco@fco-VirtualBox:~/course$
```

## Linux commands (4). Other useful commands

- **man** [command] → Get the manual of [command]
- **wc** [file] → Print newline, word, and byte counts of [file]
- **history** → Print all the commands used
- **clear** → “Cleans” the terminal
- **echo** [“Some words”] → Prints [“some words”] in the terminal

## Exercise (5)

What is the parameter in 'wc' command to print only the number of lines that file has?

Print all the 'cd' command you have used during the course

## Exercise (5)

What is the parameter in 'wc' command to print only the number of lines that file has?

```
ffuster2@figaro:~/course$ man wc
ffuster2@figaro:~/course$
ffuster2@figaro:~/course$ wc -l exercise.tsv
40 exercise.tsv
```

Print all the 'cd' command you have used during the course

```
ffuster2@figaro:~/course$ history | grep cd
```



# Merging all

We can combine different instructions if needed

```
#!/bin/bash  
  
#Do all exercises at once  
cd /home/fco #Go to home folder  
mkdir course #Create new folder  
cd course #Go to folder  
touch ex2.sh #Create new blank file  
cp /media/ffuster/USB/exercise.tsv . #Copy example exercise from the USB.  
tr ':' '-' < exercise.tsv > exercise2.tsv #Change ':' to '-' in the file and store the output in a new file  
sort exercise2.tsv > exercise3.tsv #Sort the file and store it in a new file
```

# Merging all

We can combine different instructions if needed

```
#!/bin/bash  
  
#Do all exercises at once  
cd /home/fco #Go to home folder  
mkdir course #Create new folder  
cd course #Go to folder  
touch ex2.sh #Create new blank file  
cp /media/ffuster/USB/exercise.tsv . #Copy example exercise from the USB.  
tr ':' '-' < exercise.tsv > exercise2.tsv #Change ':' to '-' in the file and store the output in a new file  
sort exercise2.tsv > exercise3.tsv #Sort the file and store it in a new file
```

My first bash script!!

# Bash scripts. Variables

Pieces of code that can get whatever value

```
fco@fco-VirtualBox:~/course$ cat script1.sh
#!/bin/bash

variable="My first variable"

echo $variable
fco@fco-VirtualBox:~/course$ ./script1.sh
My first variable
fco@fco-VirtualBox:~/course$
```

Special variables:

- \$1 - \$9 → Get the value of first - ninth option
- ...

# Bash Scripting. If statement

Piece of code that will be executed if a condition is true

```
#!/bin/bash  
  
#My first if statement  
if [ condition ]  
then  
    [Commands that will be executed if condition is true]  
else  
    [Commands that will be executed if the condition is false]  
fi
```

# Bash Scripting. If statement

Piece of code that will be executed if a condition is true. Examples

```
#!/bin/bash  
  
if [ -f file.txt ]  
then  
    [Commands will be executed if file exists]  
fi  
  
if [ -d folder ]  
then  
    [Commands will be executed if folder exists]  
fi
```

**More examples** → <https://ryanstutorials.net/bash-scripting-tutorial/bash-if-statements.php>

# Bash scripting. For loop

Piece of code that will be executed several times

```
#!/bin/bash  
  
#My first for loop  
for [times]  
do  
    [Commands that will be executed several times]  
done
```

# Bash scripting. For loop

Piece of code that will be executed several times. Examples

```
#!/bin/bash

for variable in *.txt
do
    [Commands will be executed in all txt files that are in the folder]
done

for variable in file1 file2 file3 file4
do
    [Commands will be executed only in file1, file2, file3, and file4]
done
```

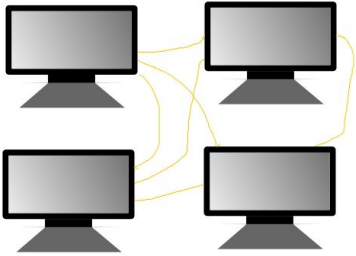
**More examples** → <https://www.poftut.com/linux-bash-loop-files/>

# Cluster

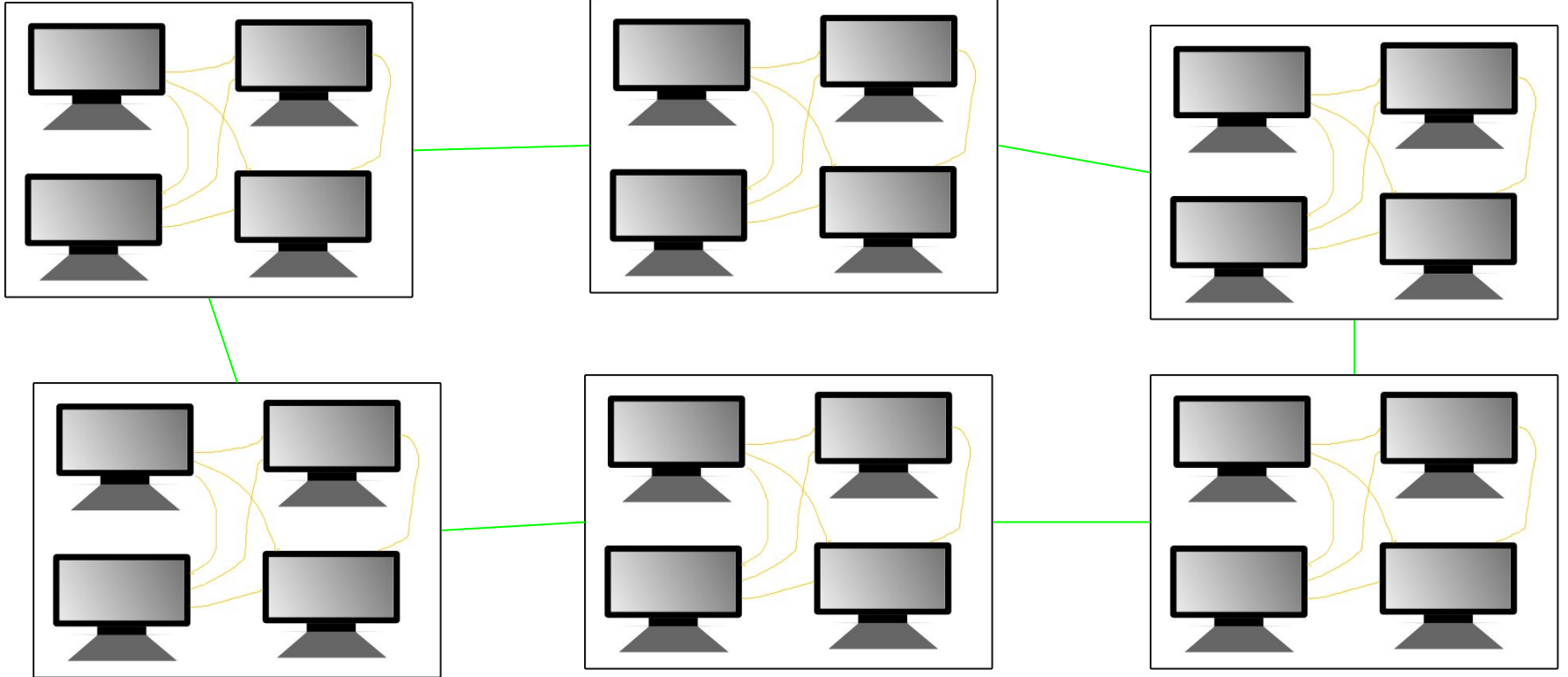




# Cluster



# Cluster



# Queuing system

Orders the jobs sent to cluster according to priorities

- SGE
- SLURM
- ...

# SGE useful commands

**qsub** [bash\_script] → Sends [bash\_script] to cluster

**qstat** → Check status of your jobs

**qdel** → Delete a job from cluster

## Exercise (6)

Connect to ironwomen

Copy cluster\_template.sh to “your\_name” folder

Modify cluster\_template.sh, with your name [use nano]

Send the job to cluster

Check the cluster status

# Additional exercises (1)

1. Print last 10 lines of exercise.tsv, but excluding last one
2. Print 2nd column of exercise.tsv, sort the output and store it in a new file called exercise2.tsv
3. Find the files in course folder that starts with 'ex' and has '2' in the name
4. Print 6th column of exercise.tsv, change ':' to ',' and sort the output. Append the output to exercise2.tsv
5. Count the number lines that have chr1 in file1.tsv
6. Check if there are chromosomal regions repeated in file1.tsv

## Additional exercises (2)

1. How many genes are in file1.tsv? and in file2.tsv?
2. Which file has more regions: file1.tsv or file2.tsv?
3. Which file is bigger (occupies more bytes): file1.tsv or file2.tsv?
4. Find regions regions in common in both files
5. Find how many times NF1 gene appears in file2.tsv

## Additional exercises (3)

1. File2.tsv is unsorted. Sort it by chromosome and chromosomal position
2. File1 is tab-separated file, but customer says that he/she needs comma-separated file. Can you do it using bash?
3. There is a typo in a gene in file2.tsv. Are you able to find it? (Tip: use sort + uniq)