Introduction to the Command-line Environment

Gustavo Adolfo Silva-Arias Ph.D

Technische Universität München

Bogotá, 23 de Agosto 2021

What is GNU/Linux?

Linux is the best-known and most-used “unix-like” **open-source** operating system

software that manages all the hardware resources associated with a computer

Linus Torvalds, inventor of Linux
1991, University of Helsinki, Finland

OS components

- **Bootloader**: manages the boot process
- **Kernel** (This is Linux): The core of the system and manages the CPU, memory, and peripheral devices. But also the set of programs, tools, and services to provide a fully functional operating system.
- **Daemons**: background services
- **The Shell** (or command line): software that allows you to control the computer via commands typed into a text interface
- **Graphical Server**: The sub-system that displays the graphics on the monitor
- **Desktop Environment**: This is the part that the users interact with
- **Applications**

Why to use Linux?

Free and open access!!!

Linux is distributed under an open-source license.
4 key philosophies:

- The freedom to run the program, for any purpose.
- The freedom to study how the program works and change it to make it do what you wish.
- The freedom to redistribute copies so you can help your neighbor.
- The freedom to distribute copies of your modified versions to others.

Why to use Linux
(or more specifically the **command-line**)?

Flexibility

- Adjust processes for specific purposes.

Practicality

- Keep a log of all user commands, easy to recreate previous steps.
- Exact analysis reproduction for use on different datasets, settings or external verification.

Increase computer power for complex and large-scale analysis

- Graphical User Interface (GUIs) are memory consuming, not friendly to use on clusters or remote machines.

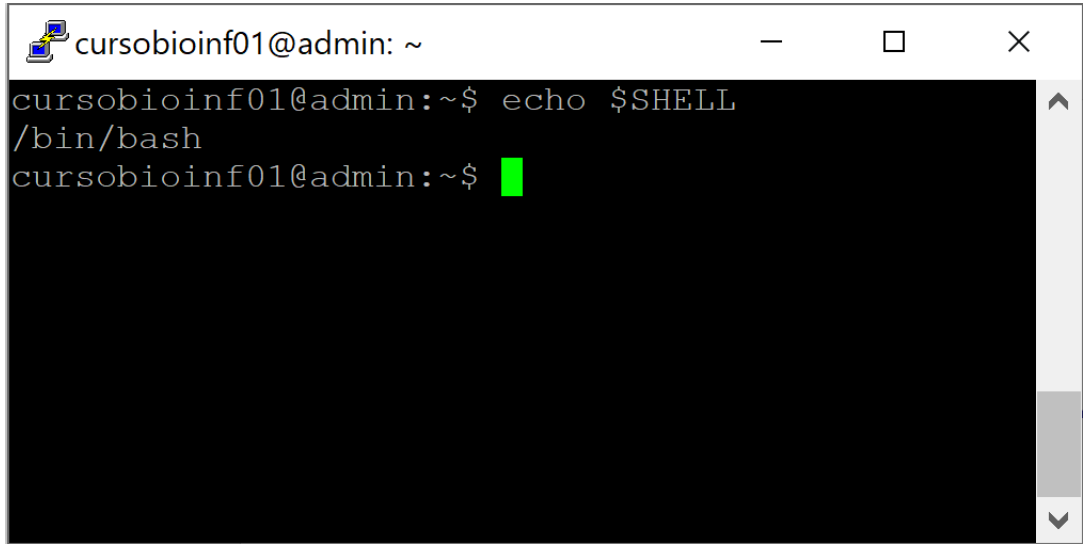
Broad uses

- GUIs are labor-intensive to make and usually work on only the specific OS they were developed.

A tool for **interacting with a computer**
through typed instructions at the command line

The SHELL
Terminal

The SHELL



A command-line environment

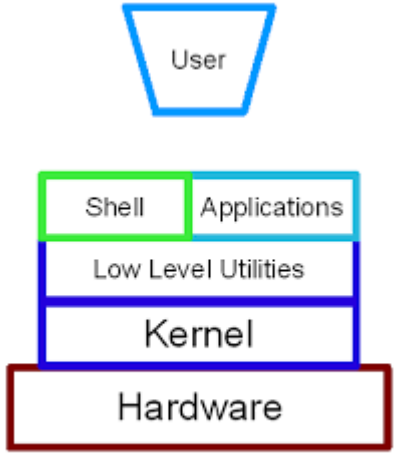
The information in the terminal is displayed by a program called a shell.

Many shell programs – default: bash

Check your shell program with the command: echo $SHELL

Should be **/bin/bash**

The SHELL

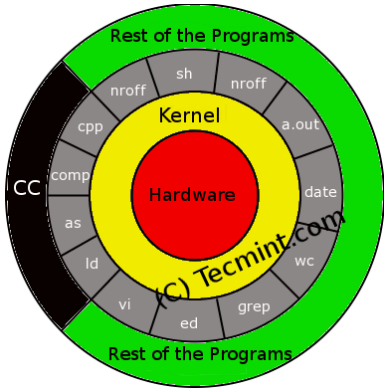


Shell is a program for **user interaction** that understands commands in English (mostly).

It is one (out of many) command language interpreter that executes commands (from the standard input device or a file)

We will use the Linux Shell called BASH.

You can add the commands one by one or make a **script**



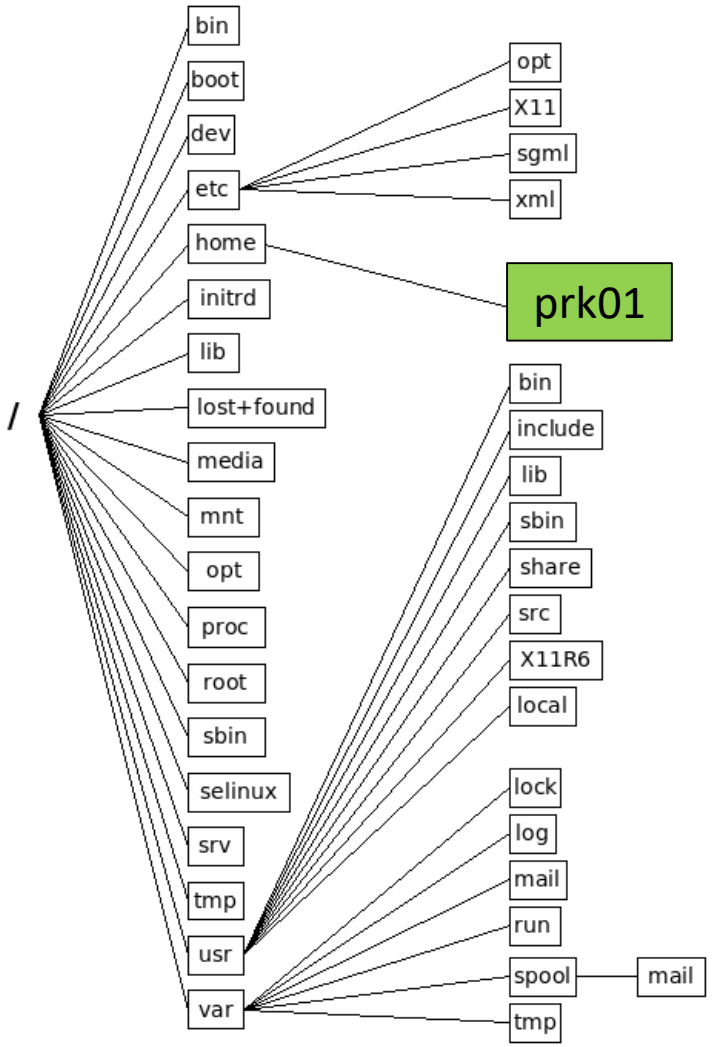
The filesystem

directory=folder

root directory: where all files on the system reside.
denoted by a “/”

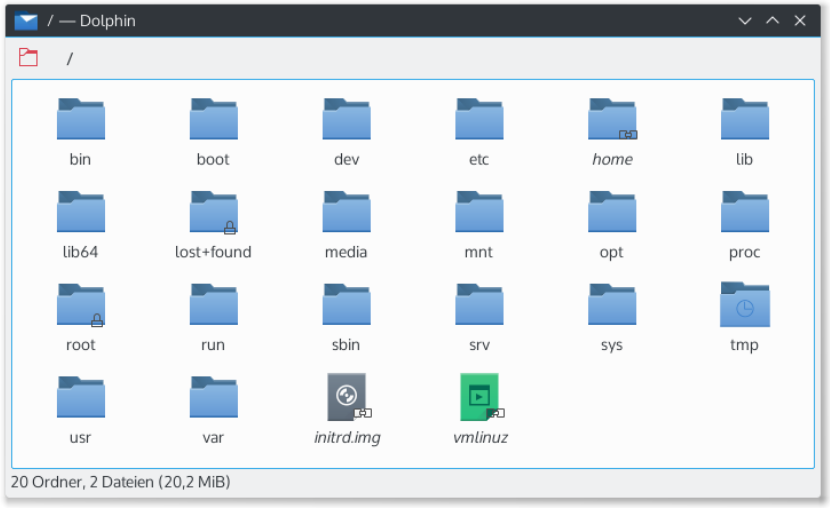
UNIX based systems have a single root directory.

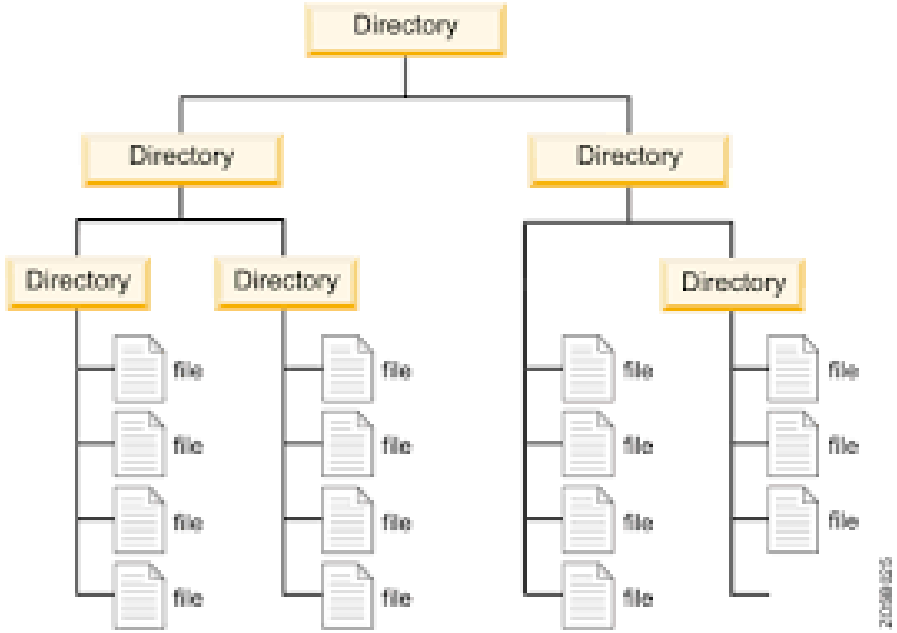
hier - description of the filesystem
hierarchy



The filesystem

Hierarchical arrangement





The path

Description of where something is located in the filesystem.

/data/home/users/g.silvaarias/NGS_course

Indicates a folder

absolute path – give the location of a directory from at the top (root) of the directory tree.

/data/home/users/g.silvaarias/NGS_course/scripts

relative path - give the location of a directory relative to the current working directory.

If I am in the g.silvaarias (or $HOME) folder it would be:

NGS_course/scripts

without “/” in the beginning

Exploring the shell

Where am I?

current directory – working directory

pwd – "print working directory”

Returns the **absolute path of the** directory you are currently located in.

What are here?

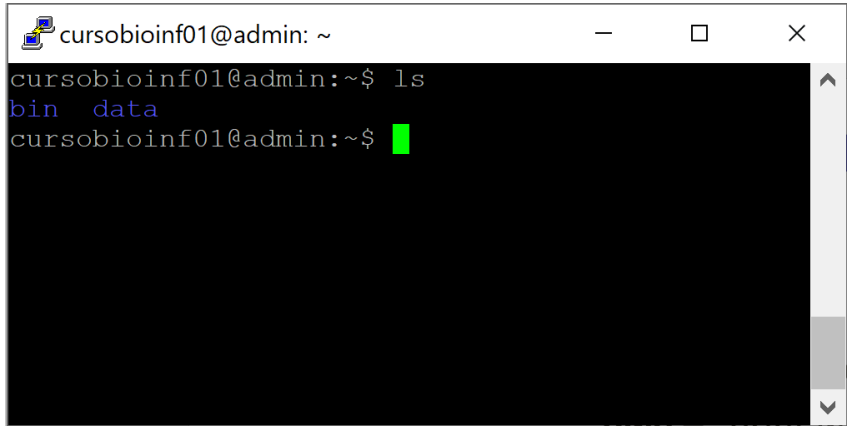
ls – “list”

Tell you all directories and files in your current directory.

Optional flags:

ls – l: output information in long-form

ls –a: lists all files, including hidden files



Exploring the shell

How to move around?

cd – “change directory”

move to any directory

cd path (absolute or relative)

./ (or just .) **means current directory**

../ means parent directory (upward in the hierarchy or ‘backwards’)

“/” before the path name will look for the file in the root directory

CASE SENSITIVE!!!

Exploring the shell

Go to the top-level directory (root):

```
cd /
```

List the content of the folder:

```
ls
```

All files, devices, directories, or applications are located under this directory.

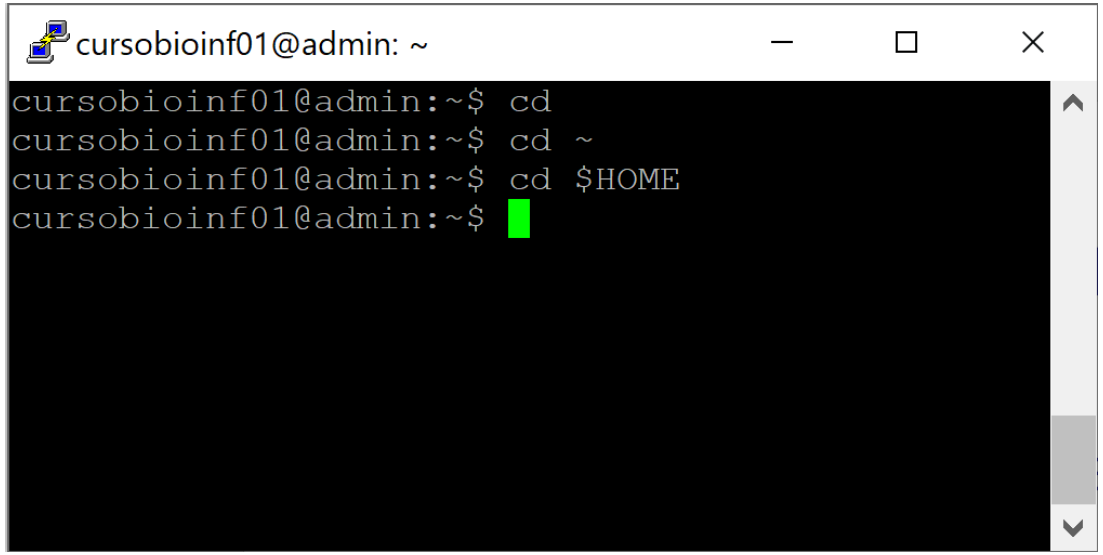
```
bin     data  home   localscratch  mnt   root  srv  usr
boot    dev   lib    lost+found    opt   run   sys  var
cluster etc   lib64  media   proc  sbin  tmp
```

Exploring the shell

Go back to your HOME directory:

Tip:

No matter where you are on the system you can type just ***cd, cd $HOME*** **or** ***cd ~*** ("tilde") for referring to your home directory.



Exploring the shell

Adding and removing directories

mkdir – make directory

Make a new directory in your home directory called NGScourse

mkdir NGScourse

Check if it is in your home directory

ls

Exploring the shell

Adding and removing directories

rmdir – remove directory

Remove the directory NGScourse

rmdir NGScourse

Check if it worked:

ls

Exploring the shell

Adding new files

touch – change file timestamps

(Update the access and modification times of each FILE to the current time)

Create the file new_test.txt

touch new_test.txt

Check if it worked:

ls

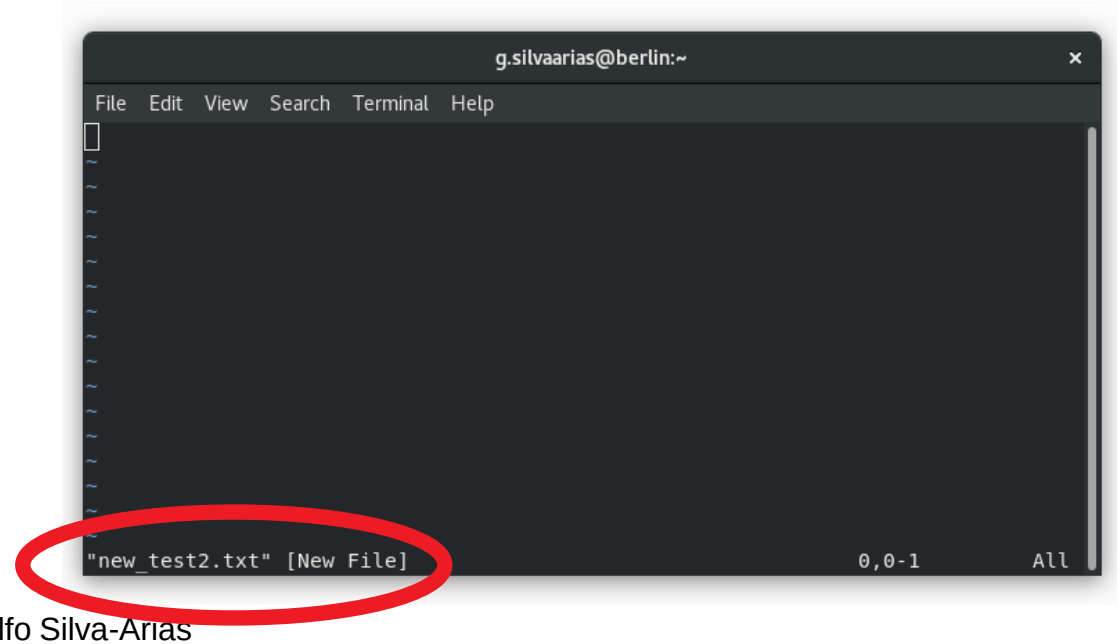
Exploring the shell

Adding new files

vim – Vi IMproved, a programmer's text editor

Create the file new_test2.txt

vim new_test2.txt



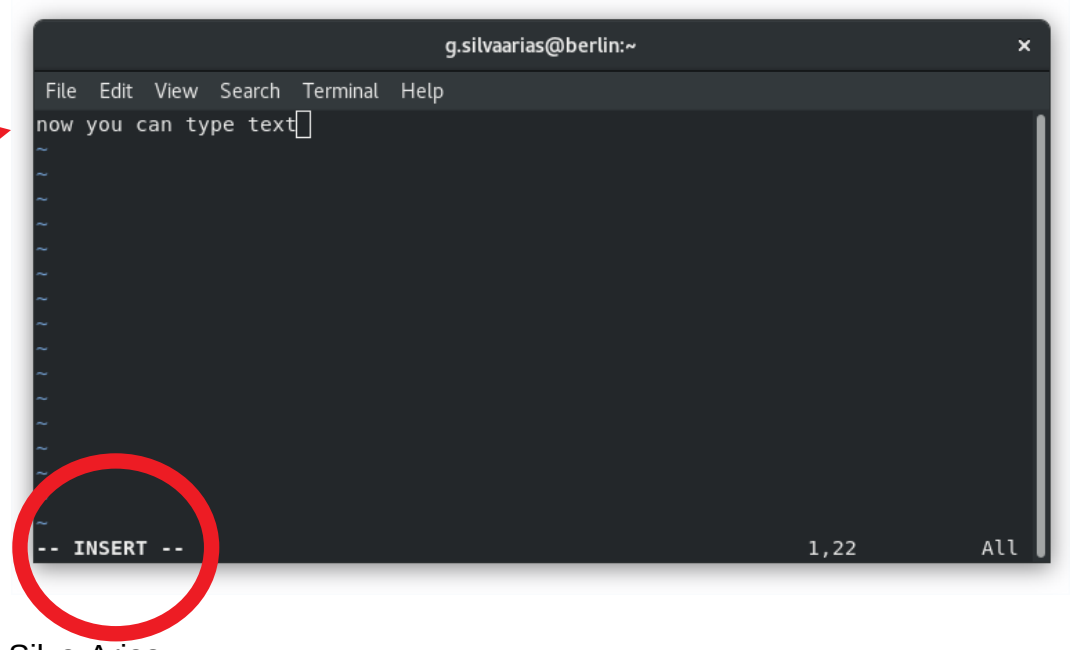
Exploring the shell

Adding new files

vim – Vi IMproved, a programmer's text editor

Change to editor mode

Press “i” (insert)



Exploring the shell

Adding new files

vim – Vi IMproved, a programmer's text editor

Exit the editor mode

Press “esc” key

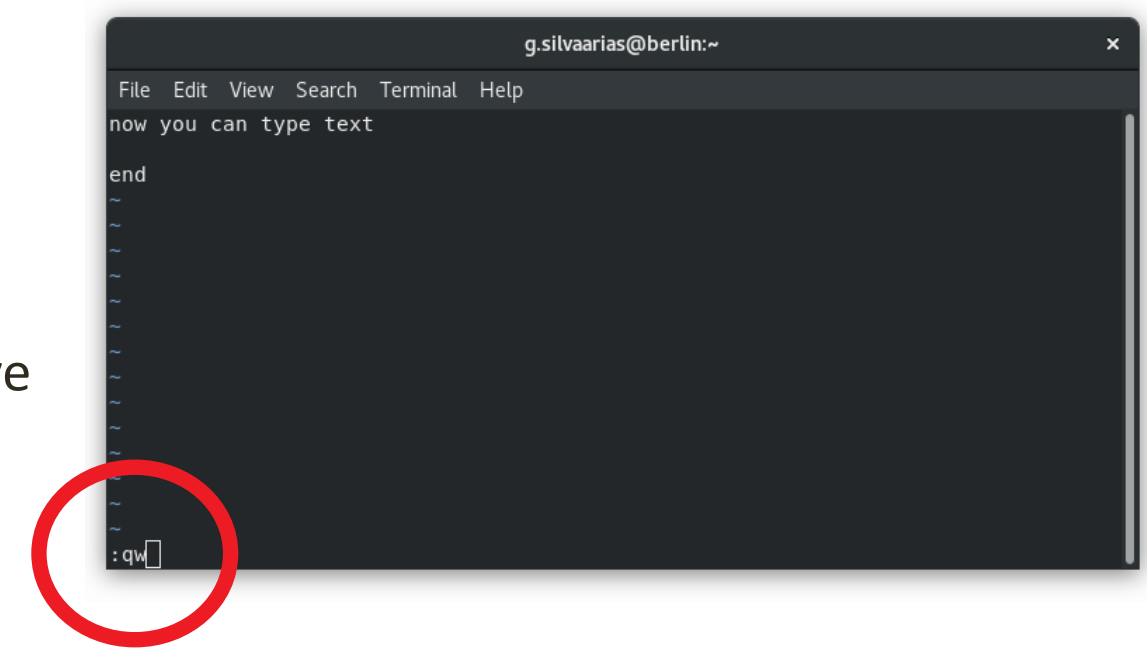
Type “:wq”

(w – white) or save

(q – quit)

Check if it worked:

ls



Exploring the shell

Naming folders and files

Reserved characters and words

It is not possible to create a file and directory entries with the same name in a single directory.

Avoid commands names (e.g. cat, ls ...)

Avoid space

Character	Name
/	slash
\	backslash
?	question mark
%	percent
*	asterisk *or* star
:	colon
|	vertical bar *or* pipe
"	quote
<	less than
>	greater than
.	period *or* dot
	space

https://en.wikipedia.org/wiki/File_name

Exploring the shell

Copying files

cp – copy

Copying files in the same directory

cp new_test2.txt new_test2_copy.txt

Exploring the shell

Copying files

cp – copy

Copying files in different directories keeping the original filename

mkdir bk

cp new_test2.txt bk/

Copying (recursively)a directory

cp -r /home/curso/data/example1 .

Check if it worked:

ls

ls example1/

Exploring the shell

Displaying and joining files

cat – concatenate files and print on the standard output

Using cat to check file content

Check the content of your file new_test2.txt

cat new_test2.txt

Exploring the shell

Displaying and joining files

cat – concatenate files and print on the standard output

Using cat with a redirect to join (or concatenate) files

Go to the **example1** folder and type:

cd example1

cat seq1.fasta (you will see a sequence file in fasta format)

Now create the file 'all_sequences.fasta' that contains all sequences in the folder: **cat *.fasta > all_sequences.fasta**

***** is a quantifier that matches all files with the specified characters

Check the new file

cat all_sequences.fasta

Exploring the shell

Try another command with cat

Match only files ending with 1.fasta

cat *1.fasta > few_sequences.fasta

Appending other files to sequences.fasta

cat *2.fasta >> few_sequences.fasta

Use >> to append at the end of the file

Check the content of your file few_sequences.fasta

cat few_sequences.fasta

Exploring the shell

Displaying a huge file

less – show contents of a file, page by page

Try with the file all_sequences.fasta you already created

less all_sequences.fasta

space – next page

b – back page

type **/seq86** – search for text “seq86”

q - quit

Exploring the shell

Moving files

mv – move (rename) files

'cut and paste'. Much faster than **cp** .

But be careful!!! Command interruption can lead to data loss

Rename the sequences.fasta file

mv example1/all_sequences.fasta my_seq.fas

Make a directory called old_data and move the fasta files into it.

mkdir old_data

mv example1/*.fasta old_data/

Check if it worked:

ls example1/

ls old_data/

Exploring the shell

Extracting specific rows from a file

grep – search for string in a file and show matching lines

Copy the folder **example2** from /home/curso/data

Check the file data_set.csv

ls, cat, less …

Let's make a file that only contains information about *Solanum pennellii*

grep "pennellii" data_set.csv

grep "pennellii" data_set.csv > S_pennellii_data.csv

How many *S. pennellii* records we have?

grep -c "pennellii" data_set.csv

Exclude all *S. pennellii* records

grep -v "pennellii" data_set.csv > data_set_wo_Spenn.csv

Exploring the shell

Extracting lines

head – shows the first lines of a file

head -n 5 data_set.csv

return the first 5 lines

tail – show the last lines of a file

tail -n 10 data_set.csv

cut – show columns of a file

cut –f1,3 data_set.csv

return columns 1 and 3 delimited by tabs (default)

cut -d " " -f 1,3 data_set.csv (delimited by space)

Exploring the shell

Shortcuts

Up and Down arrows: moves back and forward through your previous command history.

Right and Left arrows: to move along your command line

TAB key: auto-complete files, directories and command names

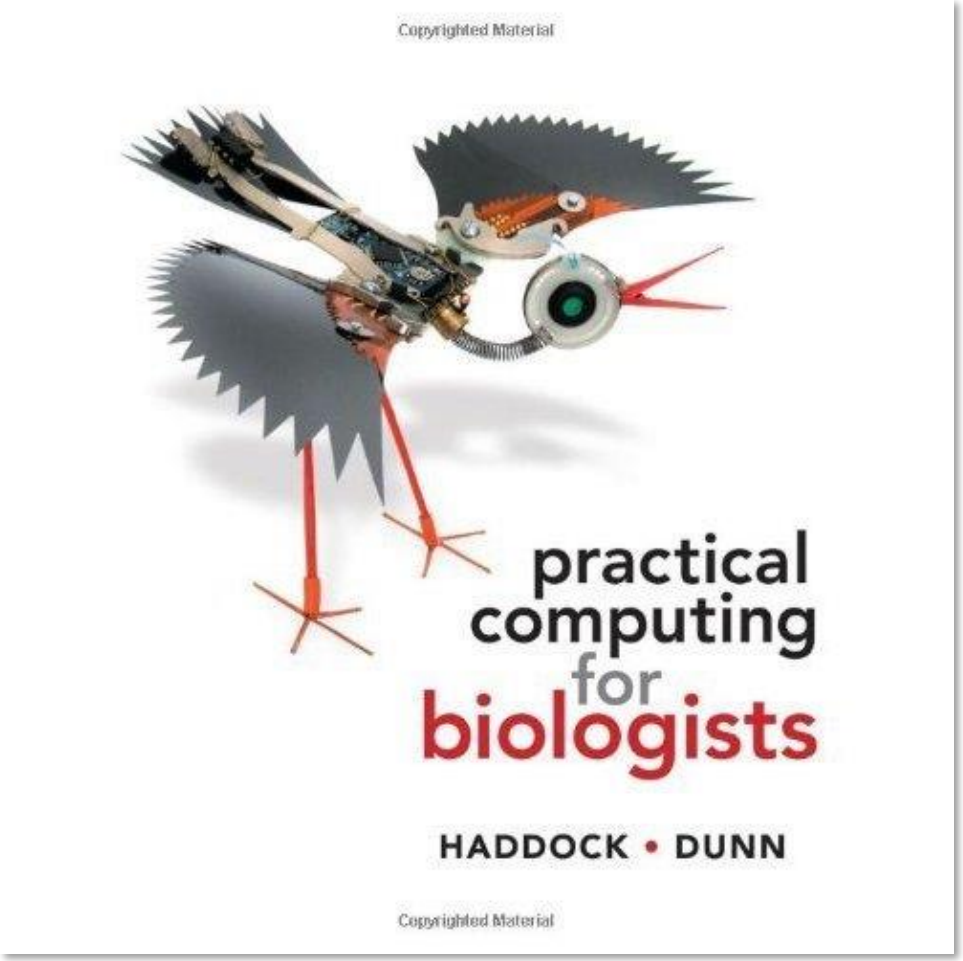
Ctrl + c – terminate the command

Ctrl + l – clear the screen

Ending your session: exit

man program_name – display information about the program (not in the frontend)

man grep



Useful tools:

textbook

http://practicalcomputing.org/

Images sources:
https://www.supinfo.com/articles/single/4323-history-of-linux
https://www.linux.com/what-is-linux
https://www.tecmint.com/understand-linux-shell-and-basic-shell-scripting-language-tips/
http://www.madcomputer.co.uk/tips-on-buying-a-new-computer
https://pixabay.com/illustrations/binary-1-0-computer-code-zero-1066983/
https://www.kullabs.com/classes/subjects/units/lessons/notes/note-detail/436
https://bigbangtheory.fandom.com/wiki/Sheldon_Lee_Cooper%27s_Laptop

Happy scripting

```
1111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111
1111111111111111111111111111111111111111111
1111000000000001111000000000000000011111
1111000000000001111000000000000000011111
1111111100111100111100011100011110001111
1111111100111100111100011100011110011111
1111111100111100111100011100011110011111
1111111100111100111100011100011110011111
1111111100111100111100011100011110011111
1111111100111100111100011100011110011111
1111111100111100111100011100011110011111
1111111100111100111100011100011110011111
1111111100111100111100011100011110011111
1111111100111100111100011100011110011111
1111111100111100111100011100011110011111
1111111100111100000000011100011110011111
1111111100111100000000011100011110011111
1111111100111100000000011100011110011111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
1111111111111111111111111111111111111111
```