Introduction to Linux

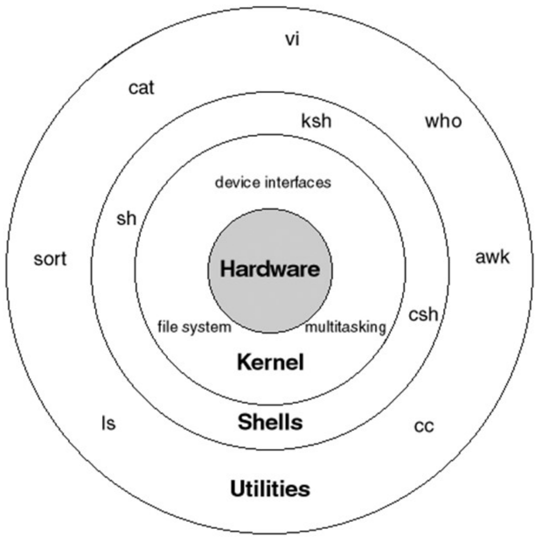
About Linux

- An **operating system (OS)** that manages a computer's resource – like MacOS and Windows.
- Derived from UNIX operating system (and so it is called **UNIX-like OS**).
- Developed by Linus Torvald, hence the name: Linux = Linus + Unix.
- A family of operating systems based on Linux as the center (**kernel**) is known as **Linux distribution or distro**.
  - —For example: Debian, Ubuntu, centOS, Fedora, Arch Linux, etc.
  - —Each distribution has their own specification – i.e. RedHat is more suitable to be run in server while Ubuntu is more suitable for desktop.

Linux comes in many flavors

redhat
MEPIS
turbolinux
LUNAR
EvilEntity
debian
Vine Linux
cAos/CentOS
MiniKazit
UTUTO
archlinux
m0n0wall
jamd
Knoppix STD
gentoo linux
DeLi Linux
Hiweed
amlug
slackware
yellow dog linux
Fedora
LPG
PLD
SLAX
COREL LINUX
Progeny
GEEXBOX
BIGLINUX
FREEDUC
Lycoris
EnGarde
Mandrakelinux
BeatrIX
Linspire
SuSE
中文征仲套件
YOPER
BearOps
ASPLINUX
kalango
Slackintosh
Frugalware
Foresight
Mint
PCLinuxOS
Haydar Linux
sabayon
ubuntu
JULEX
blag

The Linux system



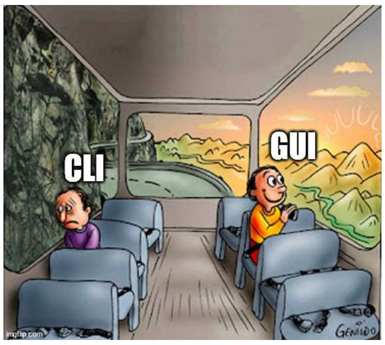
Layers of Linux

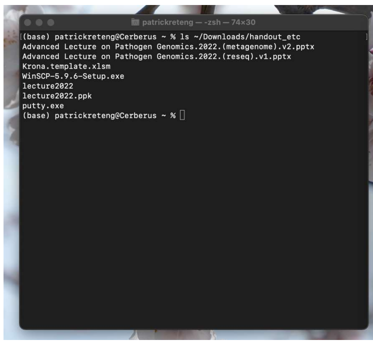
Linux and Bioinformatics

- Bioinformatics is the application of tools of computation and analysis to the capture and interpretation of biological data.
- Why Linux become the natural environment for bioinformatician?
  - —Open source – free.
  - —Linux gives freedom to control hardwares – i.e. memory allocation.
  - —Most developers working on Linux.

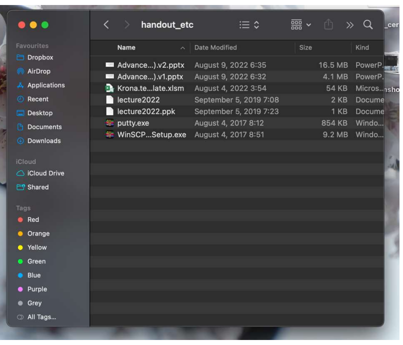
GUI or CLI?

- Command line interface (CLI) or graphical user interface (GUI).





```
(base) patrickreteng@Cerberus ~ % ls ~/Downloads/handout_etc
Advanced Lecture on Pathogen Genomics.2022.(metagenome).v2.pptx
Advanced Lecture on Pathogen Genomics.2022.(reseq).v1.pptx
Krona.template.xlsm
WinSCP-5.9.6-Setup.exe
lecture2022
lecture2022.ppk
putty.exe
(base) patrickreteng@Cerberus ~ % 
```



Name	Date Modified	Size	Kind
Advance...).v2.pptx	August 9, 2022 6:35	16.5 MB	PowerP…
Advance...).v1.pptx	August 9, 2022 6:32	4.1 MB	PowerP…
Krona.te...late.xlsm	August 4, 2022 3:54	54 KB	Micros…
lecture2022	September 5, 2019 7:08	2 KB	Docume
lecture2022.ppk	September 5, 2019 7:23	1 KB	Docume
putty.exe	August 4, 2017 8:12	854 KB	Windo...
WinSCP...Setup.exe	August 4, 2017 8:51	9.2 MB	Windo...

- Viewing files inside the folder requires knowledge of command and path.
- Viewing files inside the folder is as easy as opening the folder.

GUI or CLI?

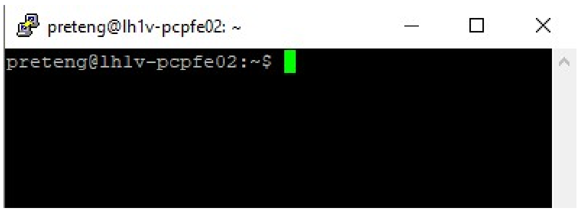
LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

- Why CLI?
  - —Repetition: easier to repeat task with CLI. More input – more reason to use CLI.
  - —Reproducibility: with good script/command/code documentation.
  - —Access to server or high performance computer: usually use CLI.
  - —Not many utilities for genome analysis is available in GUI – need to familiarize with CLI.

The shell

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

- In Linux OS, shell is a program that facilitates user-computer interaction.
  - —Most Linux systems use BASH (Bourne Again Shell).
  - —Other examples: C shell, sh.
- Communication is done by typing commands into the shell – the shell translates this command and pass the prompt to the kernel.
- The general format of the default command prompt is: [username@hostname cwd]$ (or #).



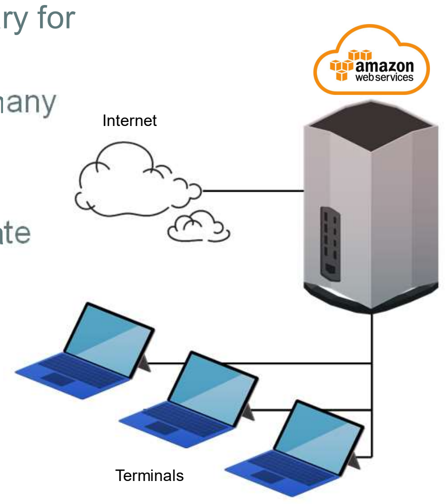
```
preteng@lh1v-pcpfe02:~$
```

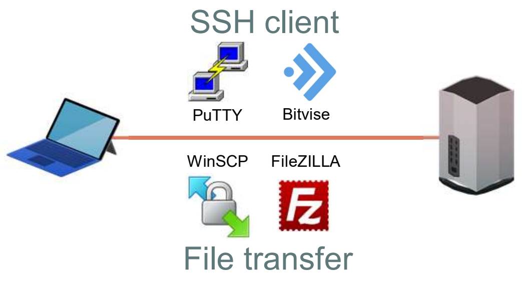
$ = regular user
# = root user

Accessing a server

LUXEMBOURG INSTITUTE OF HEALTH RESEARCH DEDICATED TO LIFE

- High performance computer (HPC) is often necessary for genome analysis.
  - —A cluster of computers that can be accessed by many users.
- Connection is established with SSH or secure shell.
  - —Secure connection is established with public/private key system.





Commands

LUXEMBOURG INSTITUTE OF HEALTH RESEARCH DEDICATED TO LIFE

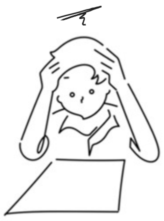
- The way to communicate with the computer.
- Structure or syntax:
  - —Starts with either a command or a program+command.
  - —Followed by options and/or arguments.

```
$ls      -l       /path/to/a/directory/
command  options  argument

$grep    -A1      chr1       ref.fasta
command  options  argument1  argument2

$samtools  view     -bS      input.bam
program    command  options  argument
```

Commands

- Computer is great at doing repetitive task.

Calculate log10 of 1 to 10,000



Errors and frustration

No complaints, less errors

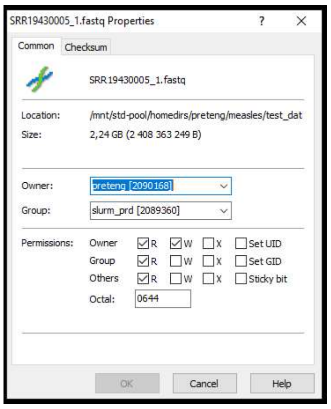
- Communicating with computer can be challenging.
- Be systematic and concise.

"Wear shoes"

- Locate shoes
- Locate socks
- Put left sock on
- Put right sock on
- Put right shoe on
- Put left shoe on

User and permission

- Linux OS is multiuser.
- A user will be given a full access to a directory – this is called home.
- All files and directory will be granted a certain level of permission.
- The level of permission can be freely manipulated - use `chmod` command.
  - Use `ls -l` to check file permission.
- There are special user: those with root access.
  - With great power comes great responsibility.



SRR19430005_1.fastq Properties

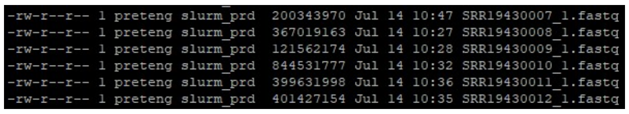
Common | Checksum

SRR19430005_1.fastq

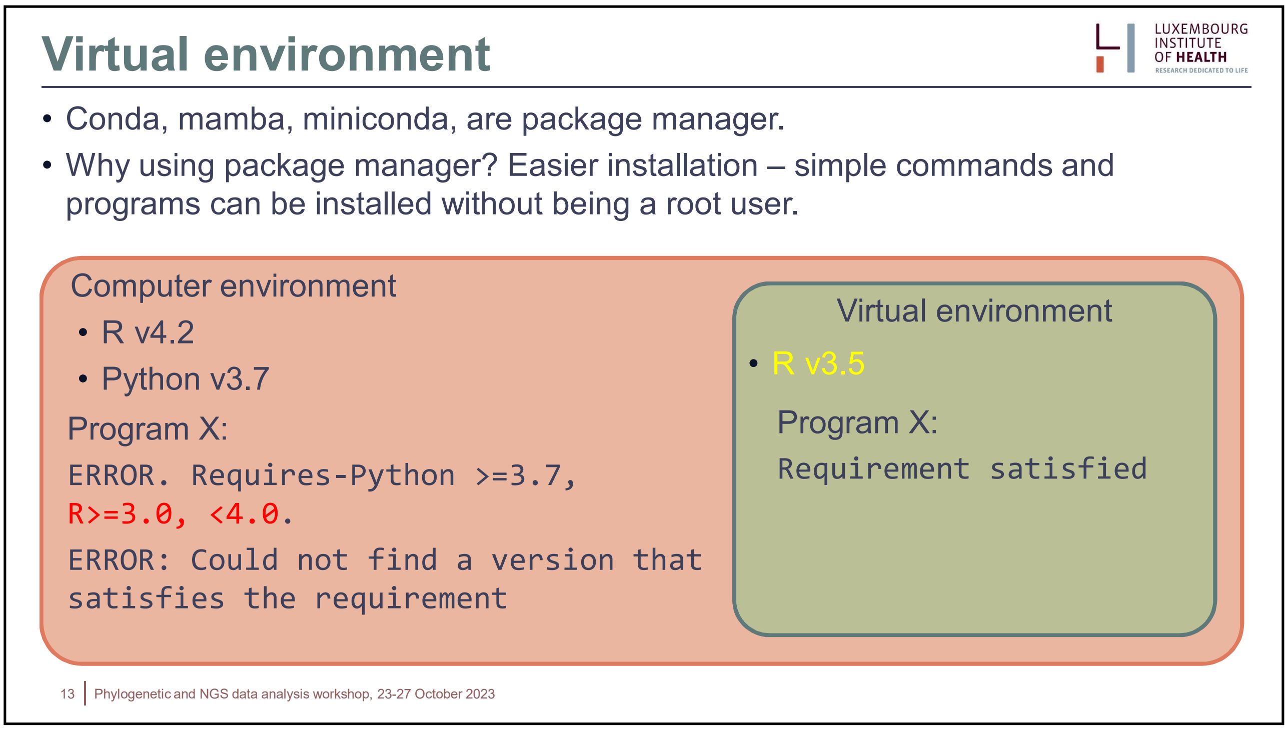
Location: /mnt/std-pool/homedirs/preteng/measles/test_dat
Size: 2,24 GB (2 408 363 249 B)

Owner: preteng [2090168]
Group: slurm_prd [2089360]

Permissions:
Owner [✓ R] [✓ W] ☐ X ☐ Set UID
Group [✓ R] ☐ W ☐ X ☐ Set GID
Others [✓ R] ☐ W ☐ X ☐ Sticky bit
Octal: 0644



```
-rw-r--r-- 1 preteng slurm_prd  200343970 Jul 14 10:47 SRR19430007_1.fastq
-rw-r--r-- 1 preteng slurm_prd  367019163 Jul 14 10:27 SRR19430008_1.fastq
-rw-r--r-- 1 preteng slurm_prd  121562174 Jul 14 10:28 SRR19430009_1.fastq
-rw-r--r-- 1 preteng slurm_prd  844531777 Jul 14 10:32 SRR19430010_1.fastq
-rw-r--r-- 1 preteng slurm_prd  399631998 Jul 14 10:36 SRR19430011_1.fastq
-rw-r--r-- 1 preteng slurm_prd  401427154 Jul 14 10:35 SRR19430012_1.fastq
```



Virtual environment

- Conda, mamba, miniconda, are package manager.
- Why using package manager? Easier installation – simple commands and programs can be installed without being a root user.

Computer environment
- R v4.2
- Python v3.7

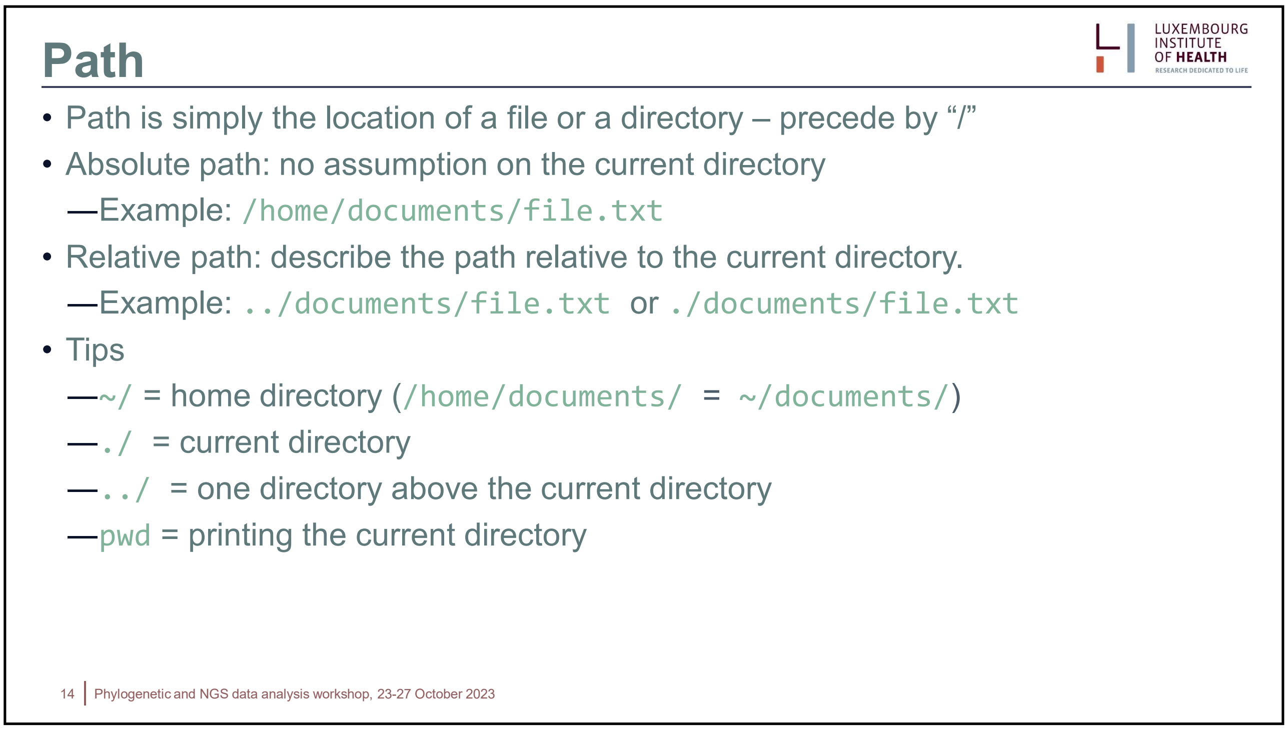
Program X:

```
ERROR. Requires-Python >=3.7,
R>=3.0, <4.0.
ERROR: Could not find a version that
satisfies the requirement
```

Virtual environment
- R v3.5

Program X:

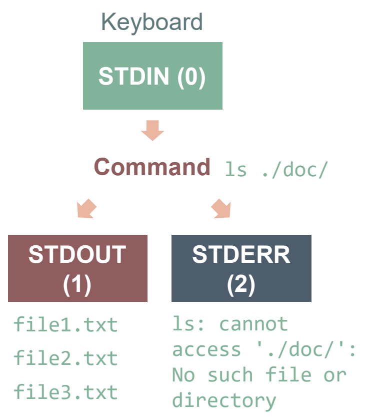
```
Requirement satisfied
```



Path

- Path is simply the location of a file or a directory – precede by “/”
- Absolute path: no assumption on the current directory
  - Example: `/home/documents/file.txt`
- Relative path: describe the path relative to the current directory.
  - Example: `../documents/file.txt` or `./documents/file.txt`
- Tips
  - `~/` = home directory (`/home/documents/ = ~/documents/`)
  - `./` = current directory
  - `../` = one directory above the current directory
  - `pwd` = printing the current directory

Redirection



```
ls ./doc/
```

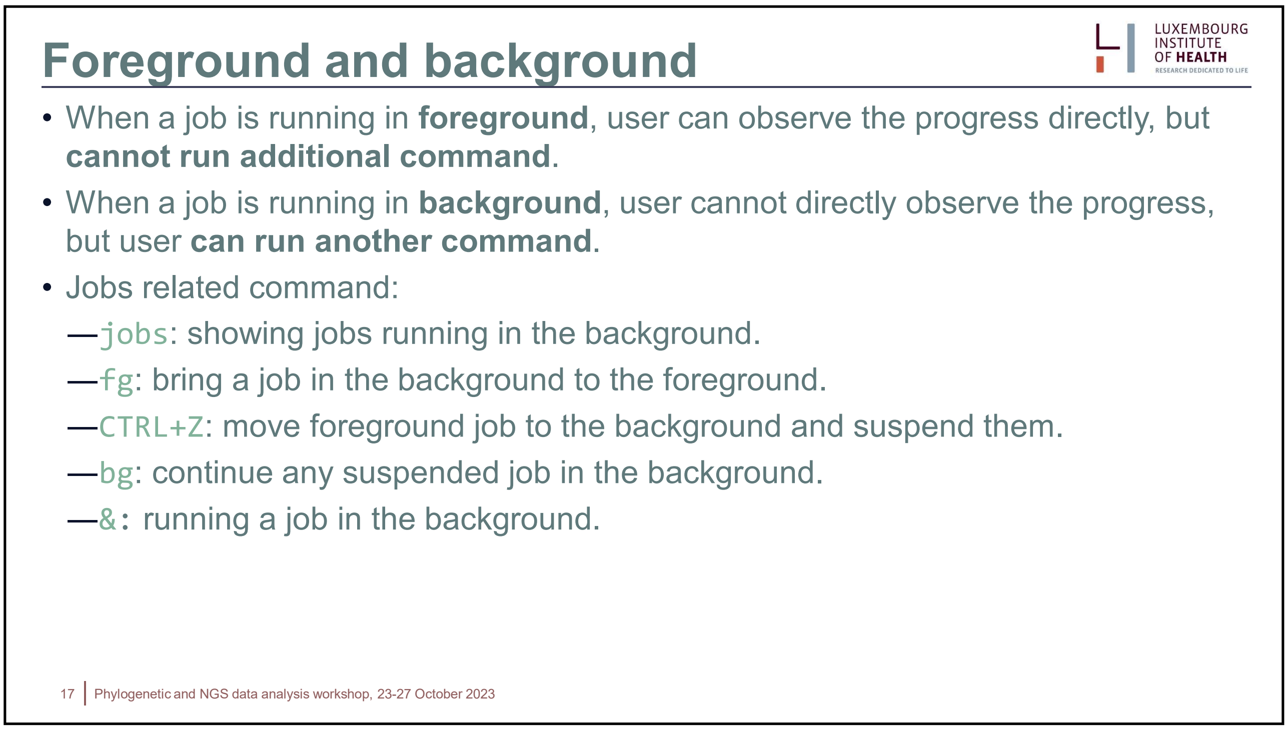
```
file1.txt
file2.txt
file3.txt
```

```
ls: cannot
access './doc/':
No such file or
directory
```

- The standard channels 0, 1, and 2 can be replaced with files.
- User can output the STDOUT and STDERR into a file.
  - `ls ./doc/ > filelist.txt #STDOUT` (or `>>` to append in a same file).
  - `ls ./doc/ 2> filelist.txt #STDERROR`
- Or, user can input STDIN from a file.
  - `grep string < file.txt`

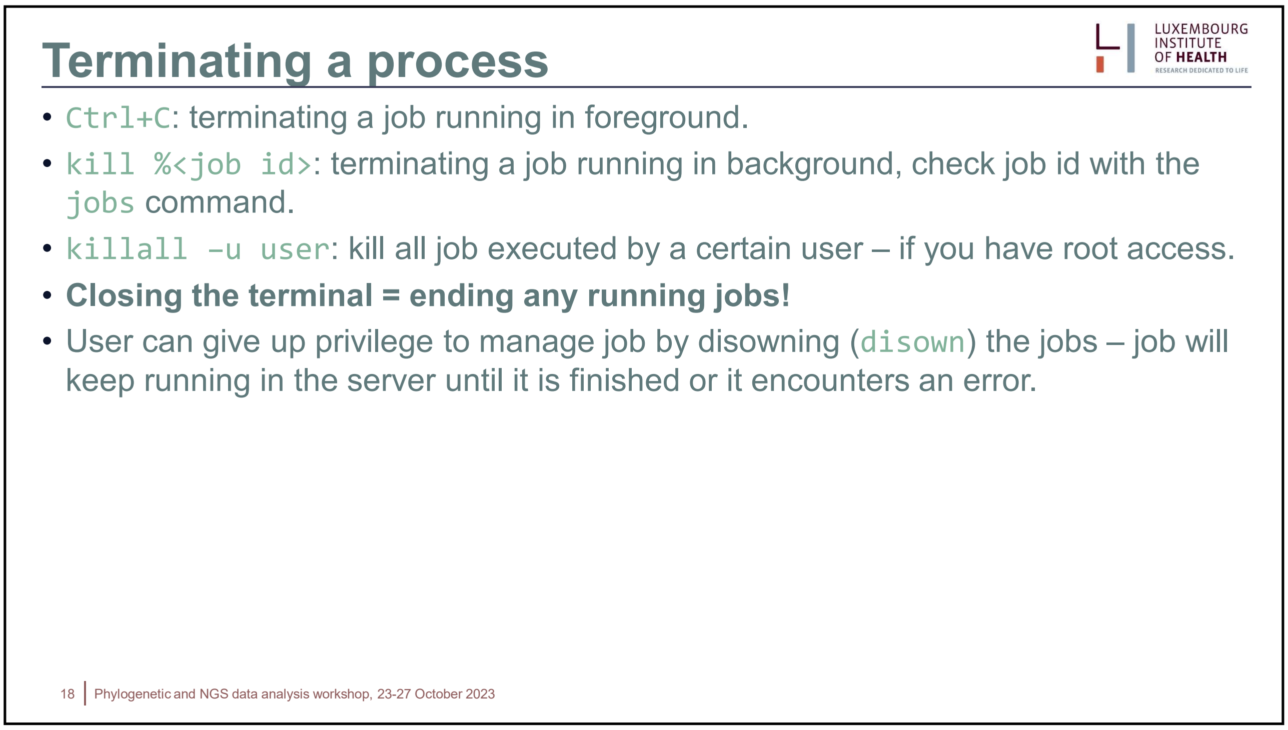
Piping

- Output from one command can be used as an input to the next command – this is called piping.
- The command should be joined by “|”.
- Example:
  - `cat file.txt | sort | uniq`
  - `grep string file.txt | awk '{print $1}' > output.txt`
- Useful when combining commands.



Foreground and background

- When a job is running in **foreground**, user can observe the progress directly, but **cannot run additional command**.
- When a job is running in **background**, user cannot directly observe the progress, but user **can run another command**.
- Jobs related command:
  - —`jobs`: showing jobs running in the background.
  - —`fg`: bring a job in the background to the foreground.
  - —`CTRL+Z`: move foreground job to the background and suspend them.
  - —`bg`: continue any suspended job in the background.
  - —`&:` running a job in the background.



Terminating a process

- `Ctrl+C`: terminating a job running in foreground.
- `kill %<job id>`: terminating a job running in background, check job id with the `jobs` command.
- `killall -u user`: kill all job executed by a certain user – if you have root access.
- **Closing the terminal = ending any running jobs!**
- User can give up privilege to manage job by disowning (`disown`) the jobs – job will keep running in the server until it is finished or it encounters an error.

Tips

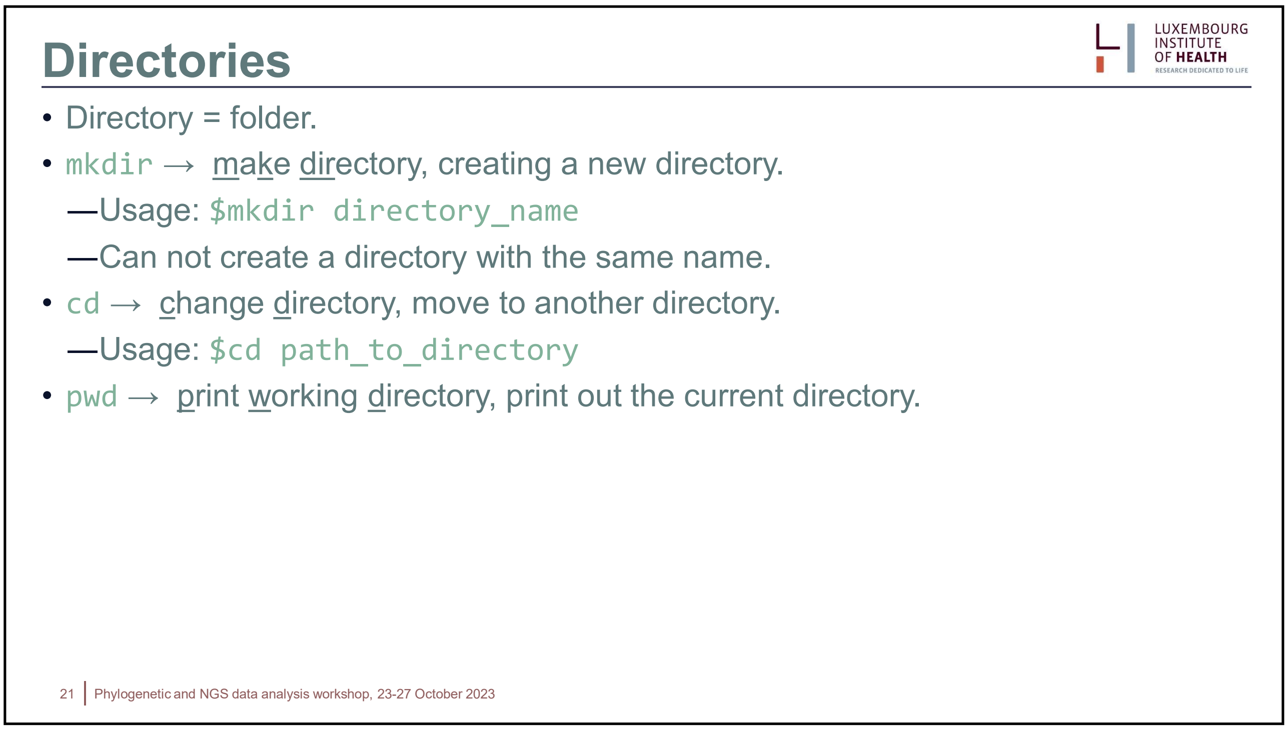
- Use the up and down arrow keys to access previous commands.
- Autocomplete = tab button.
- "/" is not allowed to be used in a file name.
- Avoid using space or other special characters in file/directory name.

man

- `man` → Printing out the manual of a certain command.
- Similar to `–h –help` or `–help.`
- Example

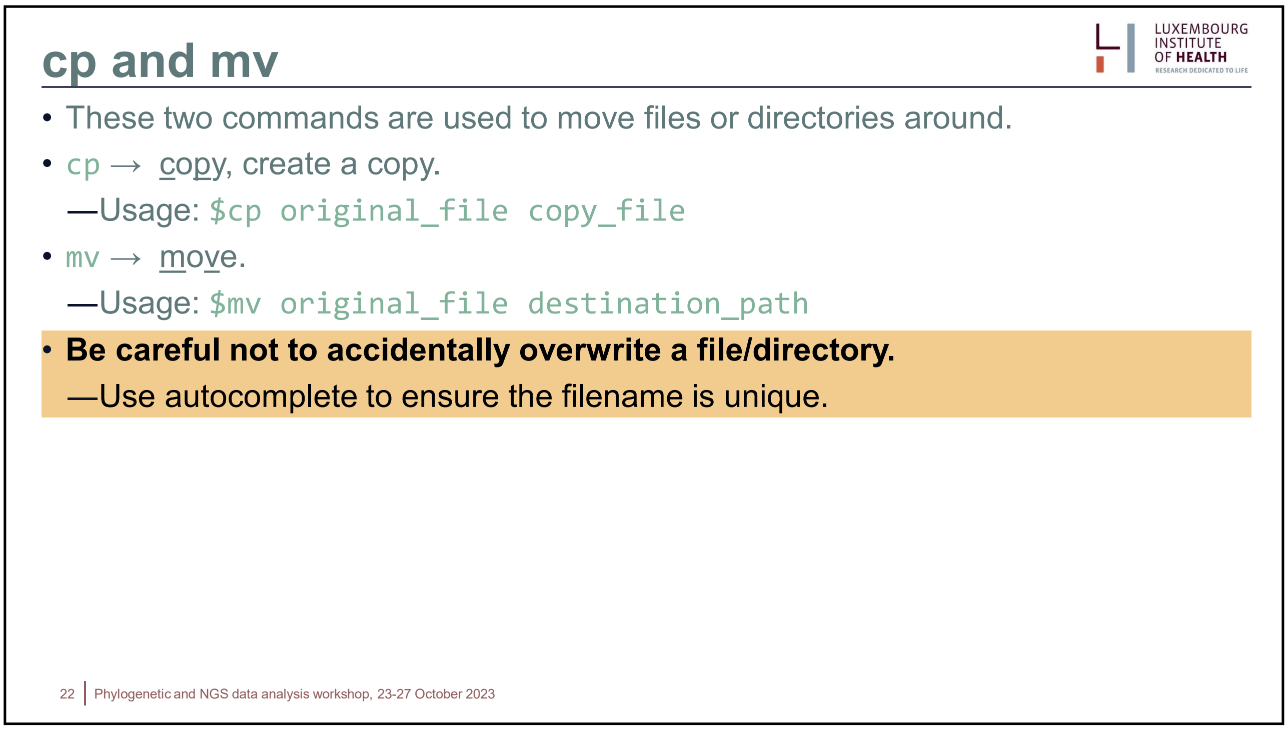
```
$man grep
$grep --help
```

- Press "Q" to quit the manual.



Directories

- Directory = folder.
- `mkdir` → make directory, creating a new directory.
  —Usage: `$mkdir directory_name`
  —Can not create a directory with the same name.
- `cd` → change directory, move to another directory.
  —Usage: `$cd path_to_directory`
- `pwd` → print working directory, print out the current directory.



cp and mv

- These two commands are used to move files or directories around.
- `cp` → copy, create a copy.
  —Usage: `$cp original_file copy_file`
- `mv` → move.
  —Usage: `$mv original_file destination_path`
- **Be careful not to accidentally overwrite a file/directory.**
  —Use autocomplete to ensure the filename is unique.

rm

- `rm` → remove, deleting a file or a directory.
  - —Add the recursive option "`-r`" for directory.
- **No warning, no backup!**
  - —Once rm-ed, it is gone forever.

ls

- `ls` → list, print out the list of files present in a directory
- usage: `$ls options directory`
  - —Option `-l` gives full list, including permission, date created, and size.
  - —Use `man ls` to get the full options list or `ls -h`.

grep

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

- `grep` → generalized regular expression parser, find a **string** in a plain-text data.
  - —Usage: `$grep options “search_term” file.txt`
  - —The good practice is to always keep the **string** inside quotation marks.
  - —The output is **lines** in `file.txt` that contains the `search_term`.
- Useful options:
  - —“`-A`” and “`-B`” – get nth line after and before the match.
  - —“`-w`” – match only whole words.
  - —“`-f`” – find pattern from a file.
  - —Use “`grep --help`” to get the full options.

string = a series of characters.

sort

LUXEMBOURG INSTITUTE OF HEALTH
RESEARCH DEDICATED TO LIFE

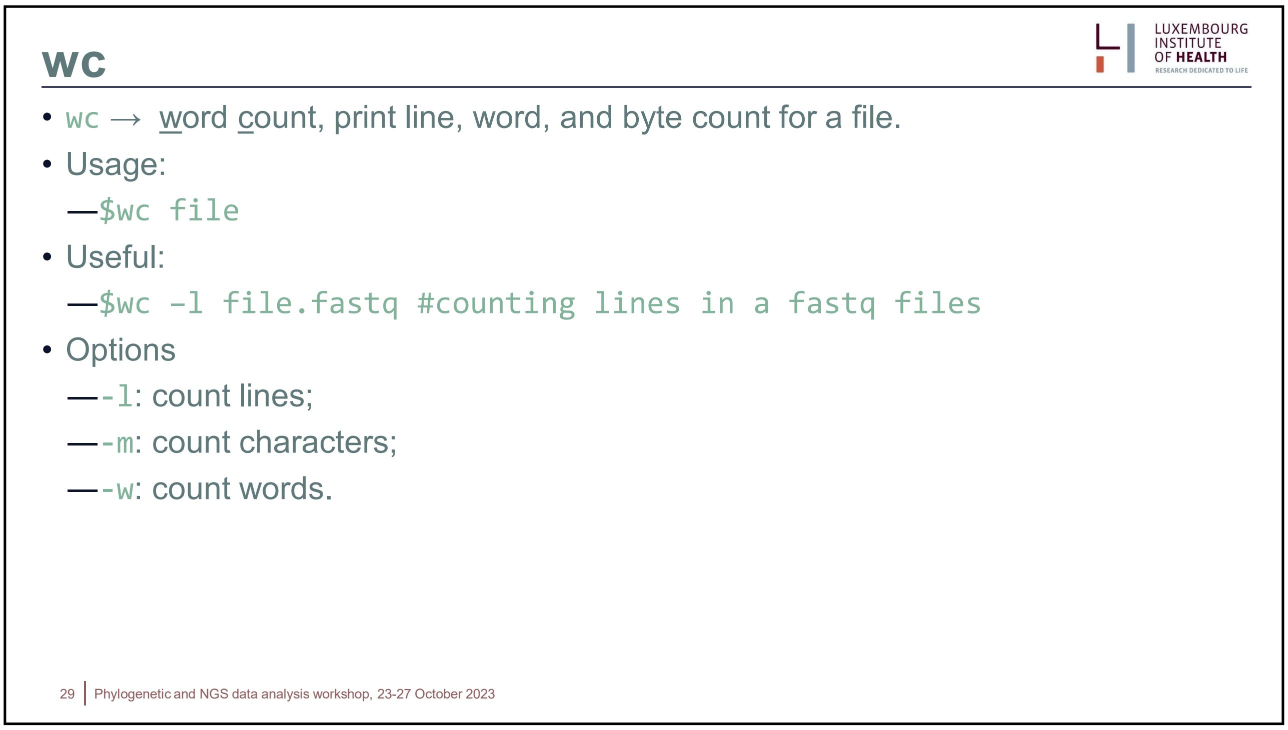
- `sort` → sort the lines.
- Usage:
  - —`$sort file`
  - —`$grep chr1 file.txt | sort`
- Options to sort based on number or alphabet, reverse sort, and other options are available.
  - —`-r`: sort in reverse order.
  - —`-n`: treat the data as integer.
  - —`-k`: sort the data based on a column.

uniq

- `uniq` → unique, remove repeated lines – file needs to be sorted.
- Usage:
  - `$uniq file`
  - `$grep chr1 file.txt | sort | uniq`
- Available options:
  - `-d`: print only the redundant lines.
  - `-c`: show how many times the redundant lines appear.
  - -i: ignore case sensitivity.
  - -s <INT>: skip the first <INT> characters when making comparison.

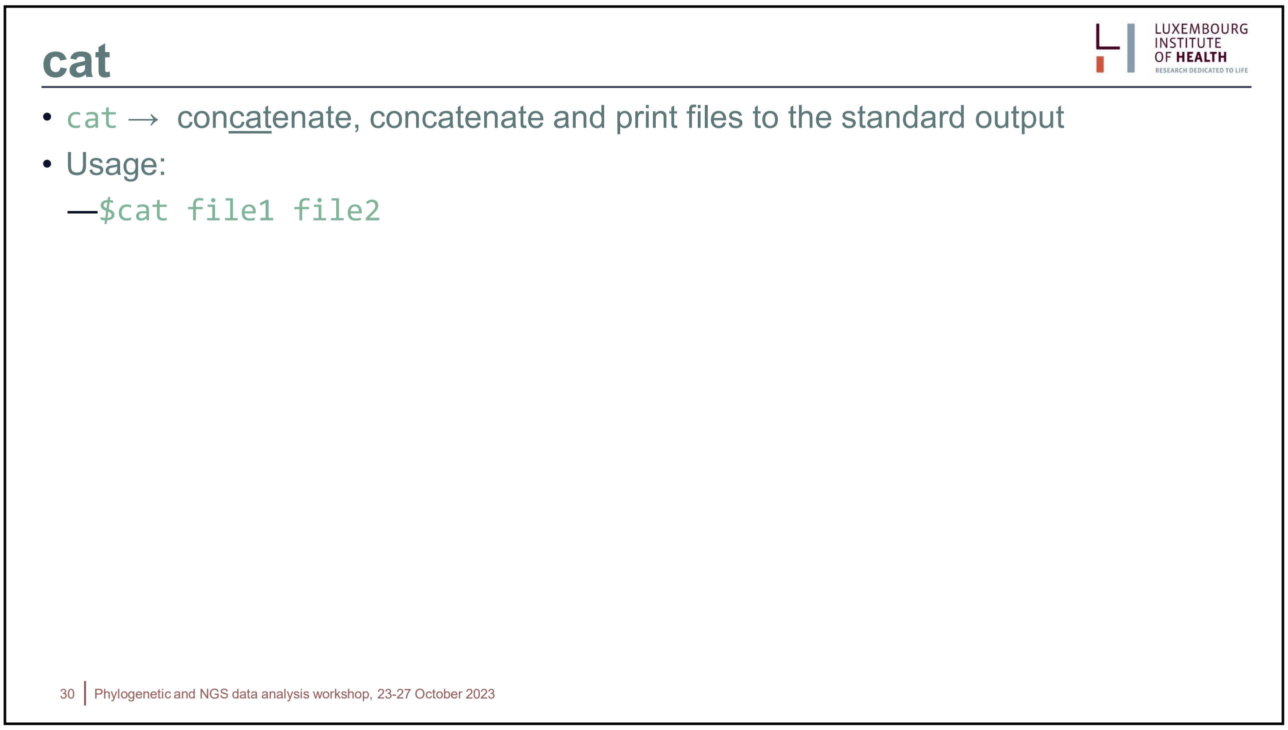
comm

- `comm` → common, find common lines between two files.
- Usage:
  - `$comm file1 file2`
- Options:
  - "`-1`" print lines common for both files and lines unique to file2.
  - "`-2`" print lines common for both files and lines unique to file1.
  - "`-3`" print lines unique to file1 and file2.
  - "`-12`" print lines common for both files.
- Both files must be in sorted order.



wc

- `wc` → word count, print line, word, and byte count for a file.
- Usage:
  - `$wc file`
- Useful:
  - `$wc -l file.fastq #counting lines in a fastq files`
- Options
  - `-l`: count lines;
  - `-m`: count characters;
  - `-w`: count words.



cat

- `cat` → concatenate, concatenate and print files to the standard output
- Usage:
  - `$cat file1 file2`

Preview a file

- Only for text file! Binary file cannot be previewed.
- `less` → preview a file without reading the entire file.
- `more` → similar to `less` but the program reads the entire file first.
- `head` → print lines from the start of the file (default 10 lines); `head -20` = print the first 20 lines.
- `tail` → print lines from the end of the file (default 10 lines); `tail -20` = print the last 20 lines.

Loop

- Very useful for automation.
- The most common is the for in command:

```
$for var in A B C; do rm file${var}.txt; done
fileA.txt
fileB.txt
fileC.txt
```

—A list name `var` is defined by the user that contains the letter A, B, and C.
—The `rm` command will be repeated to the length of `var` (in this case three times).
—The `${var}` part will be substituted with different letter in each repetition.

Loop

- Other useful command for loop is while read.

```
$cat list.txt
A
B
C
$while read var; do echo file${var}.txt; done < list.txt
fileA.txt
fileB.txt
fileC.txt
```

- Similar to previous, but instead of defining the variable in the command, we asked the computer to use each line in the file list.txt as the list of variable.

Sed

- sed → stream editor, for parsing and transforming text (similar to awk).
- Example:
  - `sed 's/string1/string2' file #substitute (s) string1 to string2 in a file.`

Awk

- awk is a domain-specific language designed for text processing and typically used as a data extraction and reporting tool.
  - —awk = Aho, Weinberger, and Kernighan, its designer.
- Useful in manipulating columns of data.
- Example
  - —`$ awk '{print $5}' file`
  - —`$ awk  awk 'NR%2==1' file`
  - —`awk '$3=="www"'`
- More explanation: https://www.gnu.org/software/gawk/manual/gawk.html.

Practice time~