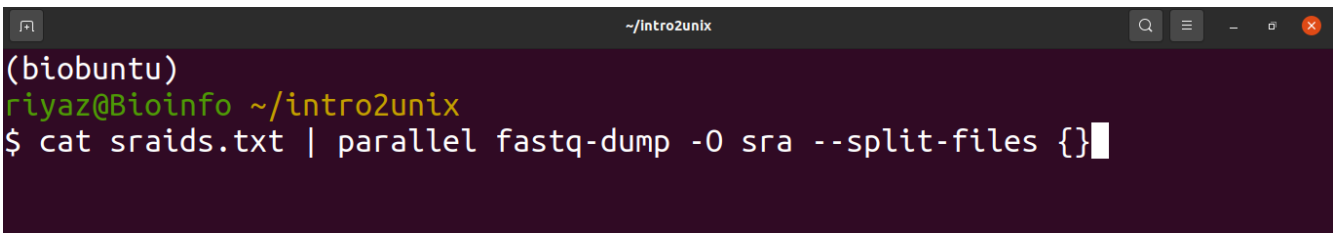
Introduction to Unix and Linux

- **Unix** is a class of operating system
  - Used the command line as the primary mode of interaction with the user.
- Developed by AT&T Bell labs and is not open source (licensed OS)
- **Linux** is an open source operating system built by **Linus Torvalds**
  - Commonly used distros: Ubuntu, Mint, Debian, Zorin, Fedora, Kali Linux, Fedora etc.

- "**Linux**" and "**Unix**" and "**Command Line**" are used interchangeably

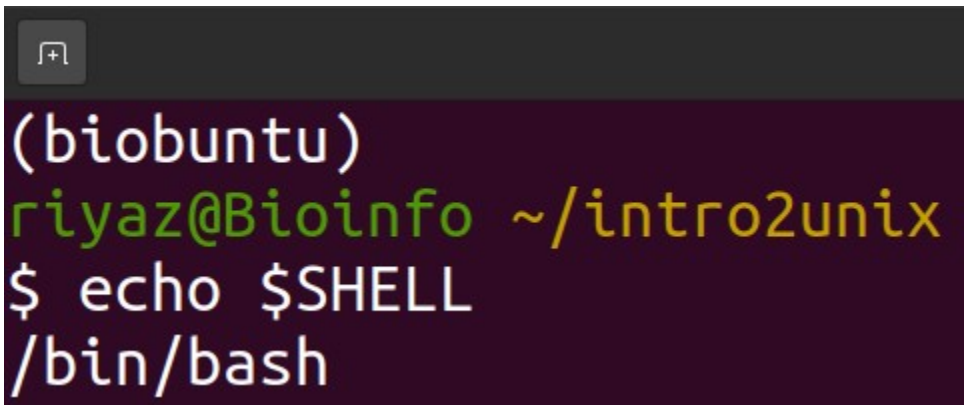
What does the command line look like?



What is Shell?

- Shell is an **environment** in Unix system
    - Provides a **command line interface** to run our commands, programs, and shell scripts.
- In Unix, there are two major types of shells −
    - **Bourne shell** − **$** character is the default command prompt
    - **C shell** − **%** character is the default command prompt (csh)
- The Bourne Shell has the following subcategories −
    - Bourne shell (sh)
    - Korn shell (ksh)
    - Bourne Again shell (bash)

The shell is not just a way to launch commands but a full programming language



Is it mandatory to learn command line for Bioinformatics Analysis?

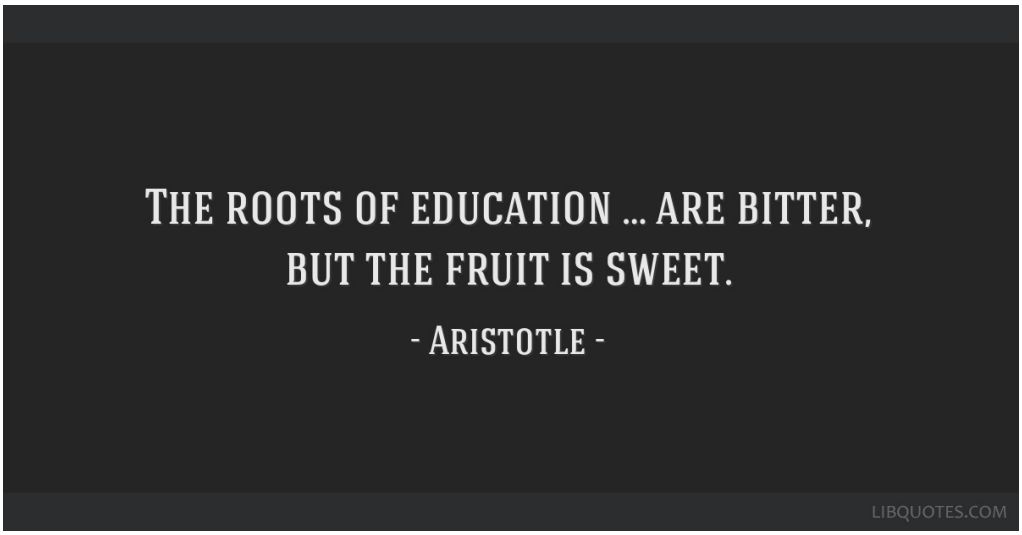
- Depends on your preference.

What are the advantages of the command line?

- Easy to share
- Reproducibility
- Combination of multiple tools
- Automation

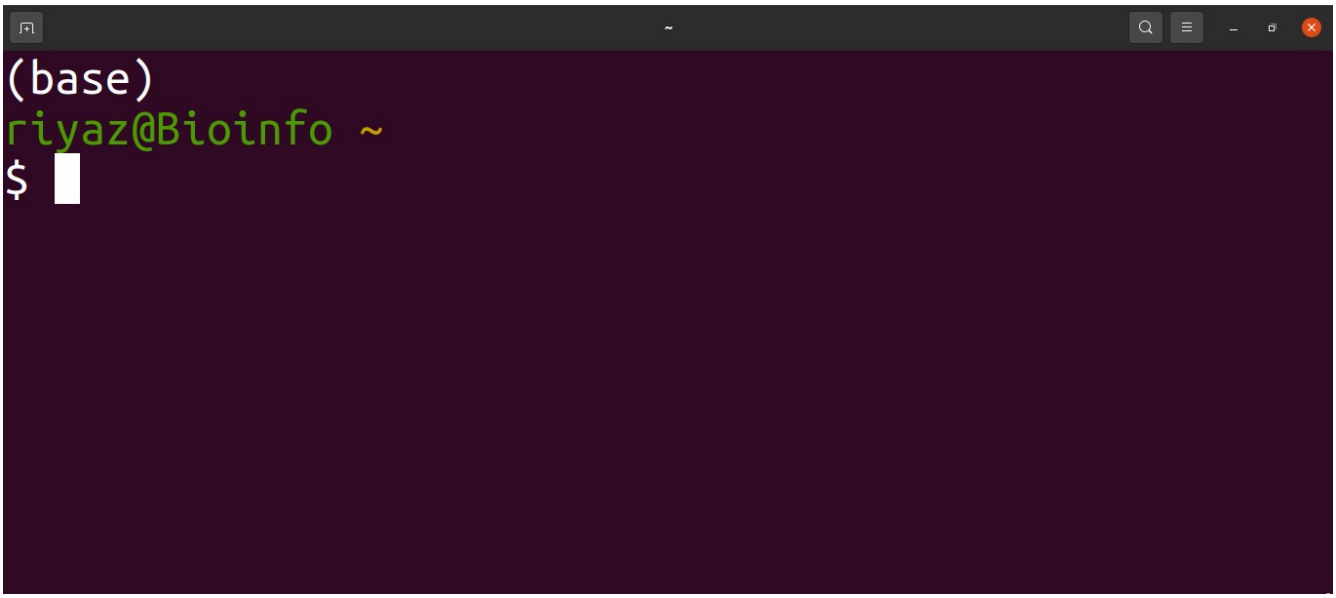
What are the disadvantages of the command line?

- Requires additional training
- Initially overwhelming



How do I access the command line?

- Command line shell is accessed via an application called “Terminal”
    - By default “Terminal” launches shell in home directory.



Let's Start

```
(biobuntu)
riyaz@Bioinfo ~/intro2unix
$ echo 'Welcome to Bioinformatics World'
Welcome to Bioinformatics World
```

```
(biobuntu)
riyaz@Bioinfo ~/intro2unix
$ echo "Welcome to Bioinformatics World"
Welcome to Bioinformatics World
```

- **!** character allows you to rerun a previous command.
  - **!e** will rerun the last command that starts with **e**
  - **!!** will rerun the previous command.

Our First Unix Command

- **ls** : lists the contents of a directory

```
(base)
riyaz@Bioinfo ~/intro2unix
$ ls
dir1  dir2  file.txt  script_1.sh  test
```

- **ls** can used to list the contents of any directory
- For example:
  - ls /bin
  - ls /
- Flags/options for **ls** command;
  - -l : uses long listing format
  - -t : sorts output based on file modification date
  - -S : sorts output by size
  - -r : reverse-sorts the output
  - -R : recursively lists output of all directories below current level
  - -1 : forces output to be one entry per line
  - -p : puts / indicator to directories
- You can combine more than one flags together, e.g.; ***ls -1trp***

```
(base)
riyaz@Bioinfo ~/intro2unix
$ ls -1trp
test/
script_1.sh
dir2/
dir1/
file.txt
```

Man (manual) Pages

- ***man command***
  - e.g.; *man ls*
  - **space key** : to scroll down a page
  - **b** : to go back a page
  - **q** : to quit manual page

Finding out where you are

- ***pwd*** : will print your current working directory

```
(base)
riyaz@Bioinfo ~/intro2unix
$ pwd
/home/riyaz/intro2unix
```

Let's make a new directory/folder

- ***mkdir*** : makes new directory

```
(base)
riyaz@Bioinfo ~/intro2unix
$ ls
dir1  dir2  file.txt  script_1.sh  test
(base)
riyaz@Bioinfo ~/intro2unix
$ mkdir new_dir
(base)
riyaz@Bioinfo ~/intro2unix
$ ls
dir1  dir2  file.txt  new_dir  script_1.sh  test
(base)
```

- important flag of ***mkdir*** command:
  - -p : no error if existing, make parent directories as needed

Navigating through directories

- ***cd*** : to change directories

```
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix
$ pwd
/home/arriyaz/Desktop/intro2unix
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix
$ cd dir1/subdir1/
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix/dir1/subdir1
$ pwd
/home/arriyaz/Desktop/intro2unix/dir1/subdir1
```

- ***cd /*** : navigate to root directory
- ***cd ~*** or, ***cd*** : navigate to home directory (~ sign is called tilde)
- ***cd ..*** : to go one step upward
- ***cd ../..*** : to go two step upward etc….

Creating Empty File with touch Command

```
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ touch 1file.txt 2file.txt 3file.txt 4file.txt
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls
1file.txt  2file.txt  3file.txt  4file.txt
(base)
```

Removing Contents and Directories

- ***rmdir*** : removes empty directories

Note, you have to be outside a directory before you can remove it with *rmdir*

```
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix
$ ls
dir1  dir2  dir3
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix
$ rmdir dir2
(base)
```

- Flags for ***rmdir***:
  - -p : remove DIRECTORY and its ancestors

```
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix
$ ls -R dir2
dir2:
subdir2

dir2/subdir2:
subdir2

dir2/subdir2/subdir2:
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix
$ rmdir -p dir2/subdir2/subdir2/
```

- ***rm*** : removes files and directories

NB: Be careful while using this command. It's extremely hard to recover deleted files by *rm* command.

```
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix
$ rm dir1
rm: cannot remove 'dir1': Is a directory
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix
$ touch foo.txt
(base)
arriyaz@riyaz-V131 ~/Desktop/intro2unix
$ rm foo.txt
rm: remove regular empty file 'foo.txt'? y
```

- Important flags of ***rm*** command:
  - **-i** : will ask for confirmation before deleting anything
  - **-f** : won't ask for permission before deleting
  - **-r, -R** : remove directories and their contents recursively
  - **-d** : remove empty directories
- **Tips:** use wildcards and have fun

```
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls
1file.txt  2file.txt  3file.txt  4file.txt
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ rm *.txt -f
```

Moving Files and Directories

- **mv** : move (rename) files
- synopsis: ***mv source desitation***

```
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ mkdir temp
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ touch earth.txt heaven.txt
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ mv heaven.txt temp
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ mv earth.txt temp
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls ./temp
earth.txt  heaven.txt
```

Move file with wildcards

```
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ mv temp/*.txt .
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls
code.txt  earth.txt  foo.txt  heaven.txt  itismorelonger0987.txt  notsolong.txt  temp
```

```
mv *.txt temp
mv *t temp
mv *ea* temp
```

Renaming Files with *mv* command

```
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ touch afile.txt
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls
afile.txt  earth.txt  heaven.txt              notsolong.txt
code.txt   foo.txt    itismorelonger0987.txt  temp
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ mv afile.txt anotherfile.txt
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls
anotherfile.txt  earth.txt  heaven.txt              notsolong.txt
code.txt         foo.txt    itismorelonger0987.txt  temp
```

- **Tips** : type ***man rename*** to explore more about renaming

Moving Directories

```
$ mkdir test
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ mv test temp
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls temp
test
```

Copying Files and Directories

- **cp** - copy files and directories

```
$ echo "Turning earth into heaven" > earth.txt
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ cat earth.txt
Turning earth into heaven
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ cp earth.txt heaven.txt
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ cat heaven.txt
Turning earth into heaven
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls
anotherfile.txt  code.txt  earth.txt  foo.txt  heaven.txt  itismorelonger0987.txt  notsolong.txt  ramp  temp
```

Copying Directories

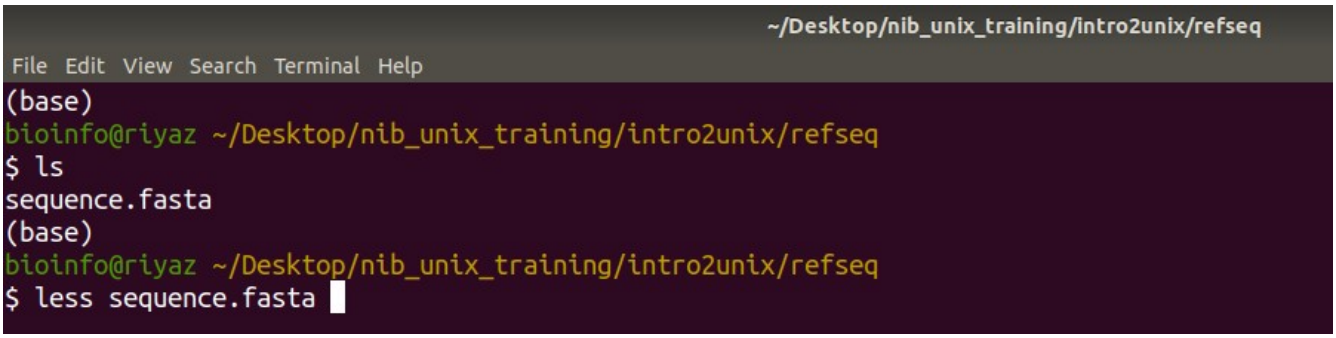
- Use **-r** or **-R** flags to copy directories *recursively.*
- **> symbol** : redirect content into an output file

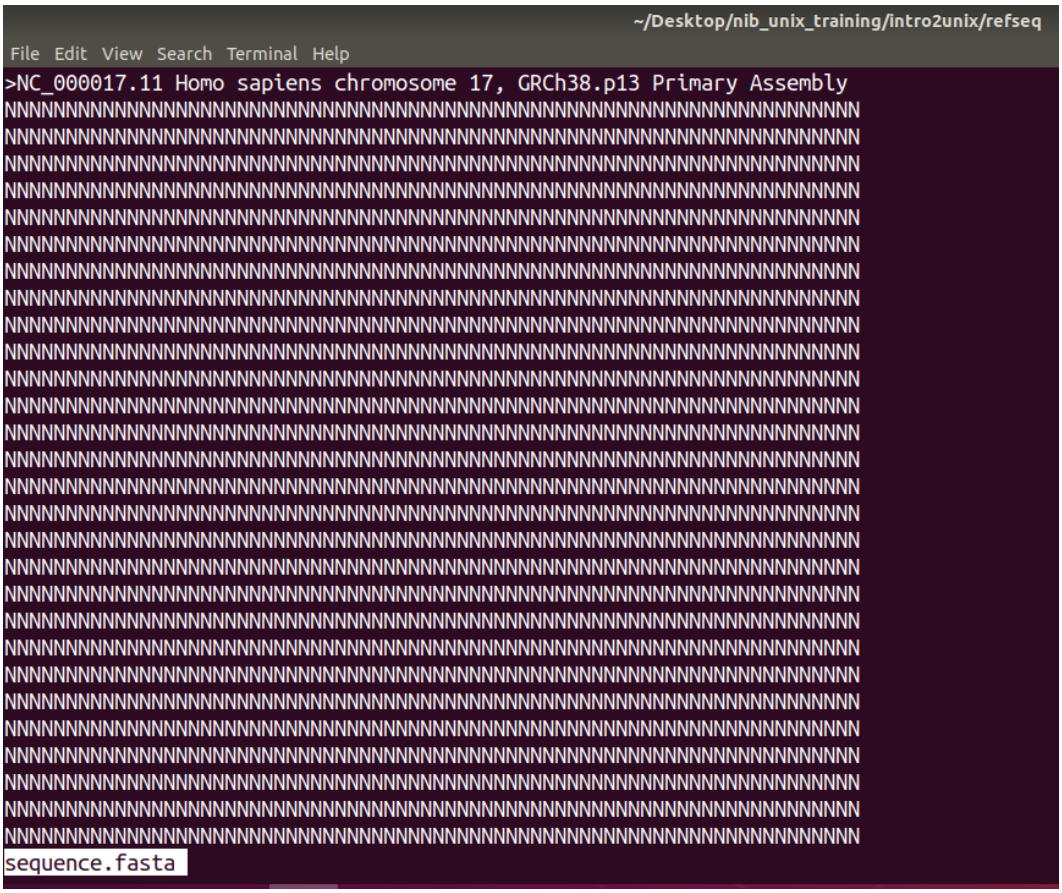
NB: Careful when using file redirection (>), it will overwrite any existing file of the same name

```
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ echo "Let's copy a directory and its contents" > temp/file.txt
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ cat temp/file.txt
Let's copy a directory and its contents
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls temp/
file.txt  test
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ cp -r temp/ copy_temp
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ ls copy_temp/
file.txt  test
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix
$ cat copy_temp/file.txt
Let's copy a directory and its contents
```

Viewing files with less or, more

- ***more*** and ***less*** : used to view files in terminal
- ***less*** is much effective than ***more*** and it can open larger files effectively
- use ***space*** key to go forward and ***b*** key to go backward
- ***h*** **key** for help and ***q*** to quit





Another Approach to Open and View Files in Terminal

- ***cat*** command plus ***head*** command
- Don’t open very large file via **cat command** without **head**

```
(base)
bioinfo@riyaz ~/Desktop/nib_unix_training/intro2unix/refseq
$ cat sequence.fasta | head -15
>NC_000017.11 Homo sapiens chromosome 17, GRCh38.p13 Primary Assembly
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
```

>> **will append our content to already existing file, where > sign will overwrite a file.**