Introduction to Linux

School on Atomistic Simulation of Biomolecules and Material Science

Dina Machuve (PhD)

Nelson Mandela African Institution of Science and Technology

03rd September 2018

Why Unix?

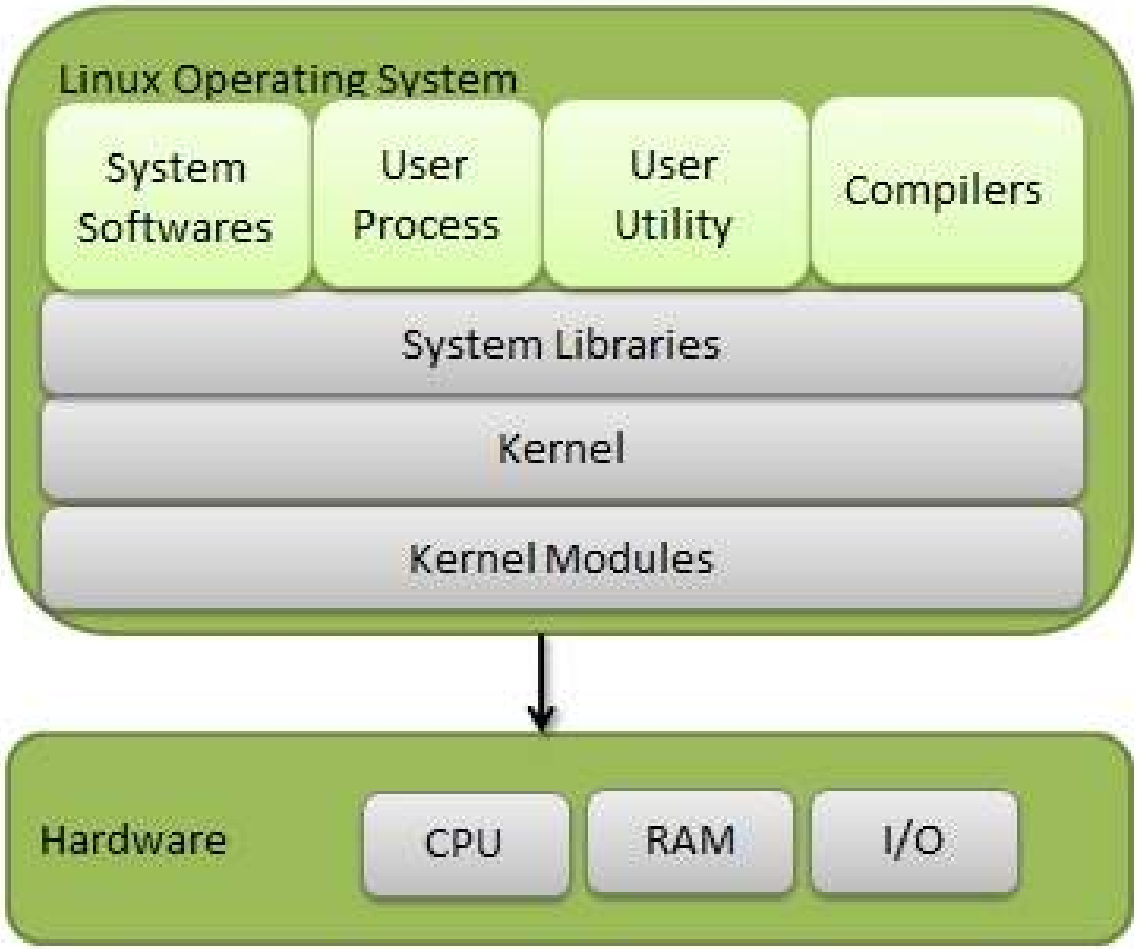
The **Unix** operating system has been around since 1969. Back then there was no such thing as a graphical user interface. You typed everything. There are several variants of Unix (including Linux), though the differences do not matter much for most basic functions.
Linux Kernel is an operating system (OS) kernel defined as Unix-like in nature. It used in different operating systems, mostly in the form of different Linux distributions (*Techopedia*).

History

Linux is a Unix clone written from scratch by Linus Torvalds with assistance from a loosely-knit team of hackers across the Net.

- It is open source and free to use
- It is the most secure OS, used in most of the servers
- There are many Linux distributions, but Ubuntu is the most popular
- Linux is only a Kernel

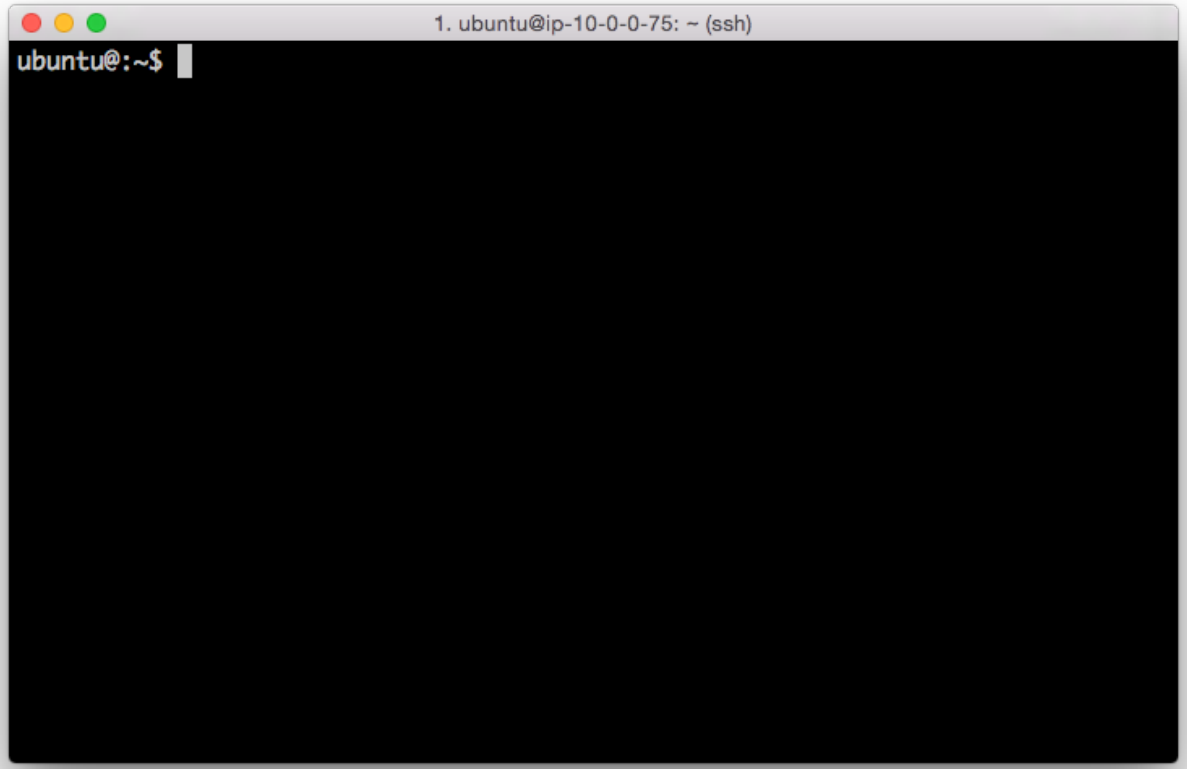
Kernel Architecture



The Terminal

The Terminal is a command line interface program.

- It allows you to type input to the computer (i.e. run programs, move/view files etc.)
- It allows you to see output from those programs.



Terminal Shortcuts

- Pressing `tab` will **autocomplete** file and folder names
- Control+C will **stop** execution of your current program!
- Control+R will let you **search** your command history!
- Control+L will **clear** your screen!
- `cmd arg1 ... argN > file1.txt` will put the output of `cmd` into `file1.txt`
- `cmd arg1 ... argN < file2.txt` will pull the input of `cmd` from `file2.txt`
- Use the **up** and **down** arrow keys to **scroll through your command history**.

Linux file pathing

- ~ is your **HOME DIRECTORY**
  - This is where you start from after you open the terminal
  - On bash, you can also use $HOME
- **.** is an alias for your **PRESENT WORKING DIRECTORY**
- **..** is the file path for the **PARENT DIRECTORY** of your present working directory.
- / is the file path for the **TOP-LEVEL DIRECTORY**

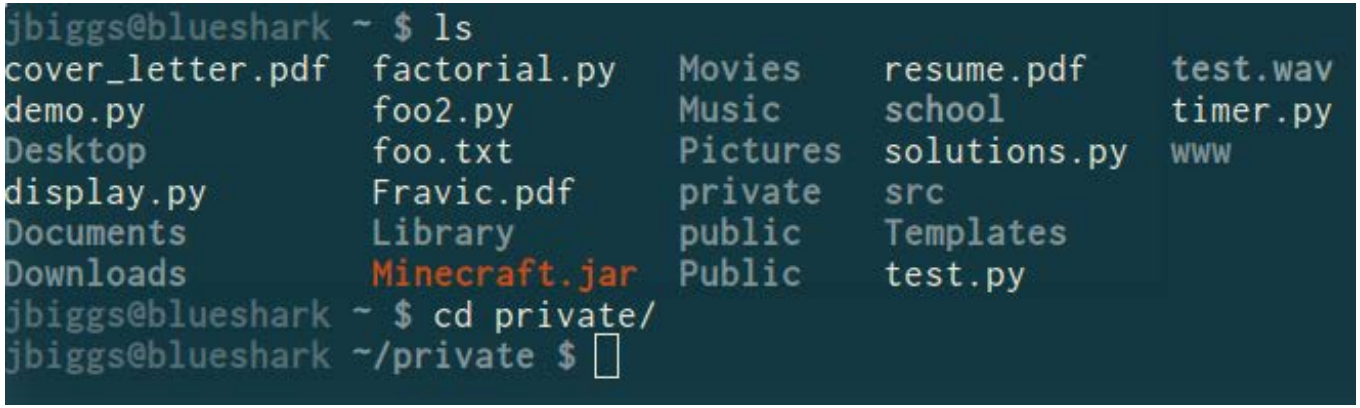
`ls <dir>` - LiSt

- Lists the files in the present working directory, or, if specified, `dir`.
- `pwd` tells your **P**resent **W**orking **D**irectory.

```
jbiggs@blueshark ~ $ ls
cover_letter.pdf  factorial.py   Movies    resume.pdf    test.wav
demo.py           foo2.py        Music     school        timer.py
Desktop           foo.txt        Pictures  solutions.py  www
display.py        Fravic.pdf     private   src
Documents         Library        public    Templates
Downloads         Minecraft.jar  Public    test.py
jbiggs@blueshark ~ $ pwd
/afs/andrew.cmu.edu/usr10/jbiggs
jbiggs@blueshark ~ $
```

cd <directory> - Change Directory

- Changes your present working directory to directory
- Your main tool for navigating a unix file system



mkdir <dirname> - MaKe DIRectory

- Makes a directory dirname in your present working directory.
- Directories and folders are the **same thing**.

```
jbiggs@blueshark ~ $ ls
cover_letter.pdf  factorial.py   Movies    resume.pdf    test.wav
demo.py           foo2.py        Music     school        timer.py
Desktop           foo.txt        Pictures  solutions.py  www
display.py        Fravic.pdf     private   src
Documents         Library        public    Templates
Downloads         Minecraft.jar  Public    test.py
jbiggs@blueshark ~ $ cd private/
jbiggs@blueshark ~/private $ mkdir 15-213
jbiggs@blueshark ~/private $ cd 15-213
jbiggs@blueshark ~/private/15-213 $
```

mv <src> <dest> - MoVe

- cp works in exactly the same way, but copies instead
  - for copying folders, use cp -r
- dest can be into an existing folder (preserves name), or a file/folder of a different name
- Also used to rename files without moving them
- src can be either a file or a folder

```
jbiggs@blueshark ~ $ cd private/
jbiggs@blueshark ~/private $ mkdir 15-213
jbiggs@blueshark ~/private $ cd 15-213
jbiggs@blueshark ~/private/15-213 $ mv ~/Downloads/datalab-handout.
tar .
```

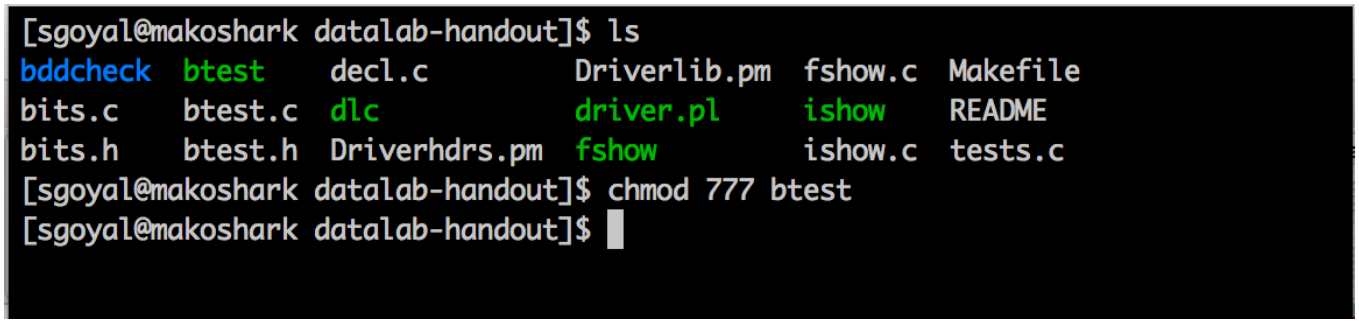
`tar <options> <filename>` - Tape ARchive

- Compression utility, similar to zip files on Windows
- For full list of options, see `man tar`
- As name suggests, was used on tapes
- `x` - extract, `v` - verbose, `f` - file input

```
jbiggs@blueshark ~/private/15-213 $ tar xvf datalab-handout.tar
datalab-handout/
datalab-handout/bits.c
datalab-handout/Makefile
datalab-handout/README
datalab-handout/btest.h
datalab-handout/btest.c
datalab-handout/bits.h
datalab-handout/decl.c
datalab-handout/tests.c
datalab-handout/fshow.c
```

chmod <permissions> <src>

- chmod is used to change the permissions of a file or directory.
  - 777 will give all permissions
  - src can be either a file or a folder



`rm <file1> <file2> ... <filen>` - ReMove

- Essentially the delete utility
- To remove an (empty) directory, use `rmdir`
- To remove a folder and its contents, use `rm -rf`
  - Please be careful, don't delete your project.
  - There is no "Trash" here. It's gone.
  - If someone asks you to use `rm rf /` ignore them

What's in a file? (using `cat`)

- `cat <file1> <file2> ... <filen>` lets you display the contents of a file in the terminal window.
- Use `cat -n` to add line numbers.
- You can combine multiple files into one.
- `cat <file1> ... <filen> > file.txt`
- Good for seeing what's in small files.

What's in a file? (using `less`)

- `less <file>` will give you a scrollable interface for viewing large files **without** editing them.
  - To find something, use /
  - To view the next occurrence, press `n`
  - To view previous occurrence, press `N`
  - To quit, use **q**

What's in a file? (using grep)

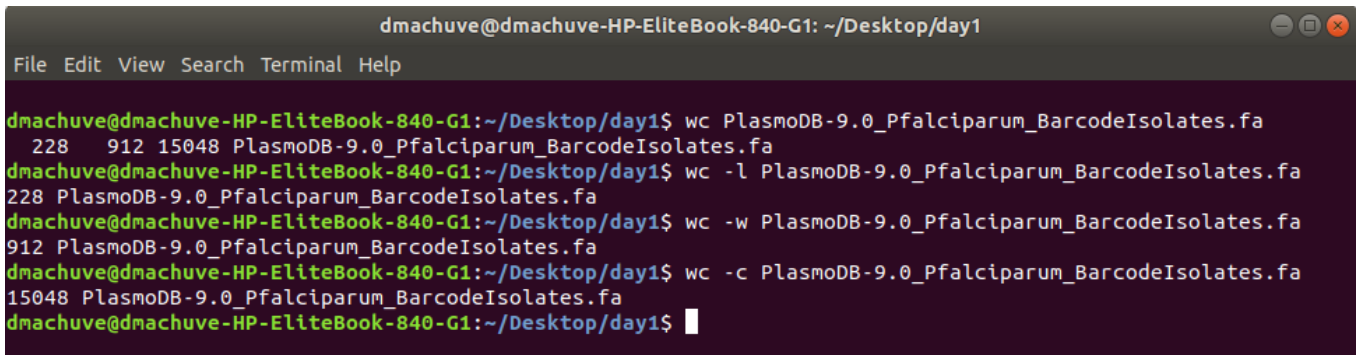
- grep `<pattern>` `<file>` will output any lines of file that have pattern as a substring
  - `grep -v` will output lines *without* pattern as substring
  - `grep -R` will search *recursively*
- Try it: `grep` 'mRNA' nrf1_seq.fa
  - `grep -v` 'ACT' nrf1_seq.fa
  - `grep -R` 'DNA'

man <thing>

- What is that command? What is this C standard library function? What does this library do? Check to see if it has a man page.
- Try it!
  - man grep
  - man tar
  - man printf
  - man strlen

wc <options> fileName

- wc is used to find out number of **newline** count, **word** count, **byte** count in a files specified by the file arguments.
- wc, or 'word count', prints a count of newlines, words, and bytes for each input file.
- The following are the 'options' and usage provided by the command:
  - wc -l : Prints the number of lines in a file.
  - wc -w : prints the number of words in a file.
  - wc -c : Displays the count of bytes in a file.
  - wc -m : prints the count of characters from a file.
  - wc -L : prints only the length of the longest line in a file.



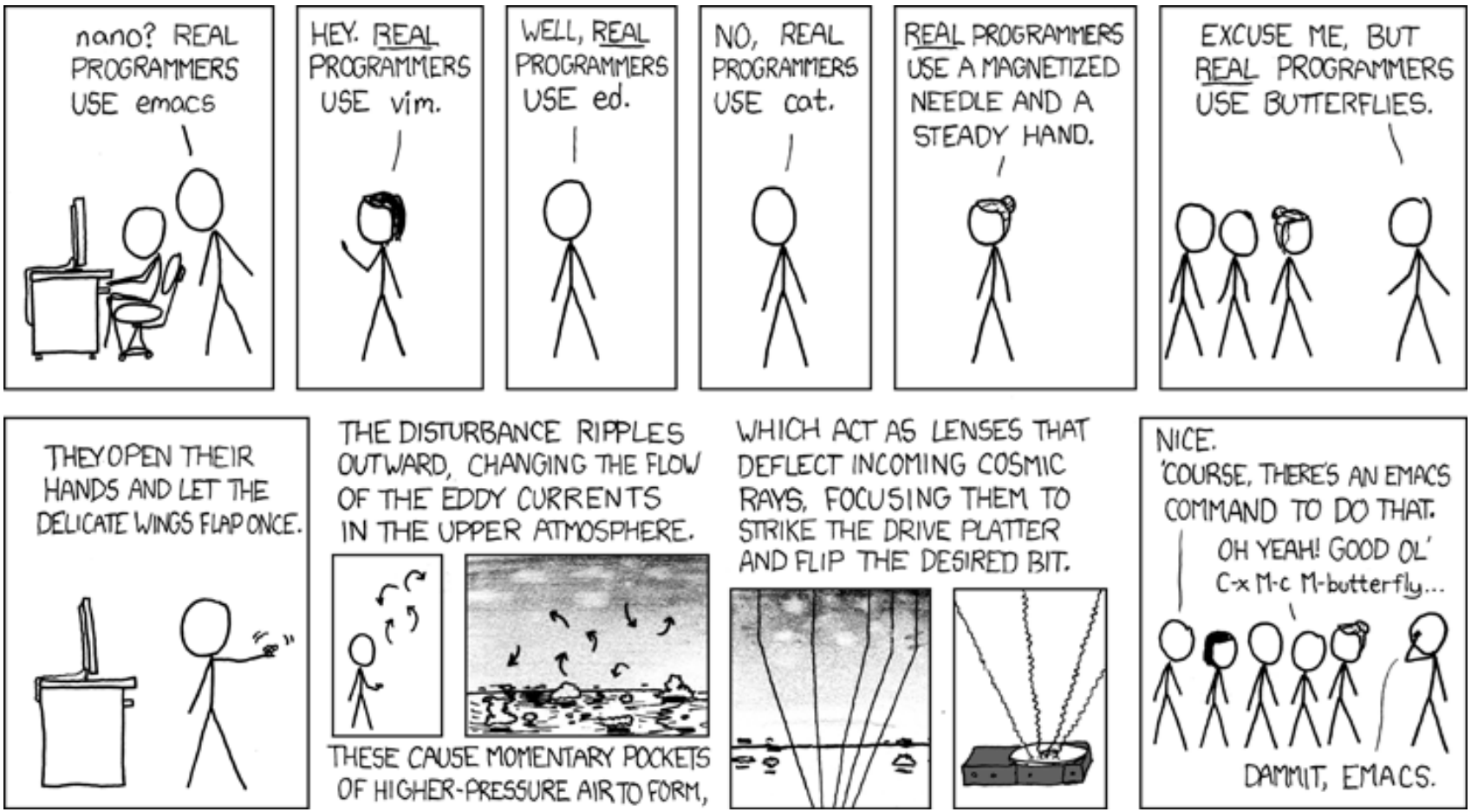
`head [options] [file(s)]`

- The `head` command reads the first ten lines of a any given file name.
- If it is desired to retrieve more number of lines than the default ten, then '-n' option is used along with an integer telling the number of lines to be retrieved. For example, the following command will display first 5 lines from the file '/var/log/check.log' file.
- For example, `$head -5 /var/log/check.log`

`tail [options] [file(s)]`

- The `tail` command allows you to display last ten lines of any text file.
- Similar to the head command above, tail command also support options 'n' number of lines and 'n' number of characters.
- For example, `$tail -5 /var/log/access.log`

Editors (a touchy subject)



Vim (`vi` – improved) Basics

Some different modes:

- Normal mode:
  - The first mode you enter. Hit the **escape key** to return to this mode at any time.
  - Everything entered here is interpreted as a *command*
- Command-line mode:
  - Used for entering *editor commands* (necessary to save file & quit the editor)
  - Enter ':' in Normal mode to get to this mode
- Insert mode:
  - To edit text
  - Enter 'i' in Normal mode to get to this mode

Vim Basics

- Useful commands:
  - Copying/pasting/deleting lines:
    - yy (yank) or 5 yy (yank next 5 lines)
    - dd (delete) or 5 dd (delete next 5 lines)
    - p (paste)
  - Search (/search_string or ?search_string)
- Useful editor commands:
  - Write (w)
  - Quit (q) quit no-save (q!)

Reference

1. Introduction to Linux for the Eastern Africa Network of Bioinformatics Training (EANBiT) Training Course, 2018.
2. Jenna MacCarley et. al, Linux Boot Camp (2015)