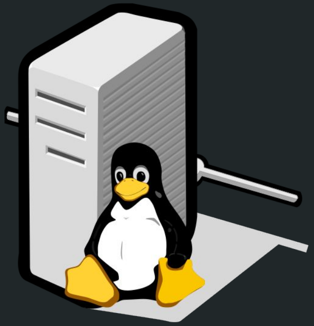
Intro to Linux

Advanced Genomics & Bioinformatics Workshop

James Oguya
Nairobi, Kenya
August, 2016



What is 'Linux' ?

- Linux is free & open source operating system originally developed for personal computers but has since been ported to more computer hardware platforms than any other operating system
- Created in October, 1991 by Linus Torvalds
- UNIX was expensive and restrictive
- Linux was born!
- Linux is:
  - free-to-use
  - open source: it's entire code is publicly available
  - stable: most stable OS you can find

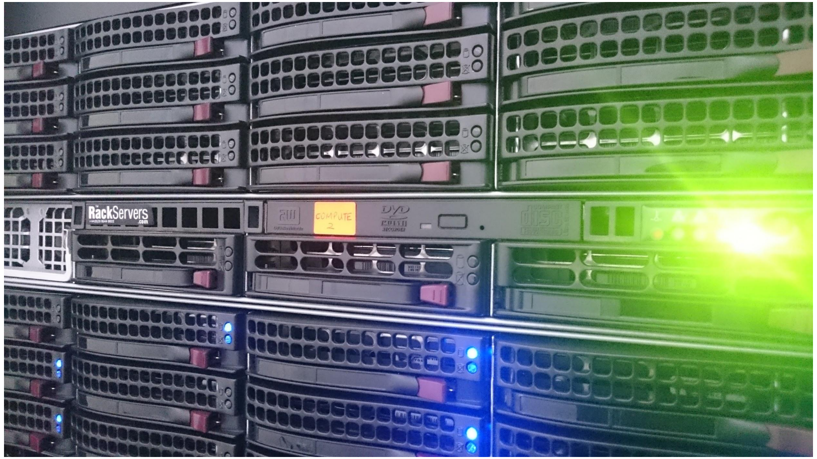
Why use Linux for Bioinformatics

- Scientific datasets are growing at an exponential rate & we need more processing power
    - computers with such power run Linux :)
    - Linux has a bunch of text processing tools
- Several bioinformatics tools & software are written & designed to run in Linux
    - bowtie, blast, clustaw, mira, tassel, e.t.c.

Getting started

Use an SSH client like MobaXterm or Putty to connect to our Linux server from Windows.

Server: `hpc.ilri.cgiar.org`
Username: `user1`
Password: `user1`

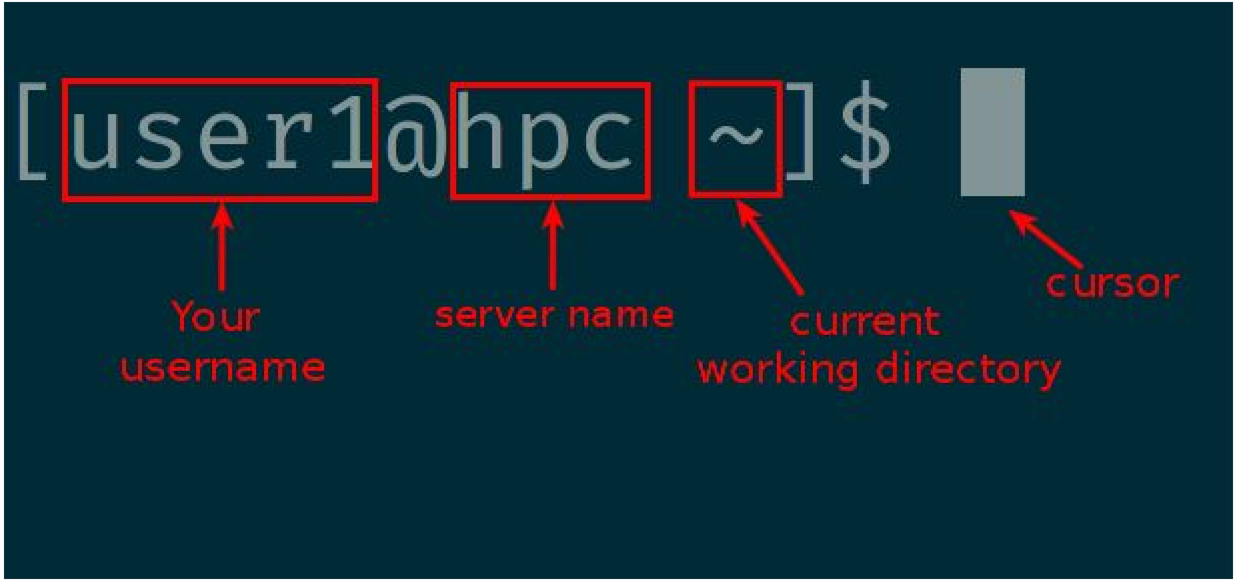


Linux shell

- Linux has a Graphical User Interface(GUI) just like Windows
  - GUI is not fun :(
  - real power lies in the Command Line Interface(CLI) or shell
- Using the shell is very easy!
  - just type a command and press Enter to run it :)
- Master the prompt!

Shell prompt

- Typically contains username, hostname & current working directory



```
[user1@hpc ~]$
```

Our first command(s)

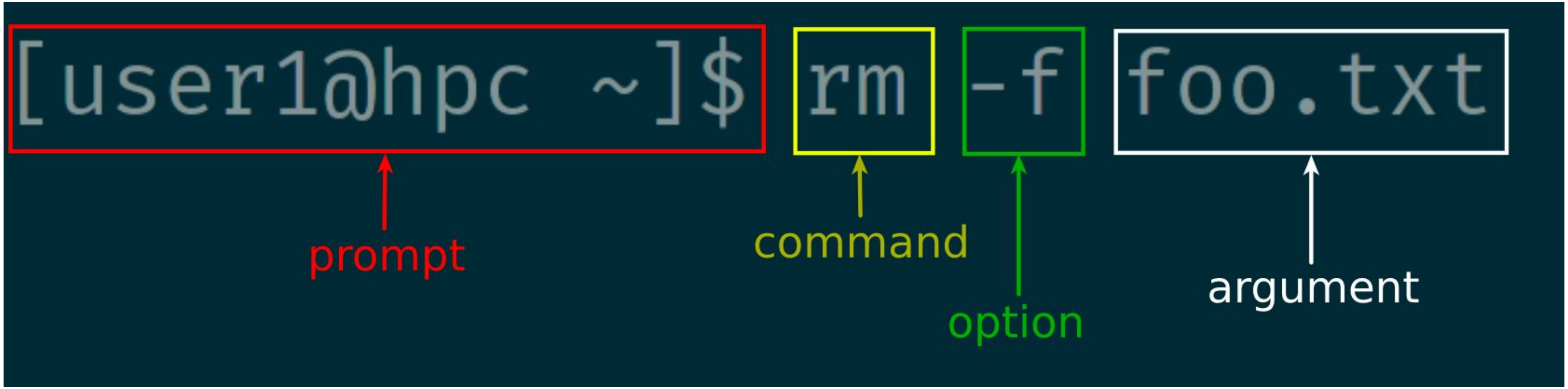
- Let’s start with the following commands(one at a time)
  - `whoami`
  - `cal`
  - `date`
- So what did each one of those command do??

Command structure

- Linux commands come in various forms
  - some are simple; can be used by themselves
    - `whoami`
    - `cal`
    - `date`
  - others require extra ‘parameters’; they don’t make sense to run by themselves
    - `mkdir`
    - `rm`
    - `cp`

Command structure

- Consists of a command, an option(flag) & an argument
  - separated by one or more spaces



```
[user1@hpc ~]$ rm -f foo.txt
```

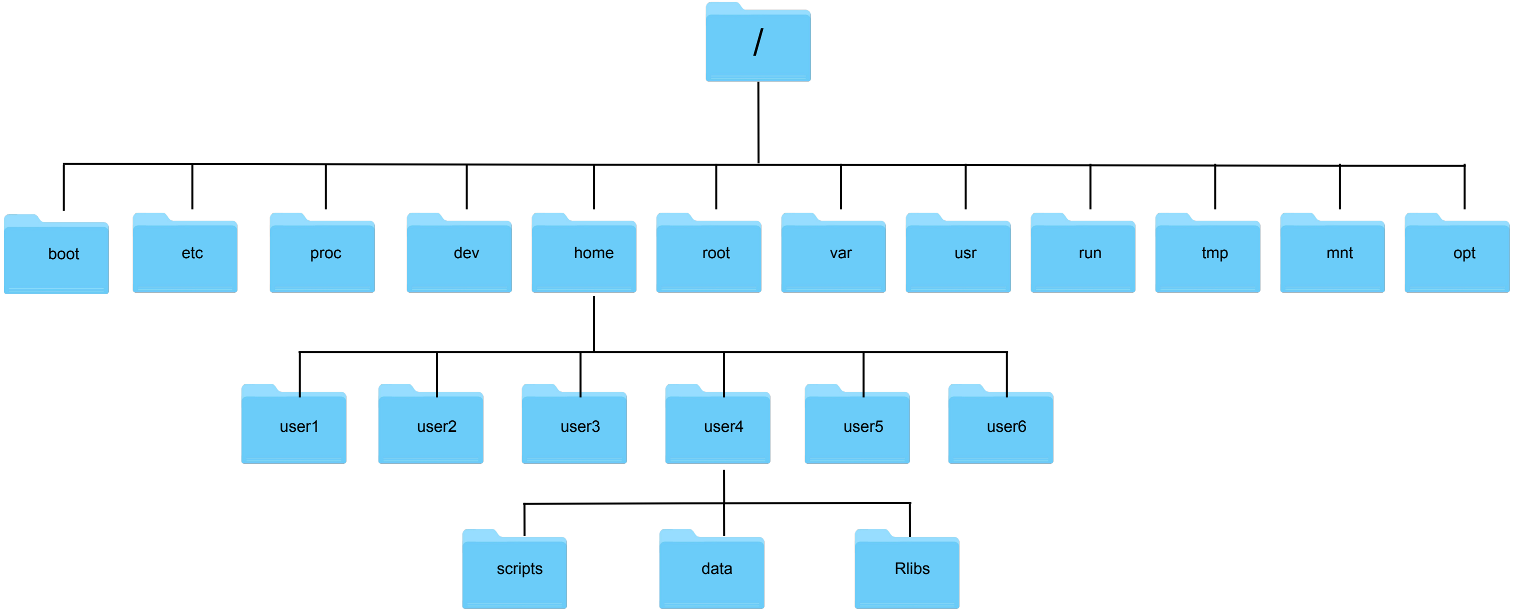
Pitfalls in Linux commands

- Case sensitivity(`ls` vs `Ls`)
- Missing spaces, quotes
- Attention to detail (`ls --l` vs `ls -l`)
- Window-ism (\ vs /)

File System hierarchy

- A hierarchical organization of files & folders(a.k.a directory)
  - similar to an upside-down tree where the top of the directory structure is called the ‘root’
- File system paths are used to specify the location of a file or folder
  - absolute path: specify the location of a file from the root directory
    - e.g. `/home/user1/data/millet`
  - relative path: specify the location a file/folder in relation to the current working directory
    - e.g. `data/millet`

File System hierarchy



Navigating the File System

- Common commands used to navigate & manipulate the file system’s directory structure:
  - `pwd` — **p**rint **w**orking **d**irectory (“*where am I?*”)
  - `ls` — **lis**t contents of the current directory
  - `cd` — **c**hange **d**irectory
  - `mkdir` — **m**a**k**e **dir**ectory

Navigating the File System

- Let’s create a few directories & navigate the file system structure
  - `mkdir earth`
  - `cd earth`
  - `mkdir -p continents/africa`
  - `ls`
  - `mkdir oceans`
  - `cd continents`
- So which folder are we in?
  - `pwd`
- How do we get to ‘oceans’ folder?

Navigating the File System

- To move to 'oceans' folder we have to first navigate up the hierarchy to 'earth' folder then navigate down the hierarchy to 'oceans'
  - `cd ..`
  - `cd oceans`
- Alternatively, you could combine both paths i.e .. + oceans ➔../oceans
  - `cd ../oceans`

Special Directories

- Special characters which represents ‘special’ directories
  - `.` current working directory i.e. ‘*here*’
    - `ls .`
    - `cd .`
  - `..` parent directory i.e. on level up the tree
    - `ls ..`
    - `cd ..`
  - `~` your home folder e.g. /home/user25

Working with Text Files

- Text editing in the CLI is a bit tricky since there are no menus or buttons to click on
  - cursor movement is done by using directional keys or key combinations; not by a mouse click
- There are several CLI-based text editors in Linux, popular ones being:
  - `nano`
  - `vim`
  - `emacs`
- Navigate to your home directory and use the command nano to create a new file called ‘file1’
  - `cd ~`
  - `nano file1`

Working with Text Files

- View the contents of the new file you just created:
  - `cat file1`
  - `wc -l file1`
  - `less file1`
    - Press ‘`q`’ to quit less
- ‘`cat`’ prints the contents of a file to the screen
- ‘`less`’ interactively shows the contents of a file one page at a time. Programs like less are called pagers

Questions?