

Introduction to Linux

Manpreet S. Katari

What is Linux?

- It is a free and open source operating system released in 1991 under the GNU GPL license.
 - GPL allows anyone to use, modify and redistribute with the requirement that they pass it on with the same license.
- It is the leading operating system of choice for servers such as supercomputers.
 - More than 90% of the top 500 fastest computers are based on Linux.

□

Relation to other operating systems

- ❑ MAC computers are related to Linux because they are also based on UNIX
- ❑ Another popular distribution of Linux operating system is called Ubuntu.
- ❑ Windows 10 now supports WSL (Windows Subsystem Linux).
 - ❑ It is a virtual machine that runs linux alongside windows.
- ❑ Depending on the purpose of the Linux machine, it may or may not have a Desktop environment that we are familiar with on our personal computers.
 - ❑ Linux uses X Window System to provide the Desktop environment.



Why do bioinformaticians use Linux?

- Many bioinformatics core tools are written in Linux
 - BLAST, CLUSTALW, PHRAP, etc
 - Many web applications are also supported on web servers hosted on linux machines
- Linux supports development of software for many different programming languages.
 - Developers are lazy so creating a software that does not require a window is much faster and easier
- Multiple users can log in at the same time
 - A user logging in over the network can do just about anything a user sitting in front of the computer can do.
 - Which means it can multitask.



The Linux Shell

- The standard user interface for personal computers is a GUI (Graphical User Interface).
- However for linux it is a command-line interpreter called shell.
- It is simply a prompt the awaits your command.
- There are several different shells, but the one used often is called “bash”, which is a mixture of a bunch of other shells.



Logging in: `ssh`

- ❑ To connect to a linux machine we will use a program called secure shell.
- ❑ If you have a MAC computer, simply open up Terminal and type the command below.
- ❑ If you have a Windows machines, please download PUTTY
 - ❑ (<http://www.putty.org/>) .
- ❑ The information that is displayed and the how the command line prompt appears has been set by the system administrator as default. There are ways to configure them if you want.

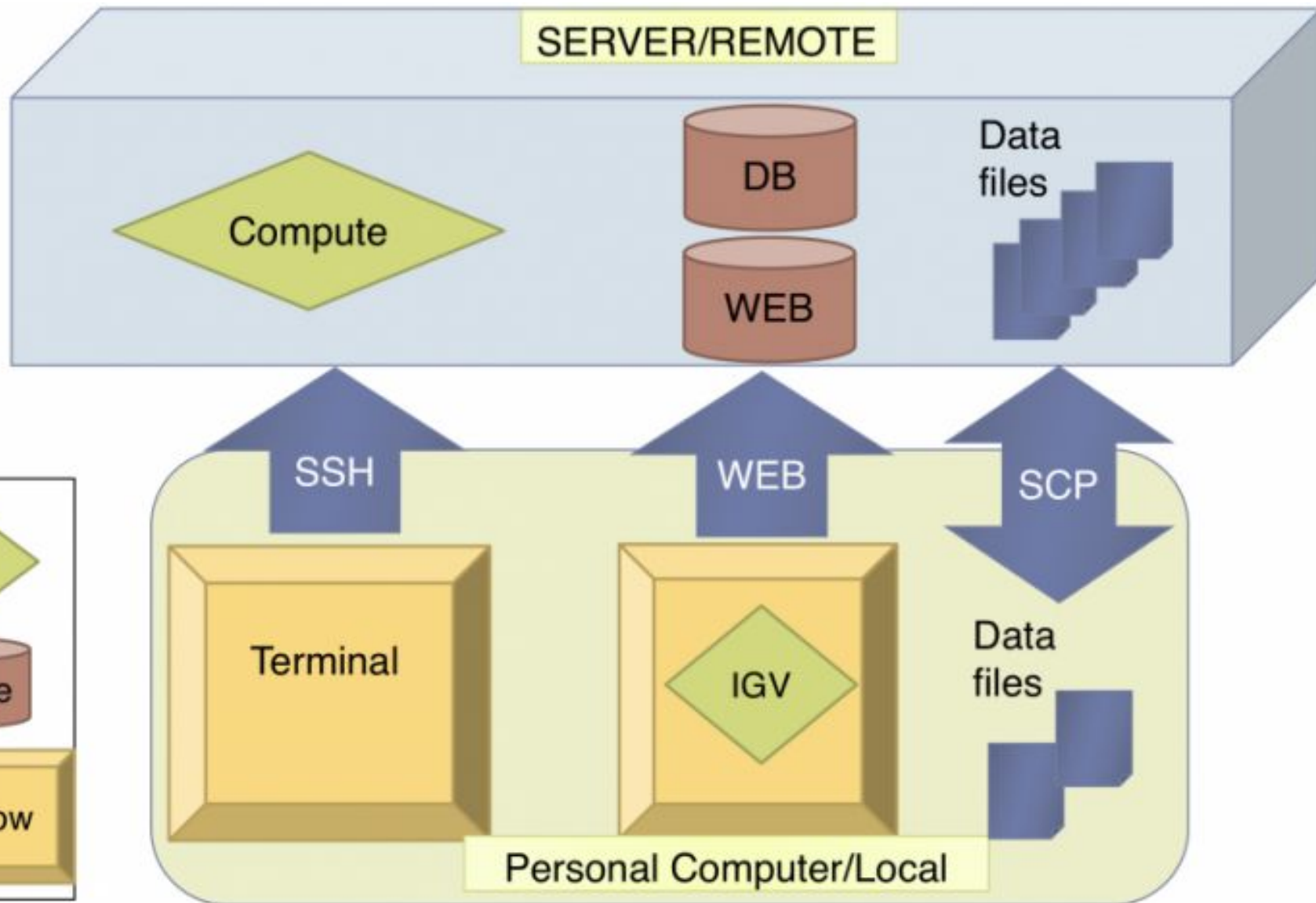


Logging in with X Windows

- In cases where a program requires a GUI, you should log in using the `-X` option.
- On a MAC, `-Y` option works better.
- This opens a tunnel to your computer allowing all windows to open in your computer.
- For this to work you need X11 installed on your computer.
 - MAC – Xquartz (<http://xquartz.macosforge.org/landing/>)
 - Windows – Xming (<http://sourceforge.net/projects/xming/>)

```
172-16-140-235:~ manpreetkatari$ssh -X mkatari@hpc.ilri.cgiar.org
mkatari@hpc.ilri.cgiar.org's password:
Last login: Mon Dec 18 12:47:18 2017 from 172-17-11-36.dynapool.nyu.edu
[mkatari@hpc ~]$
```

Terminology



Issuing commands

- The command prompt requires that you enter the command followed by arguments (if necessary)
- If there is an output to the program it usually prints it on the screen (often referred to as the standard output)

```
[mkatari@hpc ~]$ date
Tue Aug 10 07:19:40 EAT 2021
[mkatari@hpc ~]$ ls -l /home/
total 1504
drwxr-xr-x.  4 aache          aache          4096 Mar 31 08:30 aache
drwx-----.  4 aamane          aamane          4096 Jun 10  2020 aamane
drwxr-xr-x. 21 abirindwa       abirindwa       4096 Jun  3 12:05 abirindwa
drwx-----.  4 abombom          abombom          4096 Jul  6  2019 abombom
drwxr-xr-x.  6 acdarby          acdarby          4096 Jun 10  2020 acdarby
drwx-----.  9 adgg              adgg              4096 Jun 15 10:51 adgg
drwx-----x. 25 adiddi            adiddi            4096 Jun  5  2020 adiddi
drwx-----.  5 afalcucci          afalcucci          4096 Jul 12  2019 afalcucci
drwxr-xr-x. 48 afeleke          afeleke          4096 Jul 19 10:18 afeleke
drwxr-xr-x. 30 agari            agari            4096 Jul 22 15:08 agari
drwx-----.  8 aghouila          aghouila          4096 Jun 10  2020 aghouila
drwxr-xr-x.  7 agisel            agisel            4096 Jul  4  2019 agisel
```

Directing the standard output

- Instead letting the output print to the screen we can save it to a file by using the `>` sign and then giving the file name. This will replace a file if it already exists without a warning. To append use `>>`

```
[mkatari@hpc ~]$ date[mkatari@hpc ~]$ ls -l /home/ > allusers.txt
[mkatari@hpc ~]$ ls -l
total 12821
-rw-rw-r--.  1 mkatari mkatari   28730 Aug 10 07:23 allusers.txt
```

- The output can also be redirected to be the input to another command by using `|`

```
[mkatari@hpc ~]$ ls -l /home/ | tail
drwxr-xr-x.  48 vchebii          vchebii          4096 Jul 30 17:41 vchebii
drwx-----.  11 vhabimana      vhabimana      4096 Jul  6  2019 vhabimana
drwx-----.   3 visendi        visendi        4096 Jul 12  2019 visendi
drwx-----.   4 vnjunge        vnjunge        4096 Jun 13  2020 vnjunge
drwx-----.   6 wkarubiu       wkarubiu       4096 Jul  7  2019 wkarubiu
drwx-----.  13 wkimani        wkimani        4096 Aug  3 16:50 wkimani
drwx-----.   6 wsharpee      wsharpee      4096 Jun 13  2020 wsharpee
drwxr-xr-x.   4 wwint          wwint          4096 Jun 13  2020 wwint
drwxr-xr-x.  11 yajamma        yajamma        4096 Jul 10  2019 yajamma
drwx--x--x.  25 vtchiechoua    vtchiechoua    4096 Aug  6 14:39
```

Command line editing

- The command is only executed once you press enter.
- Till then you can edit the line by using the following key strokes:

- | | |
|-----------------------------|----------------------------------|
| – Backspace (DELETE on MAC) | = delete previous character |
| – Left arrow, right arrow | = move left and right on line |
| – Control-A | = go to front of line |
| – Control-E | = go to end of line |
| – Control-D | = delete next character |
| – Control-K | = delete everything on the right |
| – Control-Y | = paste |
| – Up-arrow, down-arrow | = previous and next command. |



Getting Help

- Majority of the commands have a [manual](#) entry for documenting its use.
- They include all the different options that are available but also gives examples of the most common scenarios.

```
[mkatari@hpc ~]$ man sort
```

```
SORT(1)
```

User Commands

```
SORT(1)
```

NAME

```
sort - sort lines of text files
```

SYNOPSIS

```
sort [OPTION]... [FILE]...
```

```
sort [OPTION]... --files0-from=F
```

DESCRIPTION

```
Write sorted concatenation of all FILE(s) to standard output.
```

```
Mandatory arguments to long options are mandatory for short options
```

Command-line Completion

- ❑ In some cases the commands or the file names that you need as arguments can be very long which increases the chance of spelling mistakes.
- ❑ To prevent such mistakes simply type the enough letters to unambiguously identify the command or file and then pressing tab will complete it for you.
- ❑ In the case you don't know how many letters you need, simply press tab twice to see all your options.

```
[mkatari@hpc ~]$ bl  
blkdeactivate  blkdiscard      blkid              blockdev
```

Wild Card

- In cases where you want to refer to multiple files you can use `*` to represent any characters of any length.
- You can also use `?` To represent any character of one length.

```
[mkatari@hpc ~]$ ls /usr/bin/b*
/usr/bin/base64          /usr/bin/bdftopcf      /usr/bin/build-classpath
/usr/bin/bzgrep
/usr/bin/basename       /usr/bin/bdftruncate
/usr/bin/build-classpath-directory /usr/bin/bzip2
/usr/bin/bash           /usr/bin/bg            /usr/bin/build-jar-repository
/usr/bin/bzip2recover
/usr/bin/bashbug        /usr/bin/bjobs         /usr/bin/bunzip2
/usr/bin/bzless
/usr/bin/bashbug-64     /usr/bin/bkill         /usr/bin/busctl
/usr/bin/bzmore
/usr/bin/bc             /usr/bin/bond2team     /usr/bin/bzcat
/usr/bin/bcomps         /usr/bin/bootctl       /usr/bin/bzcmp
/usr/bin/bdf2gdfont.pl  /usr/bin/bsub          /usr/bin/bzdiff
```

Home sweet home

- When you first log in, you will be in a directory called “home directory”
 - `/home/<your username>`
- Generally in this directory you have complete control over creating, modifying, and executing files in this or any sub directory you create.
- In order to return to your home directory simply type the command: “`cd ~`” at the prompt.
- Unless appropriate changes have been made you can not enter anyone’s directory or even see what is in it.



Some helpful commands

- Who am I ?
- In which directory am I ?
- On which server am I ?

```
[mkatari@hpc ~]$ whoami
mkatari
[mkatari@hpc ~]$ pwd
/home/mkatari
[mkatari@hpc ~]$ hostname
hpc.ilri.cgiar.org
```



Commands for file management.

- A path is the directory structure of the file system used to define the location of a file or directory.
 - The path to my home directory is /home/mkatari
- **mkdir** – making a directory (can not have two directories with same name)
- **cd** – changing into a directory
- **pwd** – the directory you are in presently
- **ls** – list the contents of the directory
- **cp** – copy a file (requires two arguments- source and target)
 - In order to copy a directory you must use **cp -r**
- **mv** – move a file (copy and remove the source)
- **rm** – remove/delete a file.
 - To recursively remove all directories and files in a directory use **rm -r**
- **rmdir** – removing a directory (only works if the directory is empty)



```
[mkatari@hpc ~]$ mkdir temp
[mkatari@hpc ~]$ cd temp
[mkatari@hpc temp]$ pwd
/home/mkatari/temp
[mkatari@hpc temp]$ ls
[mkatari@hpc temp]$ cp ../allusers.txt
cp: missing destination file operand after `../allusers.txt'
Try 'cp --help' for more information.
[mkatari@hpc temp]$ cp ../allusers.txt ./
[mkatari@hpc temp]$ ls
allusers.txt
[mkatari@hpc temp]$ mv allusers.txt allusers.bkup
[mkatari@hpc temp]$ rm allusers.bkup
[mkatari@hpc temp]$ cd ..
[mkatari@hpc ~]$ rmdir temp/
```



Commands for file manipulation

- ❑ **less/more** – read through the file without loading the entire file. Press **spacebar** to continue or **q** to quit.
- ❑ **touch** – create an empty file
- ❑ **head** – show the first few lines of the file
- ❑ **tail** – show the last few lines of the file
- ❑ **cat** – read through the file(s)
- ❑ **grep** – search for patterns in a file or files
- ❑ **cut** – separate file based on columns
- ❑ **comm/diff** – compare and see the difference between the files. The files have to be sorted before using either of these commands.
- ❑ **split** – splits file into smaller files based on the options
- ❑ **sort** – sort the file based on the options selected.
- ❑ **wc** - wordcount



```
[mkatari@hpc ~]$ touch just_top_bottom_users.txt
[mkatari@hpc ~]$ head allusers.txt > topusers
[mkatari@hpc ~]$ tail allusers.txt > bottomusers
[mkatari@hpc ~]$ cat topusers bottomusers >>
just_top_bottom_users.txt
```



Your command just made history

- Your shell saves all your commands and you can access them using the up and down keys.
- Typing the command “**history**” returns all the commands you have entered and a number assigned to it.
- You can run a specific one again by typing **!** (also called “bang”) Followed by the job number.
- **!!** Will perform the most recent command.



Linux Permissions

- There are three levels of permissions that can be assigned to all files, programs, and directories
 - Read: open the file and copy it
 - Write: edit the file and delete it
 - Execute: Run the commands in the file or change into the directory if it is a directory
- There are also three different levels of users:
 - User – you
 - Group – A collection of users that are in a group
 - Everyone - Not just the people who have accounts on the machine but if the directory is open to the public and any one.



Changing permissions

- **chown** – changes the owner
- **chgrp** – changes the group
- **chmod** – change read, write, and execute permissions
 - **+/- r** = read
 - **+/- w** = write
 - **+/- x** = execute
 - **u** = user level
 - **g** = group level
 - **o** = others
 - **a** = all
- **chmod** can also use three numbers to set permissions where the value of the number represents a specific combination of rwx and their order assigns it to the different levels (u,g,o)



Transferring files

- Moving files from one server to another use the scp command. This is similar to ssh only you also provide the files you want to move.

```
scp README.md mkatari@hpc.ilri.cgiar.org:.
```

- Moving files to and from your computer you can use a SCP client:
 - Cyberduck
 - FileZilla



Controlling Jobs

- Suspend Jobs
 - `ctrl-z`
- Send Jobs in background
 - `emacs test.txt &`
 - Or if job is already suspended, just type `bg`
- To see all jobs and the job number
 - `jobs`
- To put jobs in foreground
 - `fg`
- In case there are multiple jobs in background
 - `fg %<job number>`
- Looking at all jobs running on computer
 - `top`
- One can delete/cancel jobs from top by typing
 - `k` and then `processid`



Logging out

- Simply type **logout** or **exit** to end your session.



Resources at Gencore

<https://learn.gencore.bio.nyu.edu/pre-requisites/introduction-to-linux/>

