Basic Linux for HPC

Session Outcomes

- Understand the difference between personal computers and HPC.
- Understand some basic components in computer system.
- Understand the concept within HPC environment.
- Understand the difference between Linux and other Operating System.
- Understand the different commands used in Linux to perform different tasks.

Software Required for This Sessions

Operating System	System Built-in	External Installation
Windows OS	`cmd` `powershell`	PuTTY MobaXTerm
Linux	`terminal`	
macOS	`terminal`	iTerm2

What is HPC?

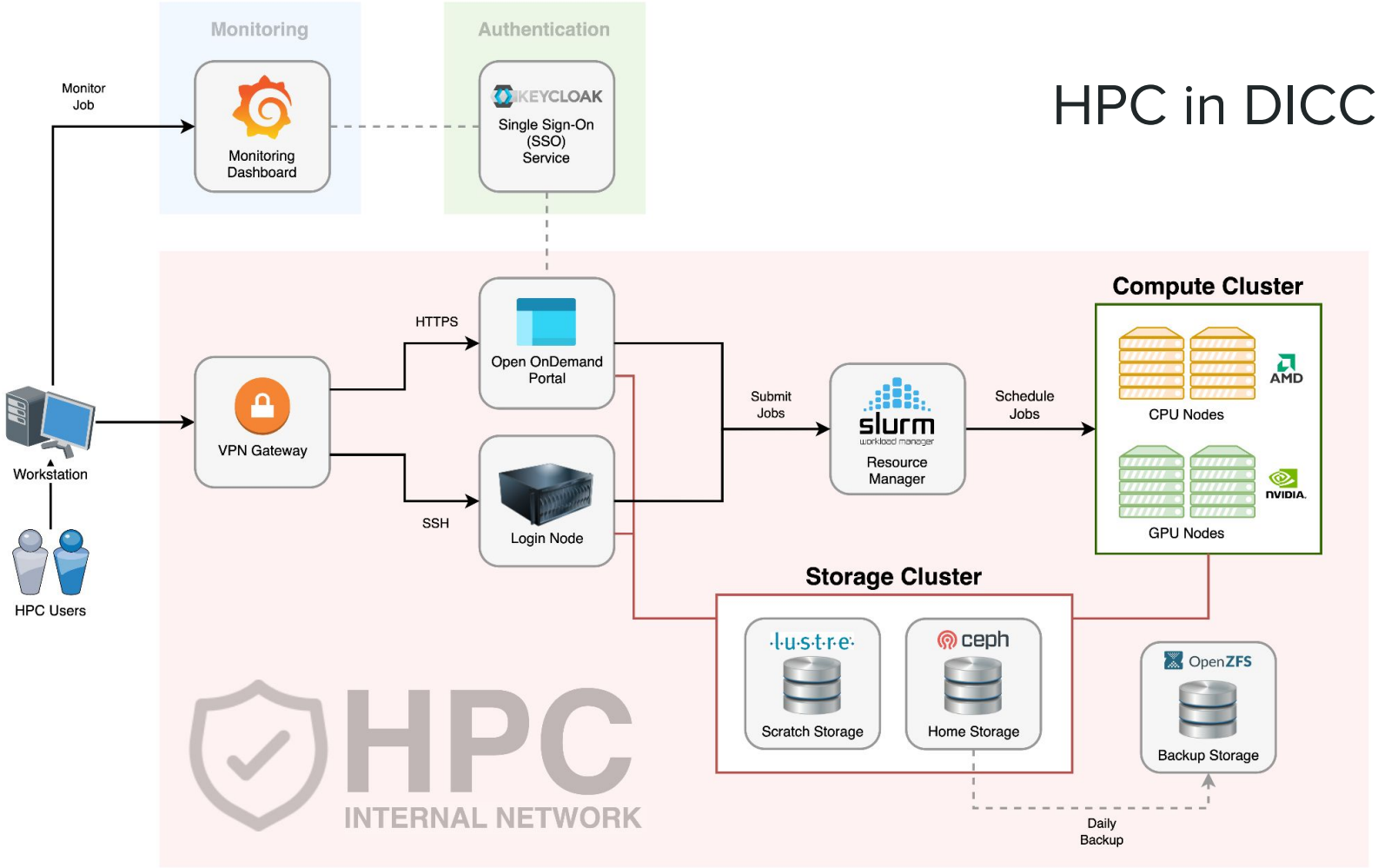
Workstation/Desktop/Laptop

Portable, affordable but limited computing power.

HPC Cluster

More investments and costs, but can be much more powerful.

HPC in DICC


Monitoring
Authentication
Monitor Job
Monitoring Dashboard
KEYCLOAK
Single Sign-On (SSO) Service
HTTPS
Open OnDemand Portal
Compute Cluster
Workstation
VPN Gateway
Submit Jobs
slurm
workload manager
Resource Manager
Schedule Jobs
AMD
CPU Nodes
NVIDIA
GPU Nodes
SSH
Login Node
HPC Users
Storage Cluster
·l·u·s·t·r·e·
Scratch Storage
ceph
Home Storage
OpenZFS
Backup Storage
HPC
INTERNAL NETWORK
Daily Backup

Why HPC?

Why do people use HPC?

- **Highly parallel calculation**
  - Can be splitted into multiple small calculations and execute concurrently.
- **Large-scale tightly coupled calculation**
  - Calculation require resources that beyond what a workstation or laptop can supply.
- **Computation require use of GPU**
  - Proven to be able to utilise GPU for massive speedup.

However,

HPC is not the magic solution for everything.

What must you know?

- Basic Application Understanding
  - To be able to run and execute your application in the HPC.
- Basic Computer System Understanding
  - To understand the resources type in the HPC.
- Basic Linux Survival Skills
  - Must have basic Linux knowledge to survive in the HPC environment.

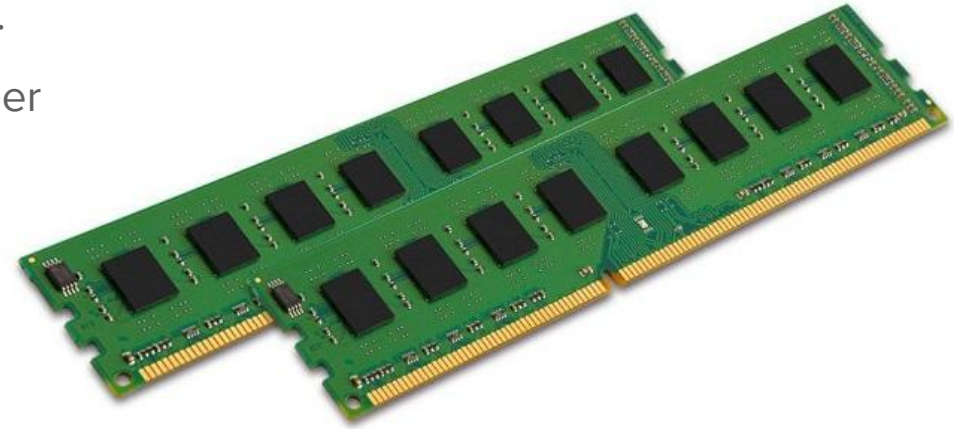
Computer System in Layman

Processor in Layman

- Processor is the brain of any computer system.
  - Core - The processing core in the processor
  - Threads - Number of threads per core
  - Clock Speed - Number of CPU execution per second
- Consumer Grade Processors:
  - Intel Core i7-1365U (10 Cores) - 5.20 GHz
  - AMD Ryzen 5 5600 (6 Cores) - 4.4 GHz
- Server Grade Processors:
  - Intel Xeon Platinum 8480+ (56 Cores) - 3.80 GHz
  - AMC EPYC 7702P (64 Cores) - 3.35 GHz

Memory in Layman

- Memory is the place where data required for CPU processing is stored.
- Common Memory Size: 1GB - 128GB per memory
- Memory Type: DDR1 - DDR5
- Newer type has lower latency, which mean faster access.
- Larger memory mean more stuff can run concurrently, and larger calculations can be supported.



GPU in Layman

- Super powerful processing unit that can dramatically accelerate additional workloads in high performance computing.
- Usually very expensive.
- Good for graphical processing, AI, accelerated mathematics calculations, and more!
- Example NVIDIA Models:
  - RTX 4090
  - A100
  - V100
  - H100

Storage in Layman

- The location where your files and directories are stored.
- Local Storage:
  - HDD
  - Hybrid-HDD
  - SSD
  - SAS
  - NL-SAS
- Network Storage:
  - NFS
  - Lustre
  - Ceph
  - GlusterFS

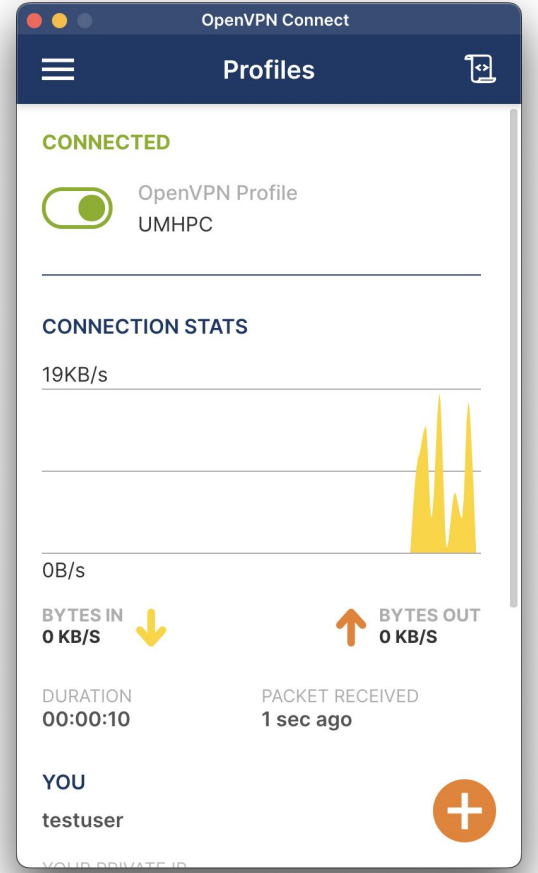
Accessing HPC Login Node

DICC Account

- DICC SSO (sso.dicc.um.edu.my)
  - Update password at DICC SSO.
  - If you forgotten your password, you can also reset your password at DICC SSO.
- Request HPC access at Service Desk.
- DO NOT SHARE YOUR ACCOUNT !!

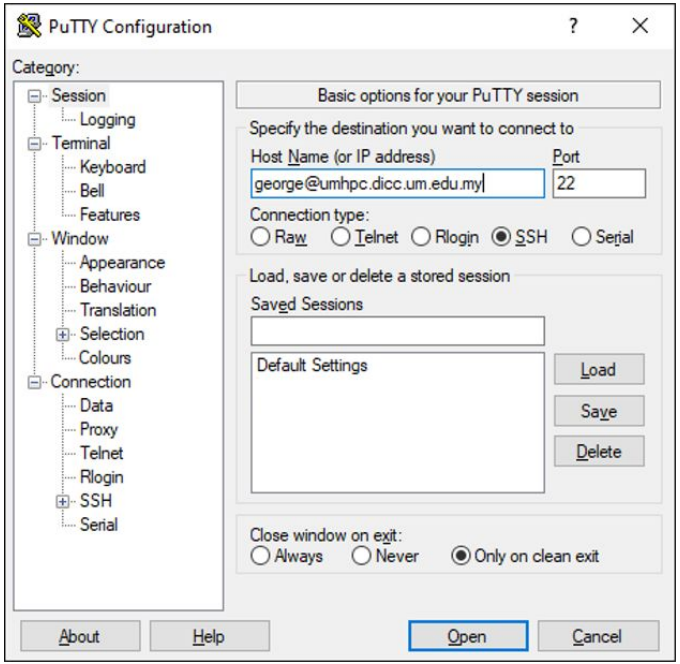
VPN Connection

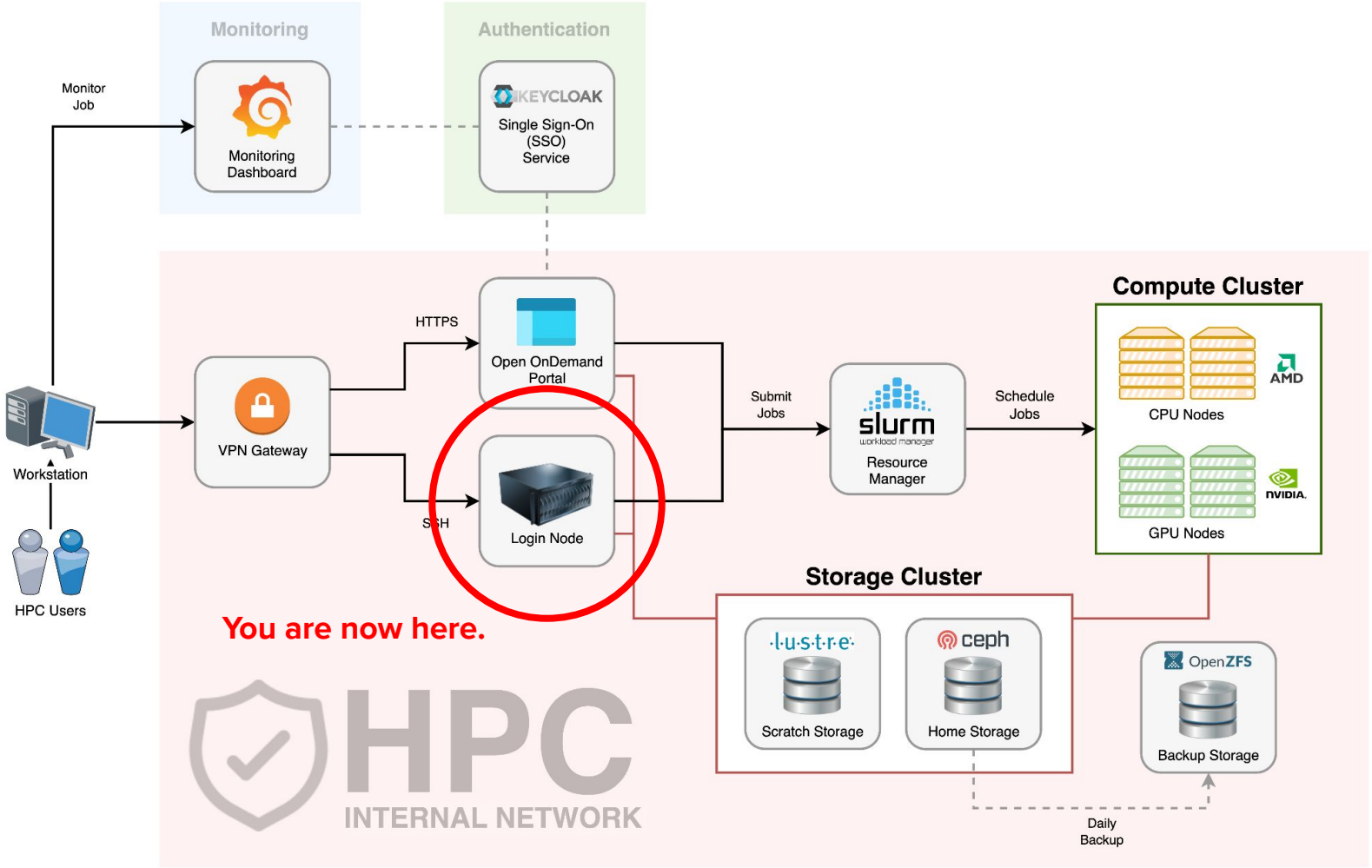
- Only account with HPC access can establish connection with the VPN gateway.
- Required software:
  - OpenVPN connect client
  - OpenVPN profile
- VPN Gateway:
  - `vpn01.dicc.um.edu.my`
  - `vpn02.dicc.um.edu.my`



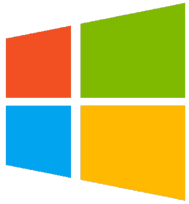
Connecting to HPC Login Node

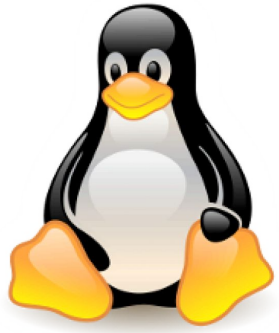
- Windows users:
  - PuTTY / MobaXTerm
- Linux / Mac OS users:
  - Use **`ssh`** command
  - **`ssh username@umhpc.dicc.um.edu.my`**
- Connection details:
  - Hostname: **`umhpc.dicc.um.edu.my`**
  - Port: **`22`**
  - Connection Type: **`SSH`**



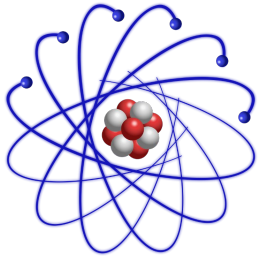
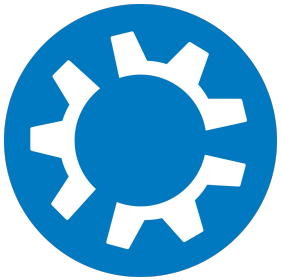

Monitoring
Authentication
Monitor Job
Monitoring Dashboard
KEYCLOAK
Single Sign-On (SSO) Service
HTTPS
Open OnDemand Portal
Compute Cluster
Workstation
VPN Gateway
SSH
Login Node
Submit Jobs
slurm
workload manager
Resource Manager
Schedule Jobs
AMD
CPU Nodes
NVIDIA
GPU Nodes
HPC Users
You are now here.
Storage Cluster
·l·u·s·t·r·e·
Scratch Storage
ceph
Home Storage
OpenZFS
Backup Storage
HPC
INTERNAL NETWORK
Daily Backup

Introduction to Linux






Linux-Based HPC

- CLI & terminal based usage and environment.
- HPC is not easy, but can be easy with practise.

What is Linux?

- Open source UNIX-like operating system.
- Many distributions and flavours:
  - Fedora
    - RedHat, CentOS, Rocky Linux
  - Debian
    - Ubuntu, Kubuntu
  - SUSE
    - SLES, OpenSUSE
- Widely used in server environments where performance matter.

Fedora Linux in DICC

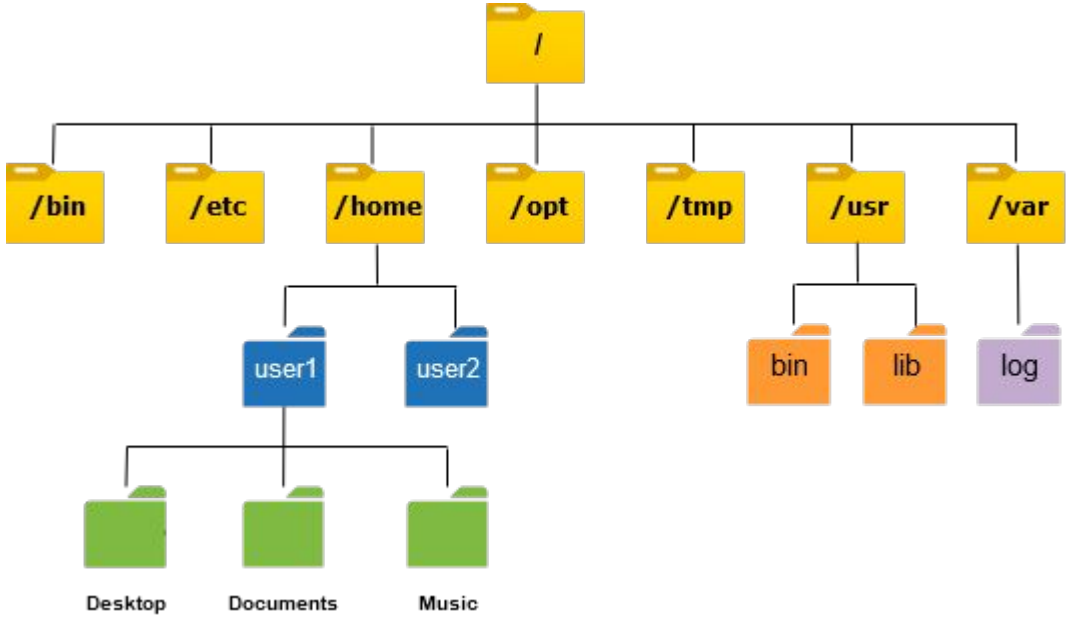
- Free, open source
- Reliable
- Lightweight
- Allow multiple concurrent connections

User Roles in Linux

- Super Users
  - System Administrator that can access everything on the system.
- Regular Users
  - Can only access files and directories owned by themselves.
  - All HPC users belongs to this group.
- Service Users
  - System users that are used to run system services.

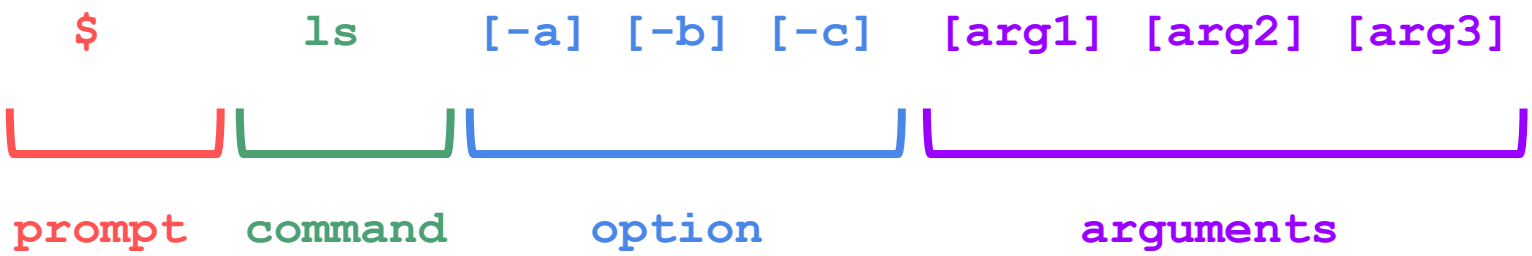
Directories in Linux

- Tree-like directory structure.
- Everything start with root directory.
  - `/home/user/`
  - `/opt/app/exe/`
  - `/tmp/scratch/`
  - `/dev/usb1/`
- No C/D/E drives like Windows



Linux Commands

Linux Command Structure



- `ls -lah /home/user/george`
- `cd /tmp`
- `df`

Basic Linux Commands

- Instructions to perform basic actions in Linux.
  - Copy file
  - Move file
  - List directory
  - Navigate to another directory
  - Remove file or directory
  - Create new directory
  - Search for file or directory
- `Ctrl + C` to cancel instruction.

Linux command	Description	Linux command example
cd	Change directory with a specified path	cd */path/directory1*
clear	Clear the screen	clear
cp	Copy file(s)	cp */path1/file1 /path2/file1*
diff	Compare the contents of files	diff *file1 file2*
exit	Log out of Linux	exit
grep	Find a string of text in a file	grep "word or phrase" *file1*
head	Display beginning of a file	head *file1*
less	View a file	less *file1*
ls	List contents of a directory	ls */path/directory1*
mv	Move file(s) or rename file(s)	mv */path1/file1 /path2/file2*
mkdir	Create a directory	mkdir *directory*
rm	Delete file(s)	rm *file1*
rmdir	Remove a directory	rmdir *directory*
tail	Display end of a file	tail *file1*
tar	Store, list or extract files in an archive	tar *file1*
vi	Edit file(s) with simple text editor	vi *file1*

List Directory Contents (ls)

Usage: `ls <destination>`

✓ `ls /home/user/george`

✓ `ls /opt`

✓ `ls /dev`

✓ `ls /`

✗ `ls /root`

Try out:

```
ls -l    <destination>
ls -la   <destination>
ls -lah  <destination>
ll       <destination>
```

Navigate to Another Directory (cd)

Usage: **`cd <destination>`**

✓ `cd /home/user/george`

✓ `cd /`

✗ `cd /root`

✓ `cd /opt`

✓ `cd`

Print Working Directory (pwd)

Usage: **`pwd`**

✓ **`pwd`**

Create Directory (mkdir)

Usage: **`mkdir <directory>`**

✓ **`mkdir test`**

✓ **`mkdir /home/user/george/test`**

X **`mkdir /root/test`**

X **`mkdir /test`**

✓ **`mkdir /home/user/george/abc.txt`**

Create Empty File (touch)

Usage: **`touch <filename>`**

✓ `touch /home/user/george/example.sh`

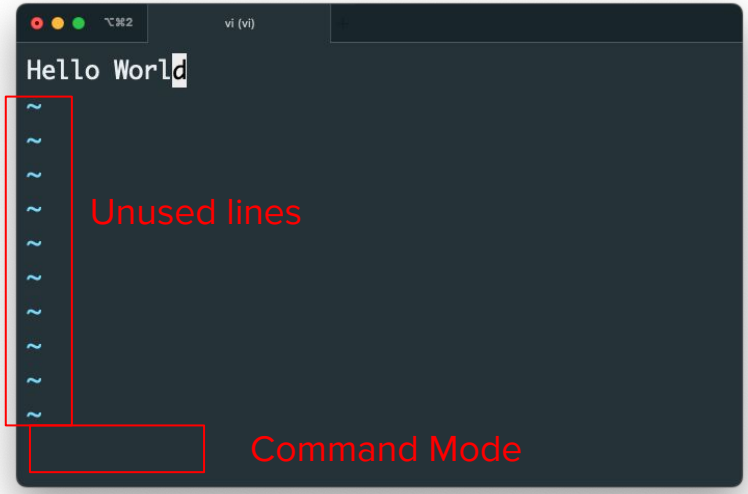
X `touch /root/abc.txt`

X `touch /abc.txt`

X `touch /home/user/george/test/`

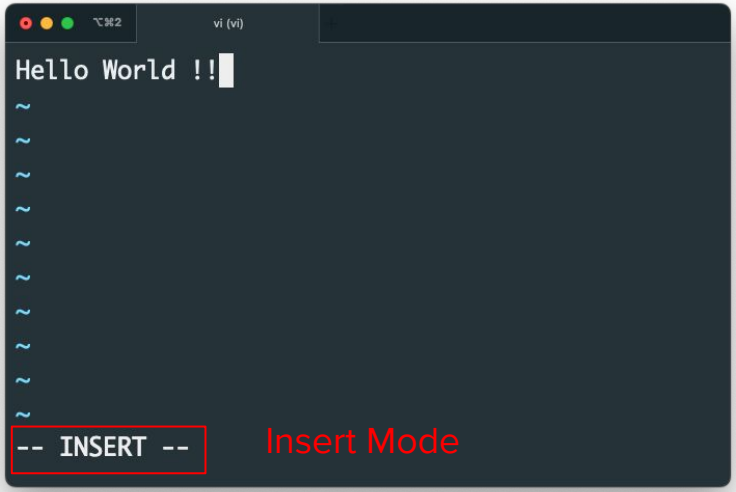
Linux Text Editor (vi)

- Command Mode (default):
  - **`arrow keys`** to navigate
  - **`u`** to undo action
  - **`dw`** to cut word
  - **`yy`** to copy the current line
  - **`dd`** to cut whole line
  - **`P`** to paste before your cursor
  - **`p`** to paste after your cursor
  - **`:w`** to save the file
  - **`:wq`** to save the file and quit
  - **`:q!`** to quit without saving



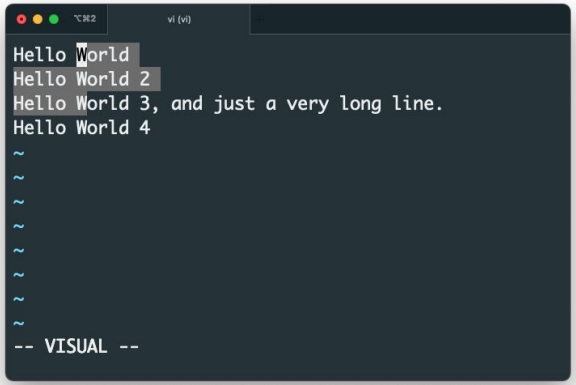
Linux Text Editor (vi)

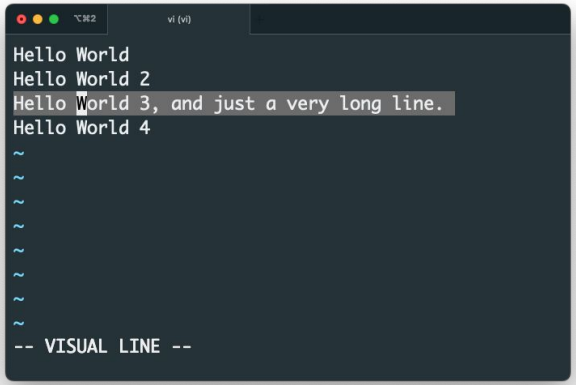
- Insert Mode:
  - **`i`** to enter Insert Mode
  - **`esc`** to leave Insert Mode and enter Command Mode
  - Can type anything in Insert Mode



Linux Text Editor (vi)

- Visual Mode:
  - **`v`** to enter visual mode
  - **`Shift + v`** to enter visual line mode
  - **`Ctrl + v`** to enter visual block mode





Create/Edit File Content (vi)

Usage: **`vi <filename>`**

✓ **`vi /home/user/george/abc.txt`**

X **`vi /home/user/george/test`**

✓ **`vi /tmp/test.txt`**

X **`vi /root/abc.txt`**

Linux WYSIWYG Text Editor (nano)

- WYSIWYG - What you see is what you get
- Commands:
  - **Ctrl + x** exit
  - **Ctrl + o** save
  - **Ctrl + w** search
  - **Ctrl + k** cut
  - **Ctrl + u** paste
  - **Ctrl + g** help menu

Create/Edit File Content (nano)

Usage: **`nano <filename>`**

✓ **`nano /home/user/george/abc.txt`**

✗ **`nano /home/user/george/test/`**

✓ **`nano /tmp/test.txt`**

✗ **`nano /root/abc.txt`**

Print File Content (cat)

Usage: **`cat <filename>`**

✓ **`cat /home/user/george/test.txt`**

X **`cat /home/user/george`**

X **`cat /abc.txt`**

X **`cat /root/abc.txt`**

Read File Content in Scrollable Mode (less)

Usage: **`less <filename>`**

✓ **`less /home/user/george/test.txt`**

X **`less /home/user/george`**

X **`less /abc.txt`**

X **`less /root/abc.txt`**

Simple Exercise (Part 01)

Let’s do it

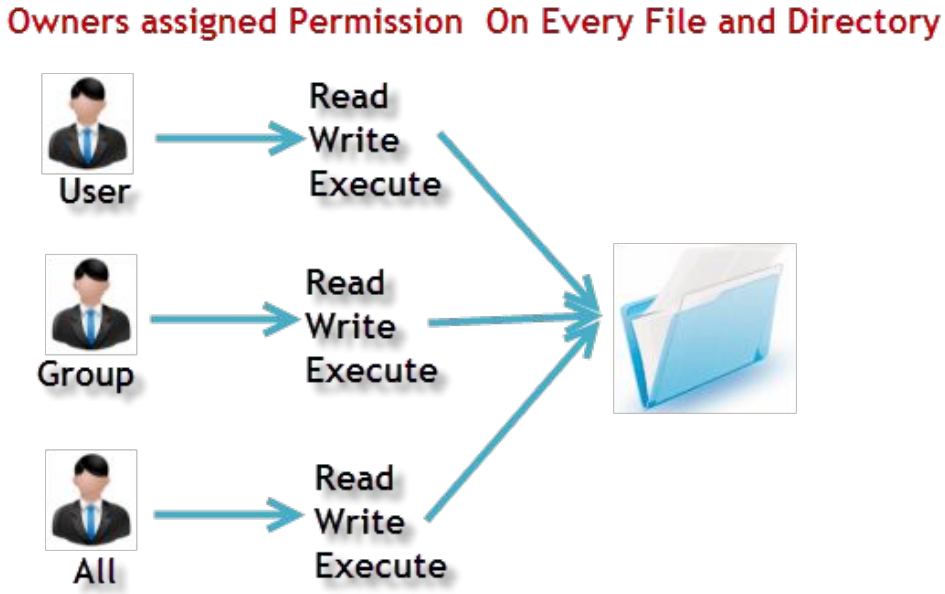
- Create new directory **training01** in your home directory.
- Navigate to the created directory.
- Create a file named **data.txt** with following content:
    - ```
      I have some sample data.
      I have more sample data.
      I have even more sample data.
      ```
- Check the content of current directory and ensure **data.txt** present.
- Print out the content of the **data.txt** and verify.
      ```

Let’s do it Together (Answer)

- `mkdir ~/training01`
- `cd ~/training01`
- `nano data.txt or vi data.txt`
- `ls -l`
- `cat data.txt or less data.txt`

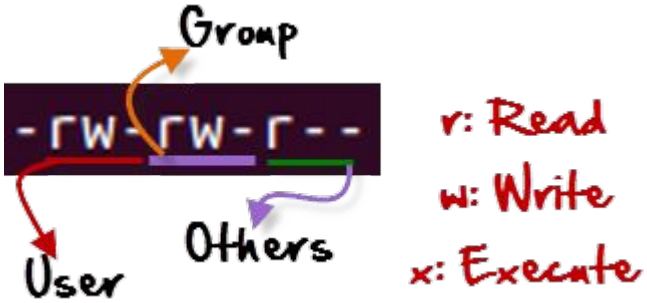
File Permissions in Linux

- **File Ownership**
  - User
  - Group
  - All
- **File Permissions**
  - Read
  - Write
  - Execute



Example Ownerships & Permissions

- John is a member of alpha.
- Can John read the content of the files with following permissions?
  - `rwx --- --- john john 1.sh`
  - `rw- r-- r-- george alpha 2.sh`
  - `rwx rwx rwx william beta 3.sh`
  - `rwx --- r-- george alpha 4.sh`
  - `--- --- --- john john 5.sh`



Important !!

DO NOT

set your home directory or scratch directory to permission 777 !!

Change File/Directory Permission (chmod)

Usage: **`chmod <permission> <filename>`**

✓ `chmod +x /home/user/george/example.sh`

✓ `chmod 700 /home/user/george/example.sh`

✓ `chmod u=rwx,g=-,o=- /home/user/george/example.sh`

✗ `chmod /root/test.txt`

✓ `chmod 644 /home/user/george/dir`

Change File/Directory Owner (chown)

Usage: **`chown [-R] <owner>:<group> <filename>`**

✓ **`chown george:george file.txt`**

✓ **`chown george:alpha file.txt`**

X **`chown george:root file.txt`**

✓ **`chown george:george dir`**

✓ **`chown -R george:george dir_with_files`**

Execute File/Script (source or .)

Do not require execute permission on file.

✓ `source example.sh`

✓ `source dir/example.sh`

X `source dir`

✓ `. example.sh`

✓ `. dir/example.sh`

Execute File/Script (Direct Run)

Require execute permission on file.

✓ `./example.sh`

✓ `dir/example.sh`

✗ `dir`

Create Link to File/Directory (ln)

Usage: `ln -s <target_file> <link_location>`

✓ `ln -s ~/test.txt ~/linkfile`

✓ `ln -s ~/test/ ~/linkdir`

X `ln -s /root/abc.txt ~/linkfile`

Copy File/Directory (cp)

Usage: `cp [-R] <source> <destination>`

✓ `cp file.txt copy_of_file.txt`

✓ `cp -R dir copy_of_dir`

✓ `cp dir/* dir2/`

✗ `cp dir dir2`

✓ `cp dir/file.txt dir2/copy_of_file.txt`

Rename File/Directory (mv)

Usage: **`mv <source> <destination>`**

- ✓ **`mv file.txt file2.txt`**
- ✓ **`mv dir dir2`**
- ✓ **`mv dir/* dir2/`**
- ✓ **`mv file.txt dir2/`**

Remove File/Directory (rm)

Usage: `rm [-r] <filename or directory>`

✓ `rm /home/user/george/test.txt`

X `rm /home/user/george`

X `rm /abc.txt`

X `rm /root/abc.txt`

✓ `rm -r /home/user/george/test`

Remove Directory (rmdir)

Usage: **`rmdir <directory>`**

✓ `rmdir /home/user/george/test/`

X `rmdir /home/user/george/test.txt`

X `rmdir /root/`

X `rmdir /opt/app`

Simple Exercise (Part 02)

Let’s do it Together

- Navigate to the **training01** directory created earlier.
- Create a script named **script.sh** with the following content.
  - ```
    #!/bin/sh

    echo Hello HPC!!
    echo I am now in $(pwd) directory.
    echo These are the contents from $1.
    cat $1
    ```
- Add executable permission to the **script.sh**.
- Execute **script.sh** with **data.txt** as argument.
- Remove the directory **training01**.
    ```

Let’s do it Together (Answer)

- `cd ~/training01`
- `nano script.sh or vi script.sh`
- `chmod +x script.sh`
- `./script.sh data.txt`
- `cd ~`
- `rm -r training01 or rm -rf training01`

Any Questions?