

# Introduction to Linux™

**Jason Li**

HPC User Services

LSU HPC / LONI

[sys-help@loni.org](mailto:sys-help@loni.org)

Louisiana State University, Baton Rouge

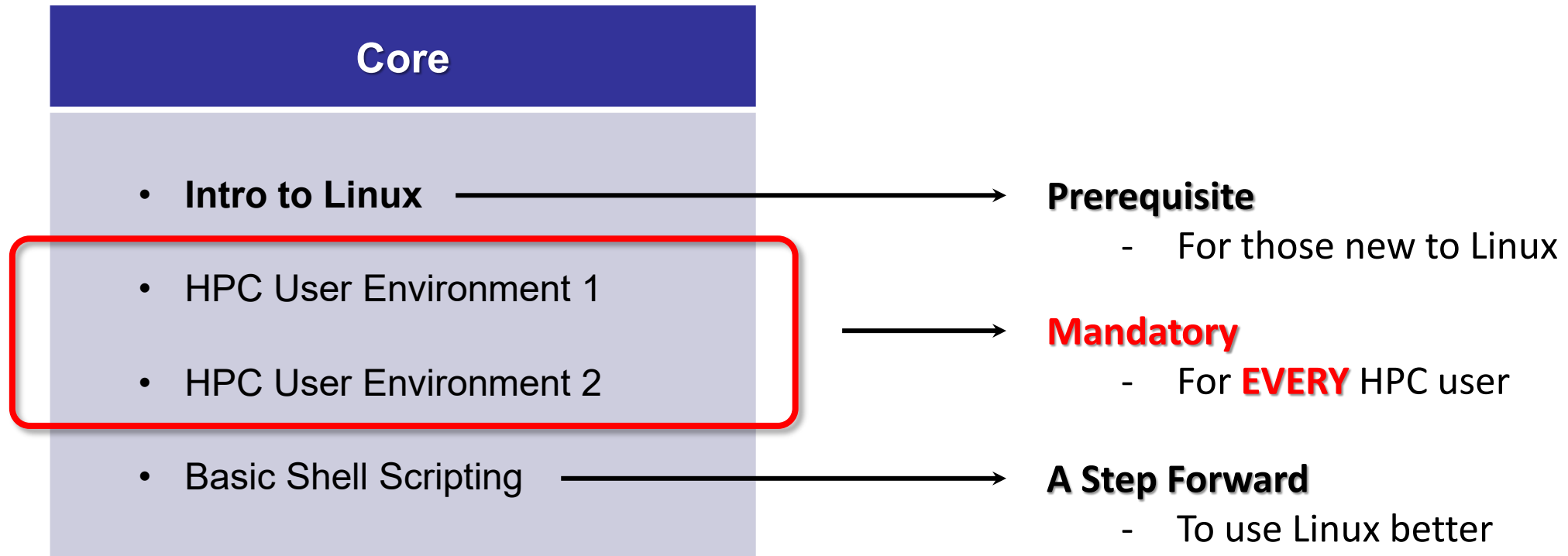
Jul 1, 2026

- Recordings will be available at:
  - LSU HPC: <https://www.hpc.lsu.edu/training/tutorials/past/>
  - LONI HPC: <https://hpc.loni.org/training/tutorials/past/>

- **Research Computing Trainings**

Core	Interested-based
<ul style="list-style-type: none"><li>• <b>Intro to Linux</b></li><li>• HPC User Environment 1</li><li>• HPC User Environment 2</li><li>• Basic Shell Scripting</li></ul>	<ul style="list-style-type: none"><li>• Open OnDemand</li><li>• Intro to Python</li><li>• Magic Tools 1: Singularity Container</li><li>• Magic Tools 2: Conda Environment</li><li>• GNU Parallel</li><li>• Molecular Dynamics</li><li>• ...</li></ul>

- Research Computing Trainings



- **Purpose of today's training...**

- To make you a Linux expert...



- To make you familiar with basic Linux usage, so you can use our HPC



## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

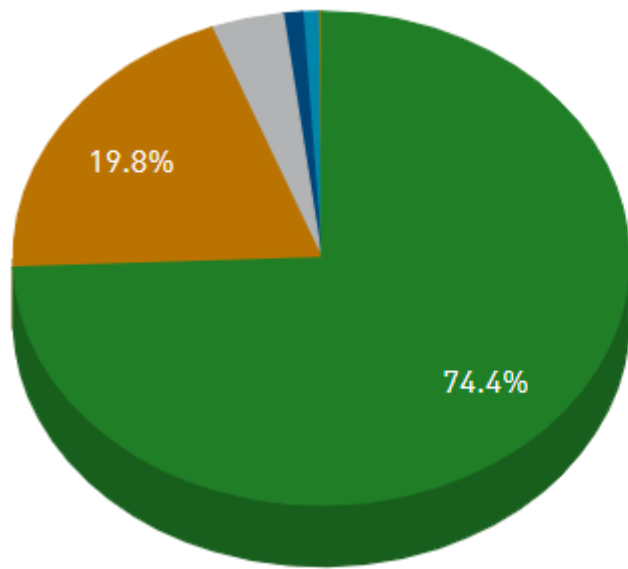
## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

- **Question: What operation system(s) do you use on a daily basis?**

# 1) What's Linux?

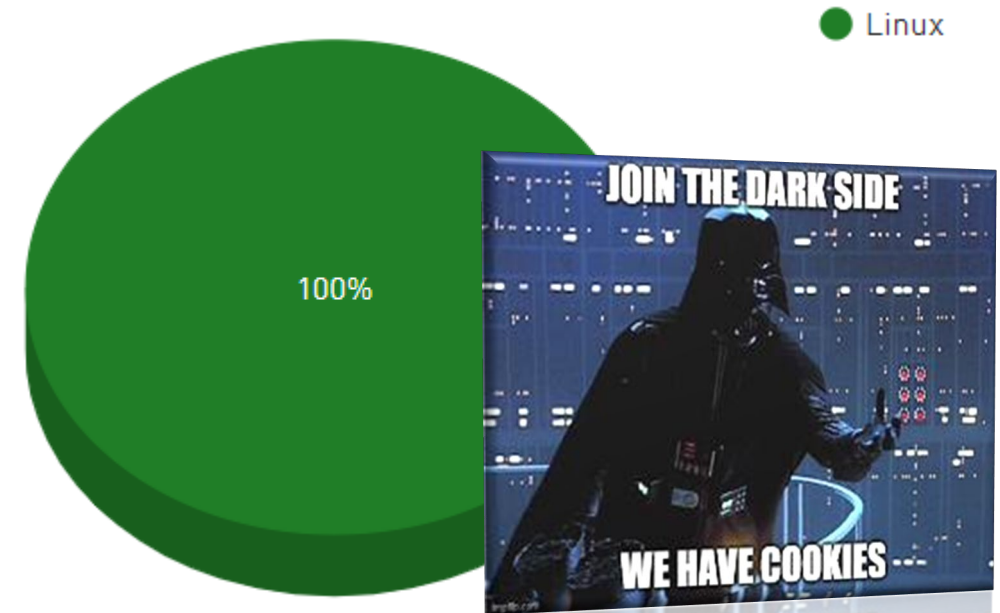
- OS on HPC
  - (Market share among Top 500 HPC)



Nov 2005

- Linux
- Unix
- Mixed
- Mac OS
- BSD Based
- Windows

*Windows' swan song!*



Nov 2017 - Today

[1] <https://top500.org/statistics/list/>



# 1) What's Linux?

- Linux – A journey to complete freedom!

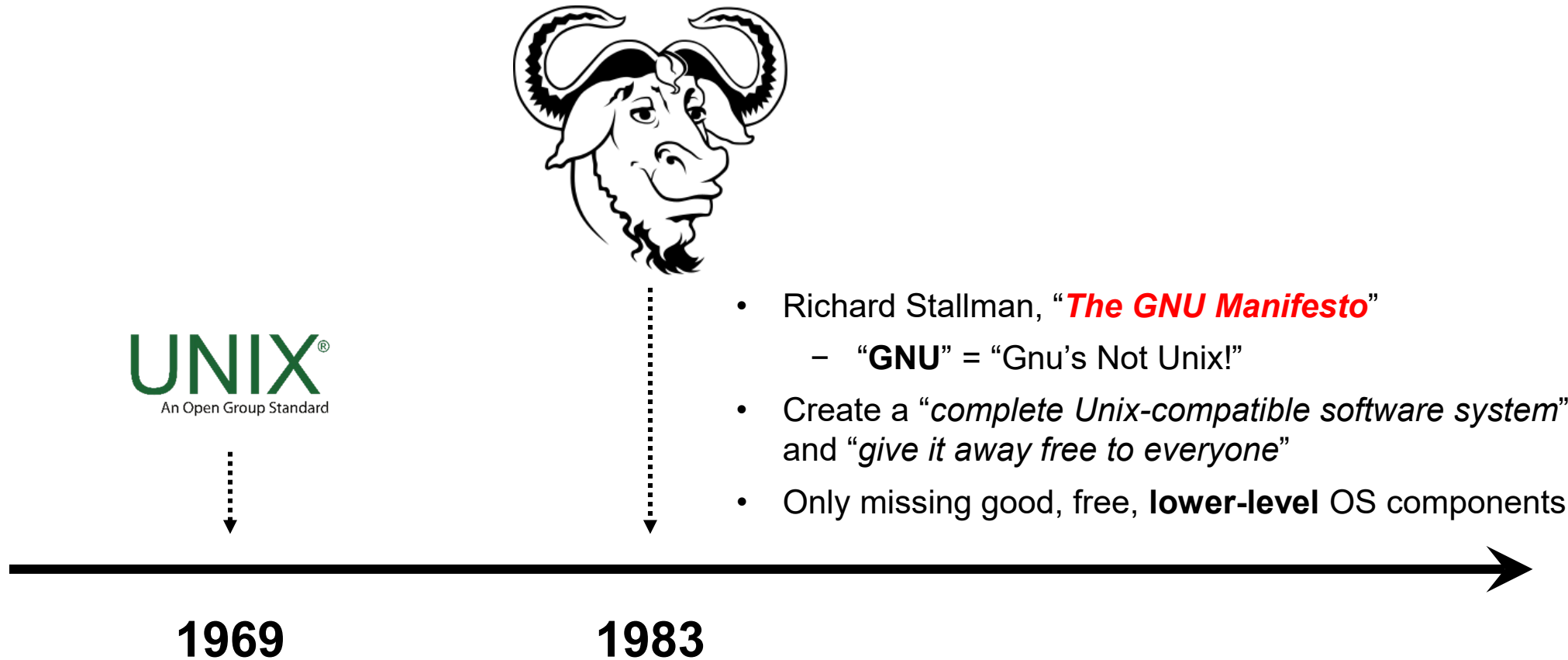
**UNIX**<sup>®</sup>  
An Open Group Standard

- Bell Labs
- A multitasking, multiuser operation system
- Traditionally proprietary & closed-source

1969

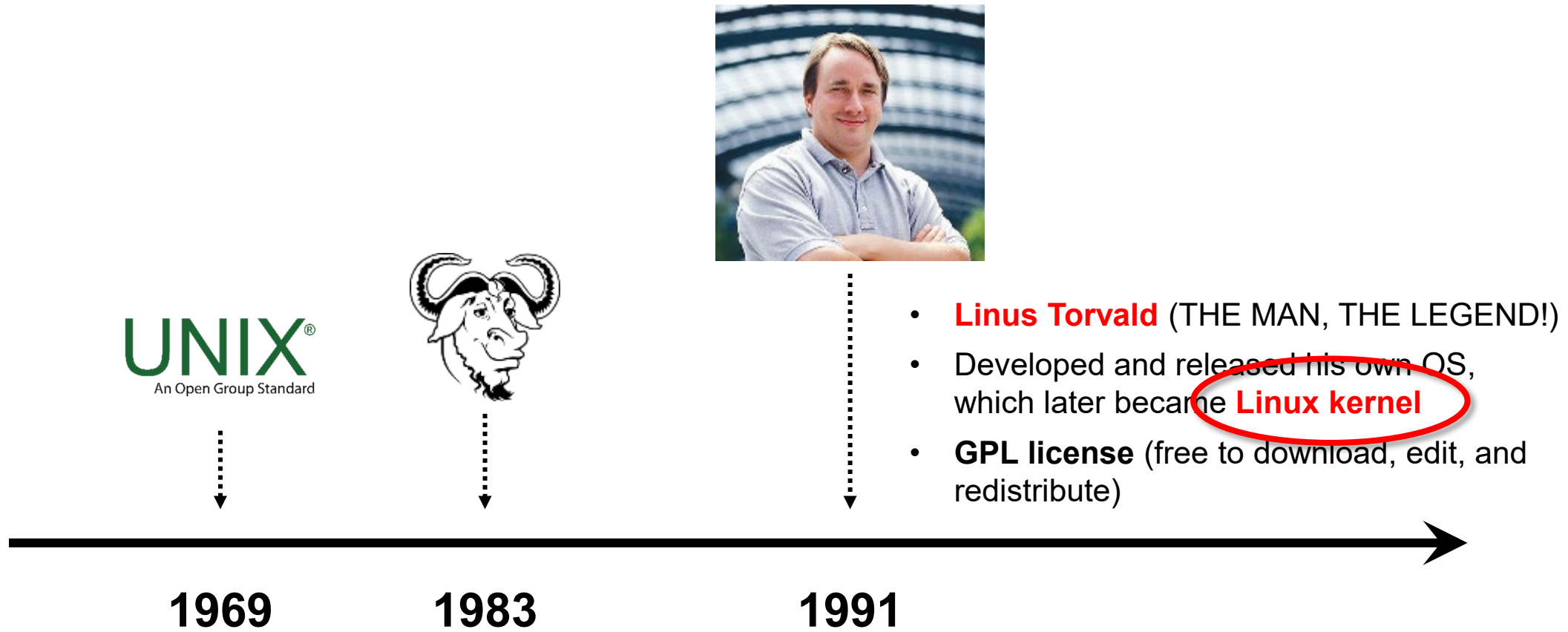
# 1) What's Linux?

- Linux – A journey to complete freedom!



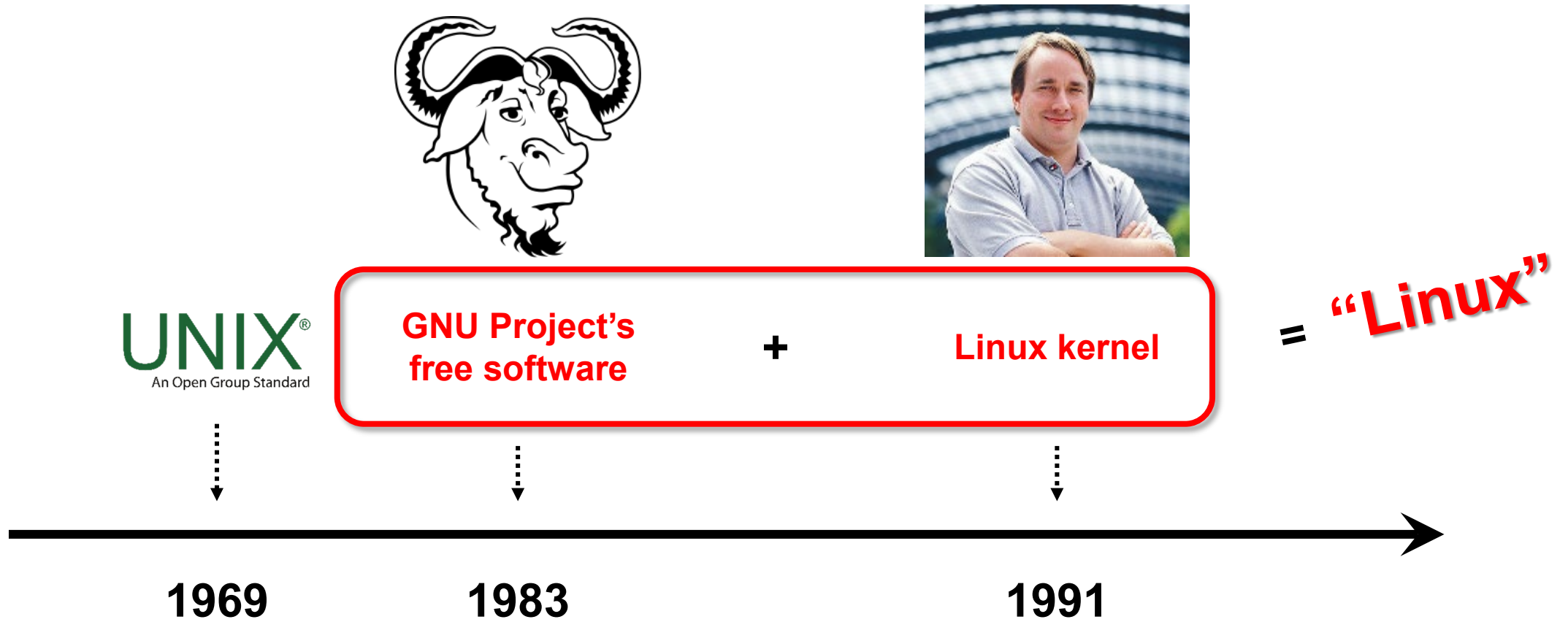
# 1) What's Linux?

- Linux – A journey to complete freedom!



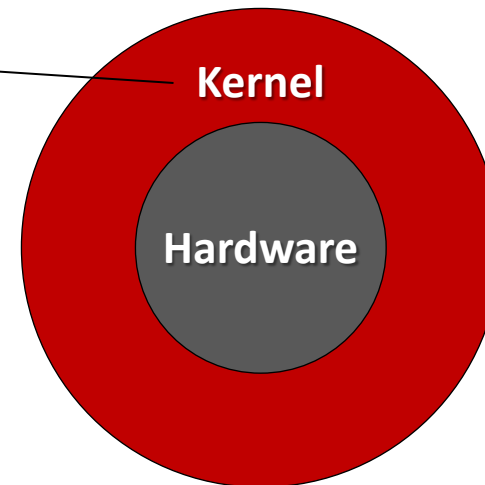
# 1) What's Linux?

- Linux – A journey to complete freedom!



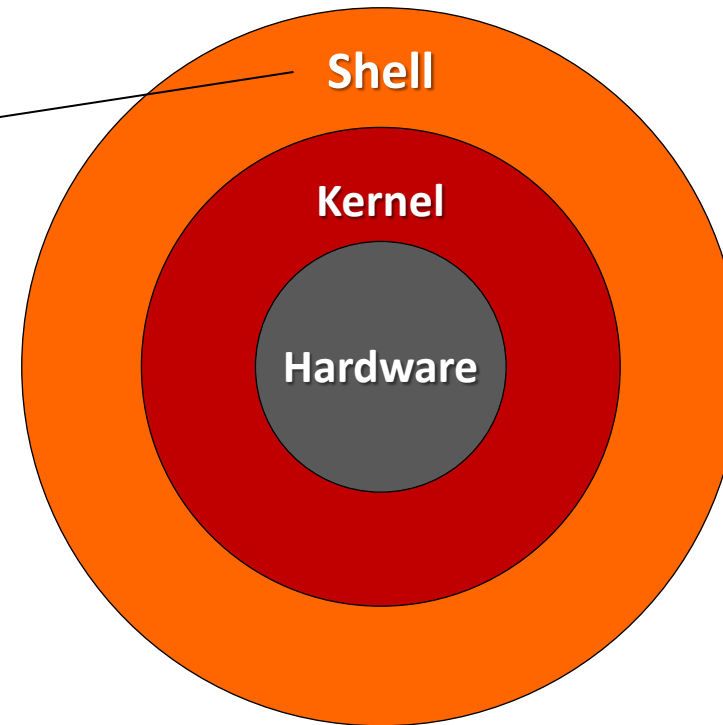
- A closer look at Linux architecture

- Contains (**low-level**):
  - Resource management
  - Process management
  - Device Drivers
  - System Calls
  - ...
- **Linux Kernel**, the part Linus Torvald developed and released for free



- A closer look at Linux architecture

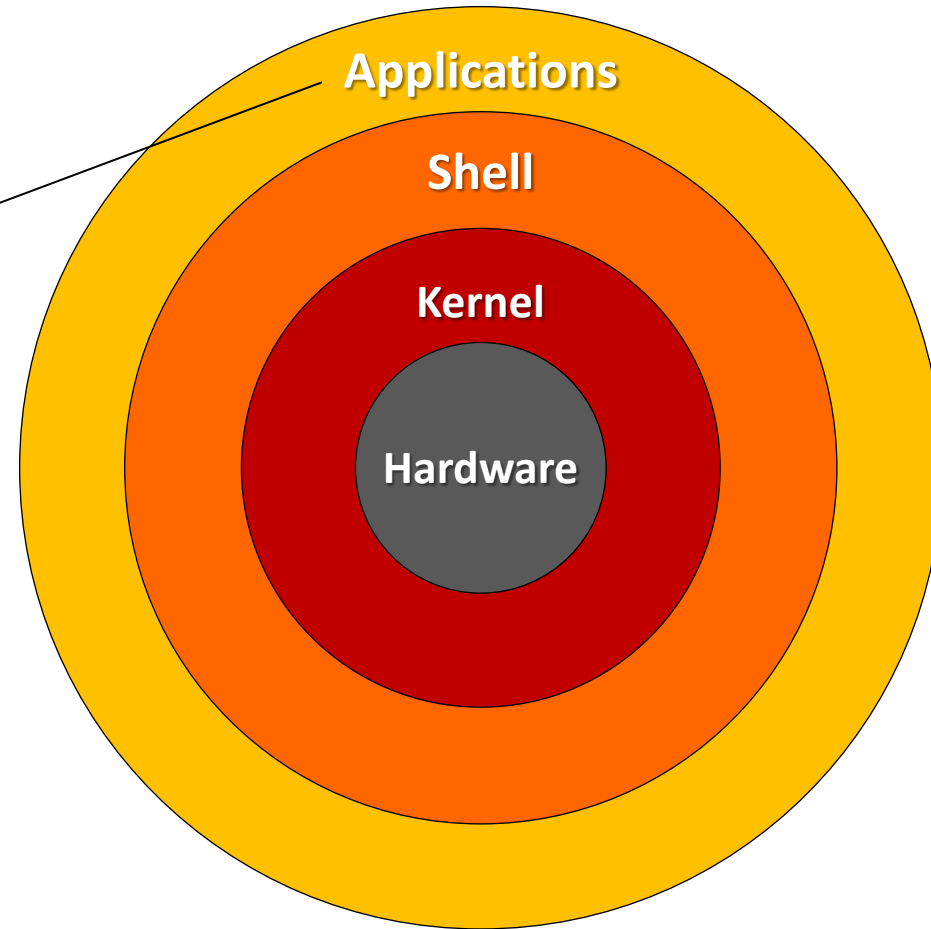
- Contains (**higher-level**):
  - Command execution
  - Scripting
  - ...
- **GNU Project** comes in



\*Join us on 7/22/2026 for **Basic Shell Scripting** !

- A closer look at Linux architecture

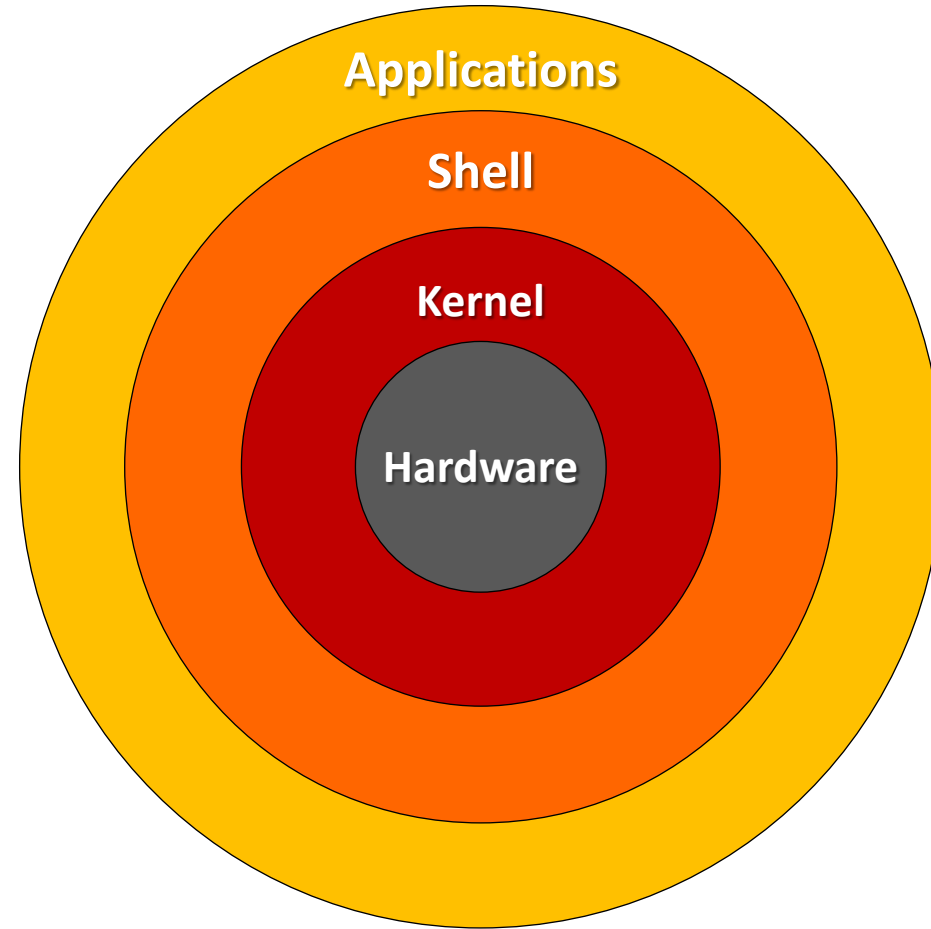
- Contains (**highest-level**):
  - File browser
  - Text editor
  - Multimedia
  - Internet browser
  - Office suites
  - ...
- **GNU Project** & beyond



# 1) What's Linux?

- A closer look at Linux architecture

“Linux”



# 1) What's Linux?

- Key to a legendary project

*Visionary boss (love & hate)*

*Stylish European designer*

*Quiet Geek Engineer*

**Apple**



Steve Jobs



Jony Ive



Steve Wozniak

**Tesla**



Elon Musk



Franz von Holzhausen



JB Straubel

**Linux**



Linus Torvald



Linus Torvald



Linus Torvald

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

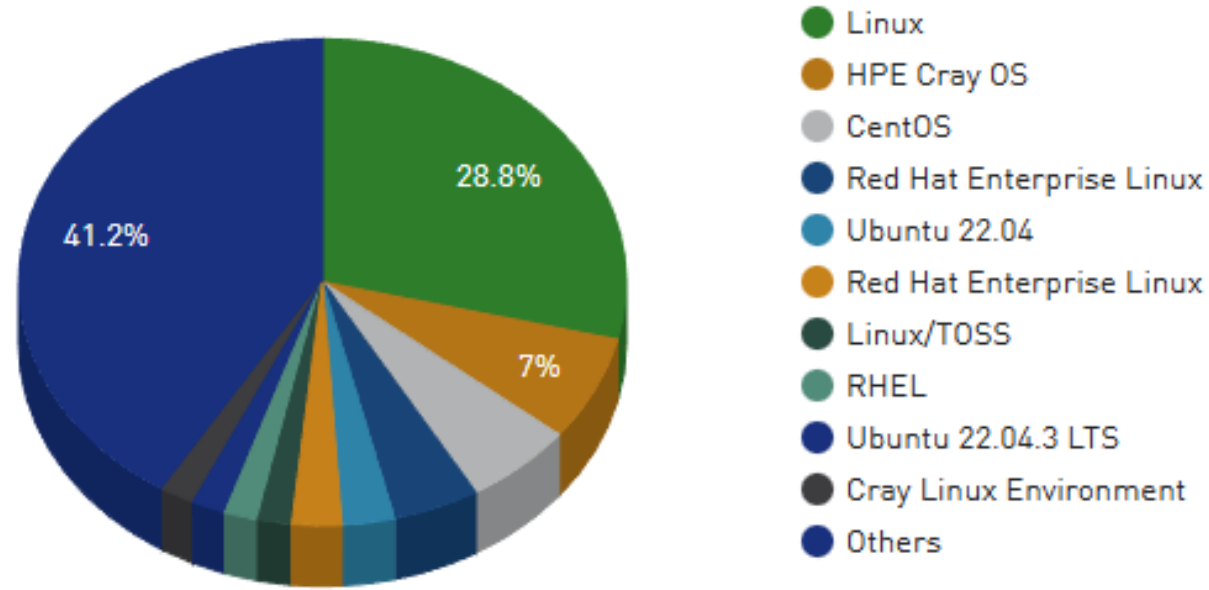
## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

# 2) Many Faces of Linux



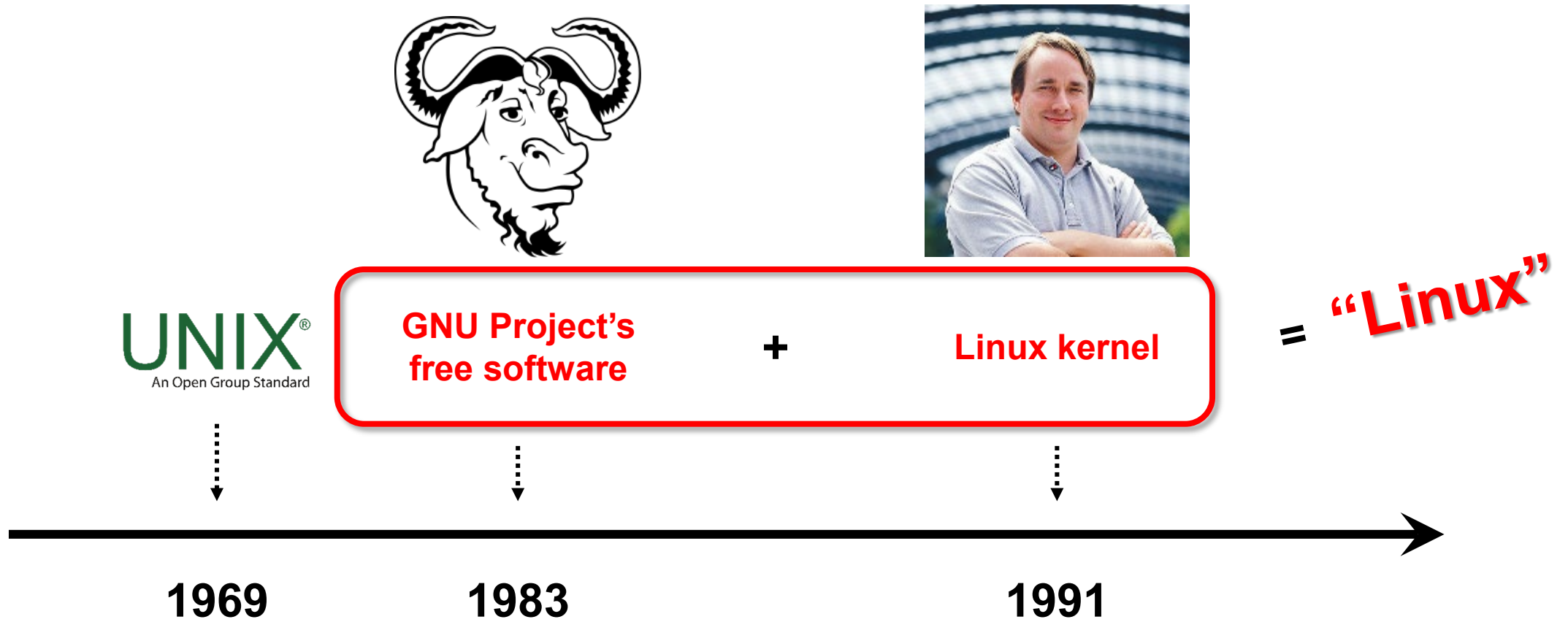
Nov 2025

[1] <https://top500.org/statistics/list/>



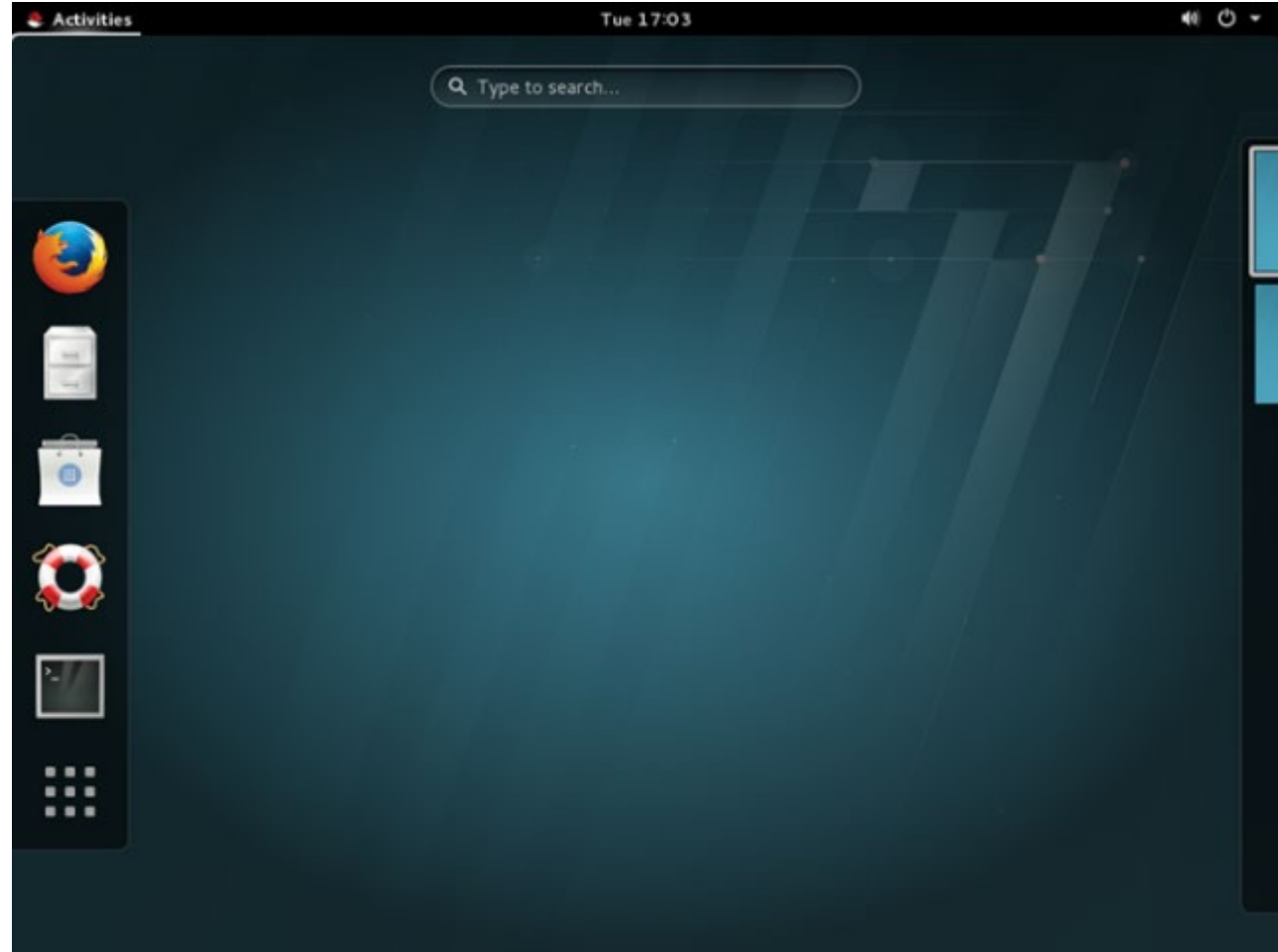
# 2) Many Faces of Linux

## a) Distributions (Distro)



## a) Distributions (Distro)

**Red Hat**  
(one of the industrial favorites)



# 2) Many Faces of Linux

## a) Distributions (Distro)

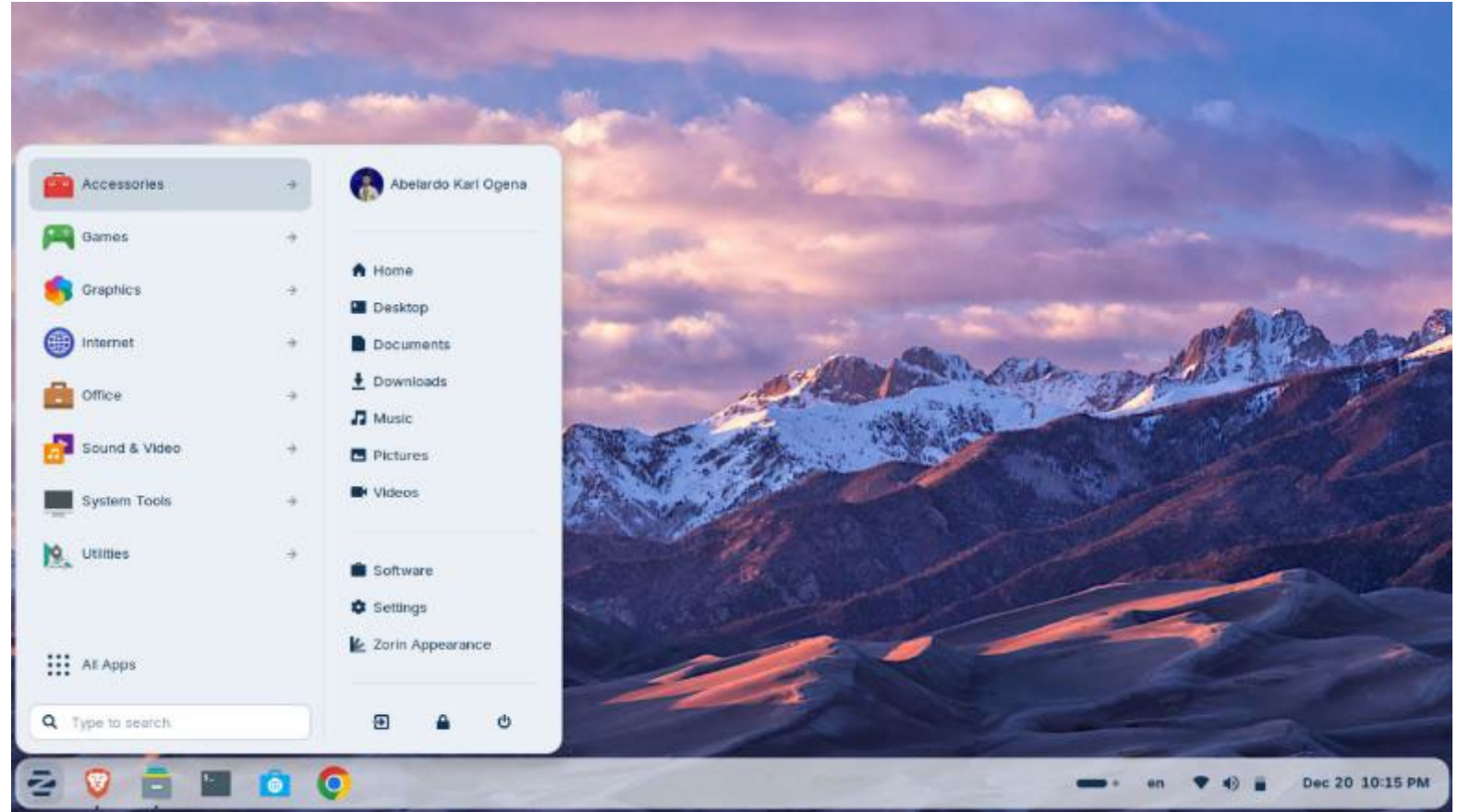
**Ubuntu**  
(one of the personal favorites)



# 2) Many Faces of Linux

## a) Distributions (Distro)

**Zorin OS**  
(one of the most  
Windows-like)



# 2) Many Faces of Linux

## a) Distributions (Distro)

**Nobara**

(Like gaming on Linux?)



### a) Distributions (Distro)

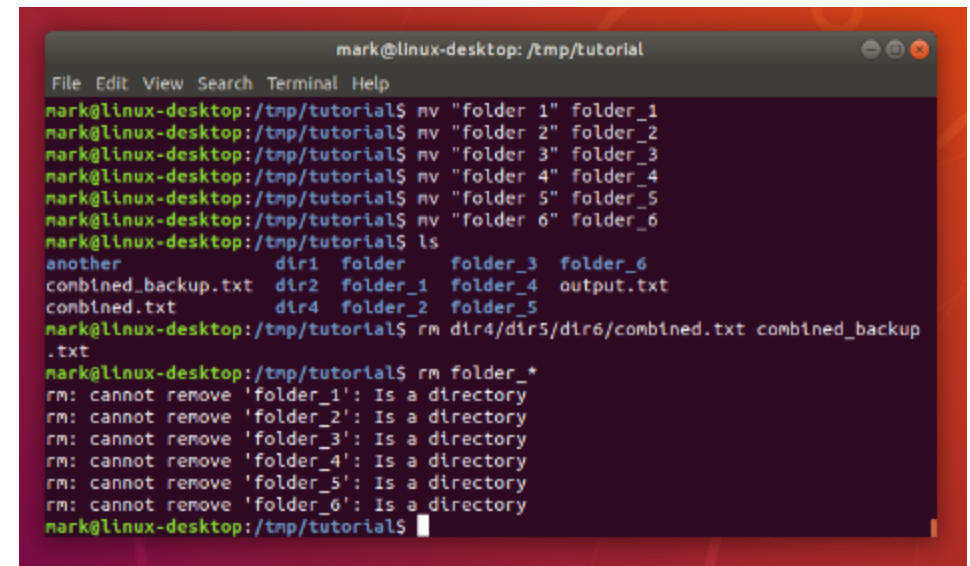
Question: Do you know what distro LSU / LONI HPC use?

**Red Hat Enterprise Linux (RHEL)**

## b) Desktop vs Terminal



Desktop (GUI)



Terminal (Command-line)

### b) Desktop vs Terminal

**We will only focus on terminal!**

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

# 3) Get Linux

- I am ready for Linux. How do I get my own Linux?

# 3) Get Linux

a) You do NOT have to!!

**Windows:**



User

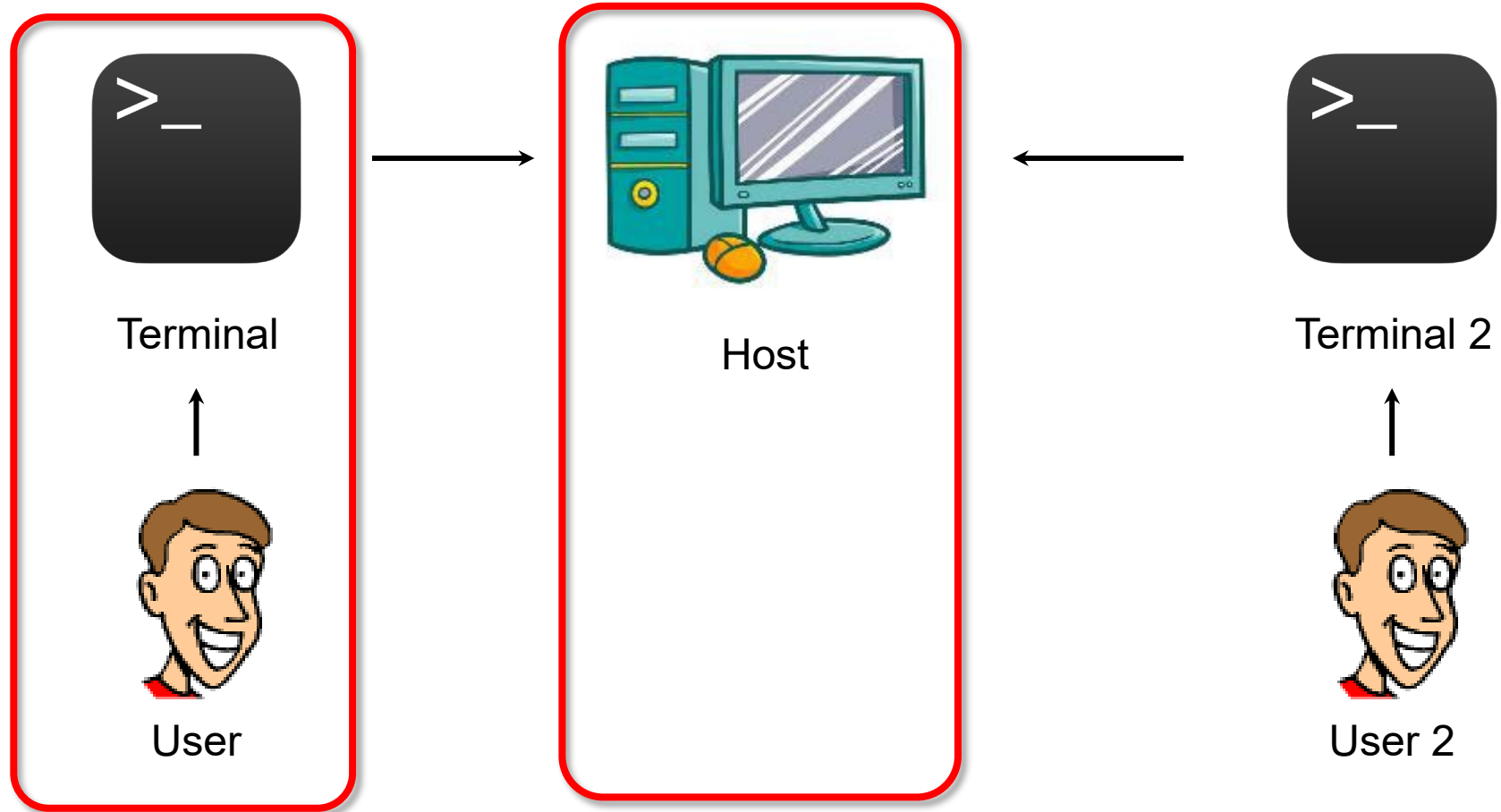


Machine

# 3) Get Linux

a) You do NOT have to!!

Linux:



a) You do NOT have to!!

**A terminal is all you need!**

Your OS ...	Your choices...
Linux / Mac / ChromeOS	Terminal
Windows	<b>MobaXterm</b> , Putty, ...
iOS / iPadOS	Termius, Blink Shell, ...
Android	Termux, ...

\* Join us next week for **HPC User Environment 1** !

## b) Install Linux on your own machine

- Pick a distro you like
- Read the instructions
- Either wipe your current OS and reinstall, or install alongside your current OS (dual OS)

## c) Virtual Machine

- Virtualize an entire OS on an existing OS
  - Pros: Full experience of Linux w/o wiping your current system
  - Cons: Could be slow
  
- Popular choices:
  - **VMWare**
  - **VirtualBox**
  - **Parallel Desktop** (Mac)
  - **Hyper-V** (Windows)

## c) Virtual Machine

- Special shout-out (Windows users only):

### Windows Subsystem for Linux (WSL)



- High-performance, resource efficient (i.e., works like native installation)
- Try me in cmd:

```
wsl --install
```

- **Why choose Linux?**
  - You have no choice, friend! 😊
- **Brief history of Linux**
- **Basic architecture**
- **Linux Distributions (Distros)**
- **Desktop vs terminal**
- **How to get Linux**
  - You don't have to!

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

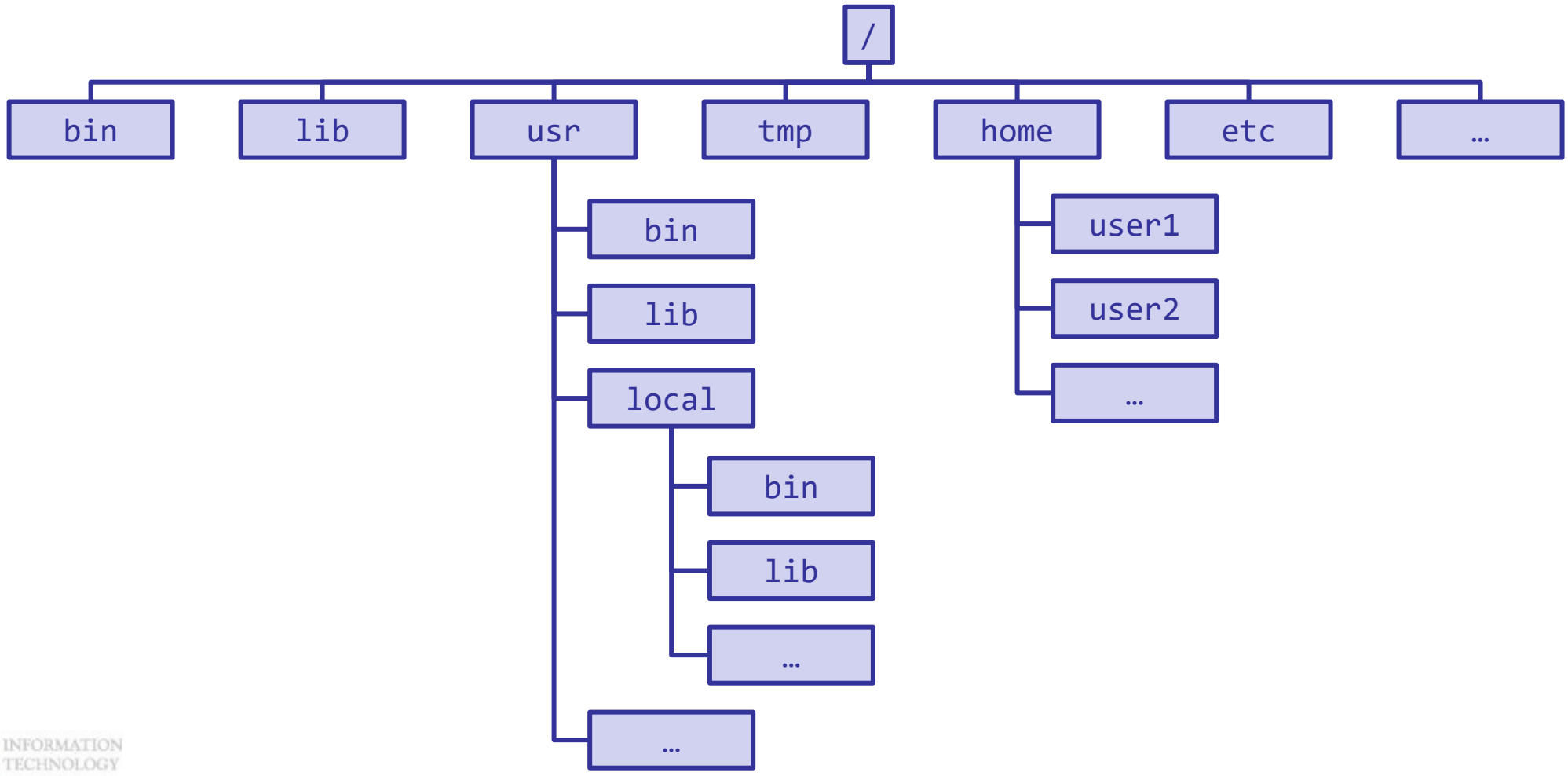
## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

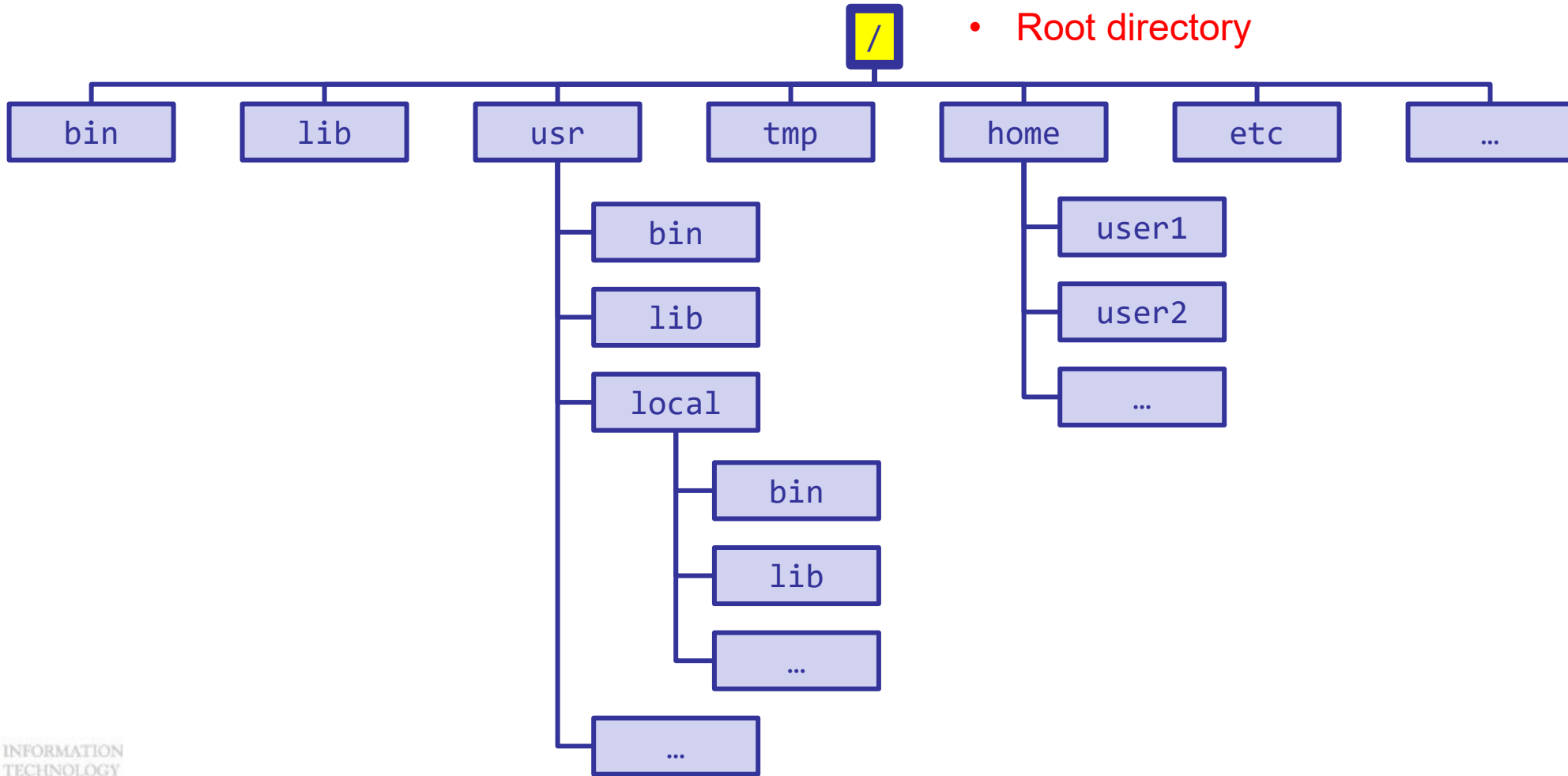
## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

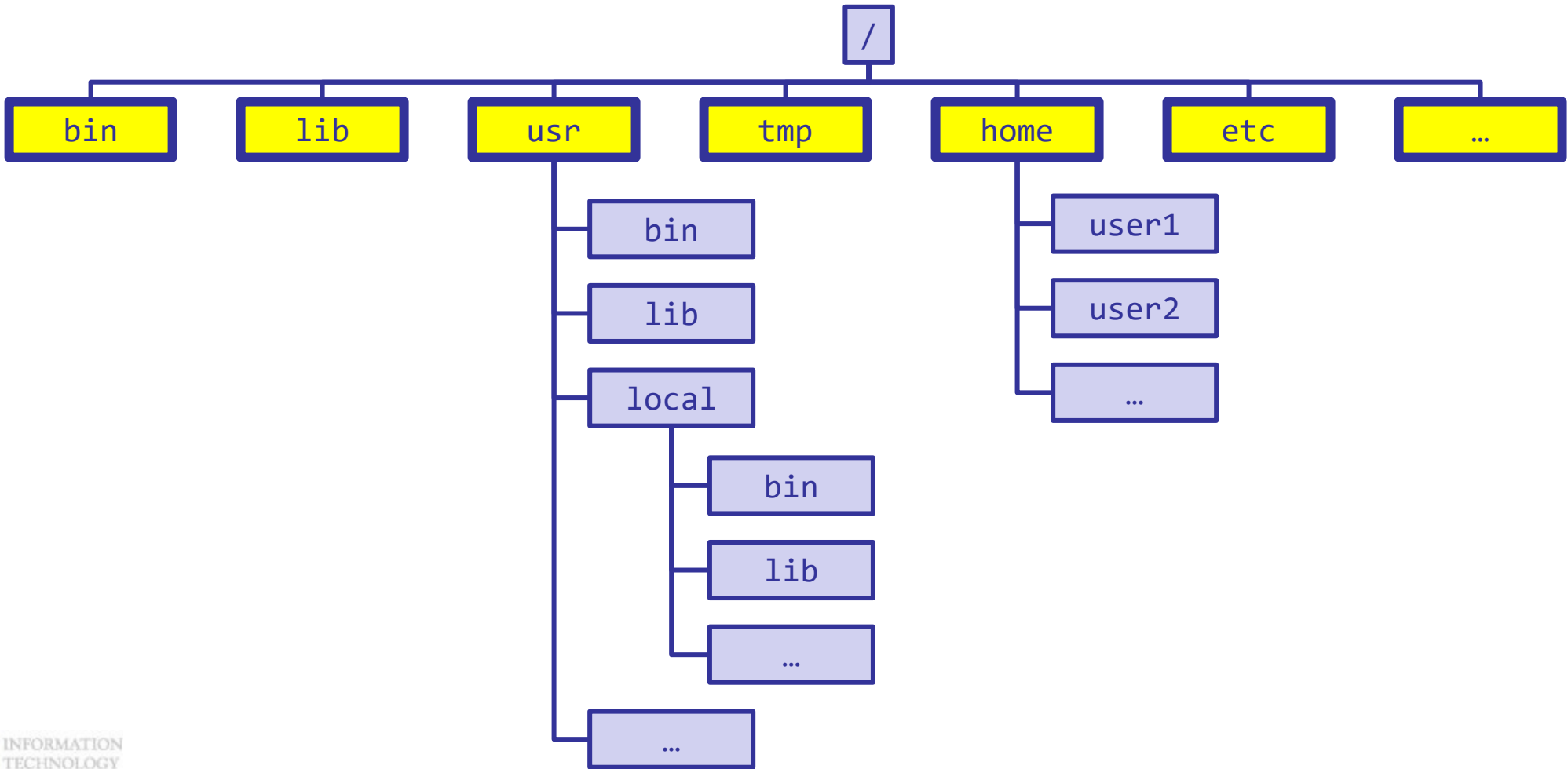
## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



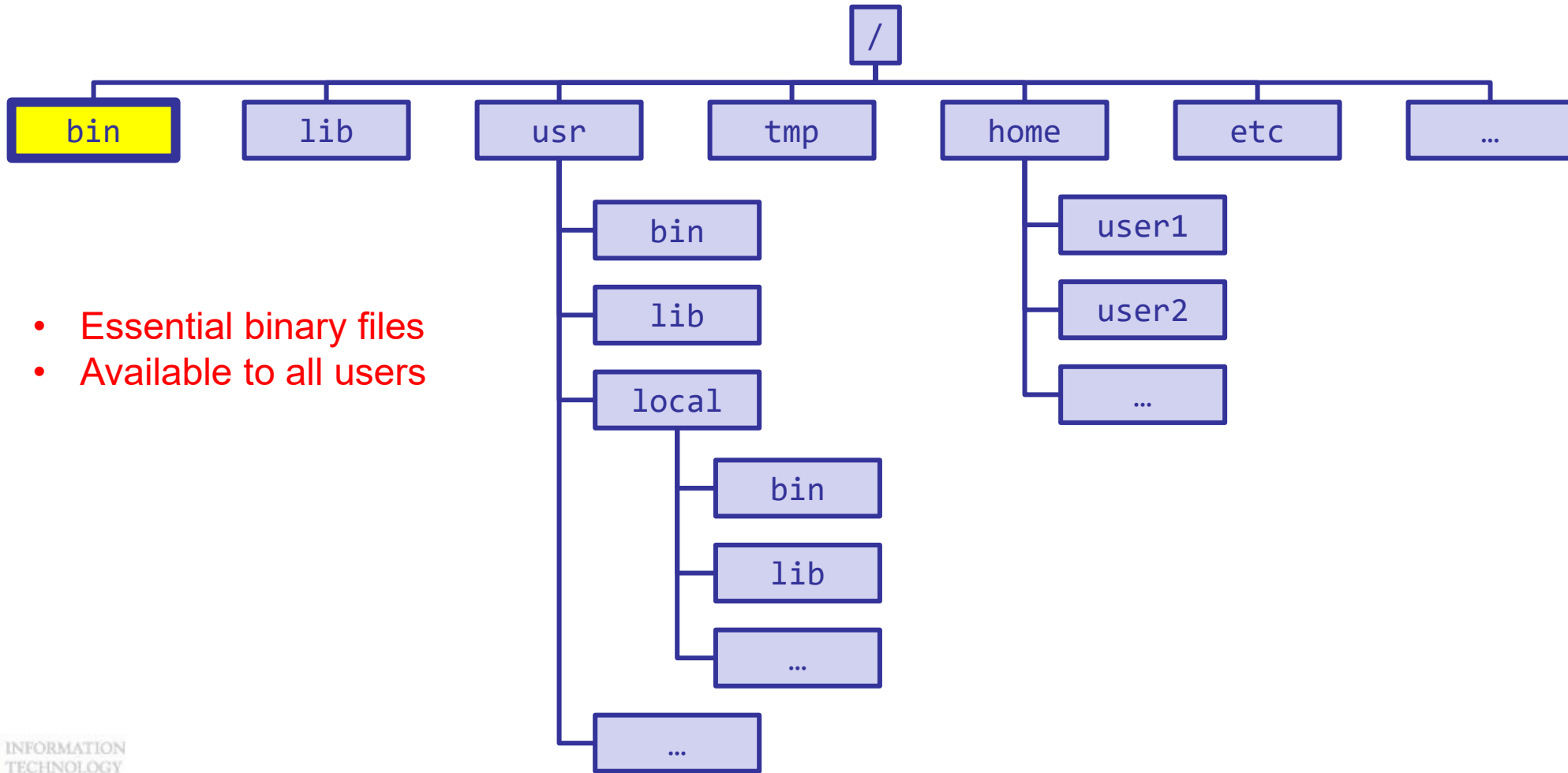
## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



## a) Filesystem Hierarchy Standard (FHS) : An inverted tree

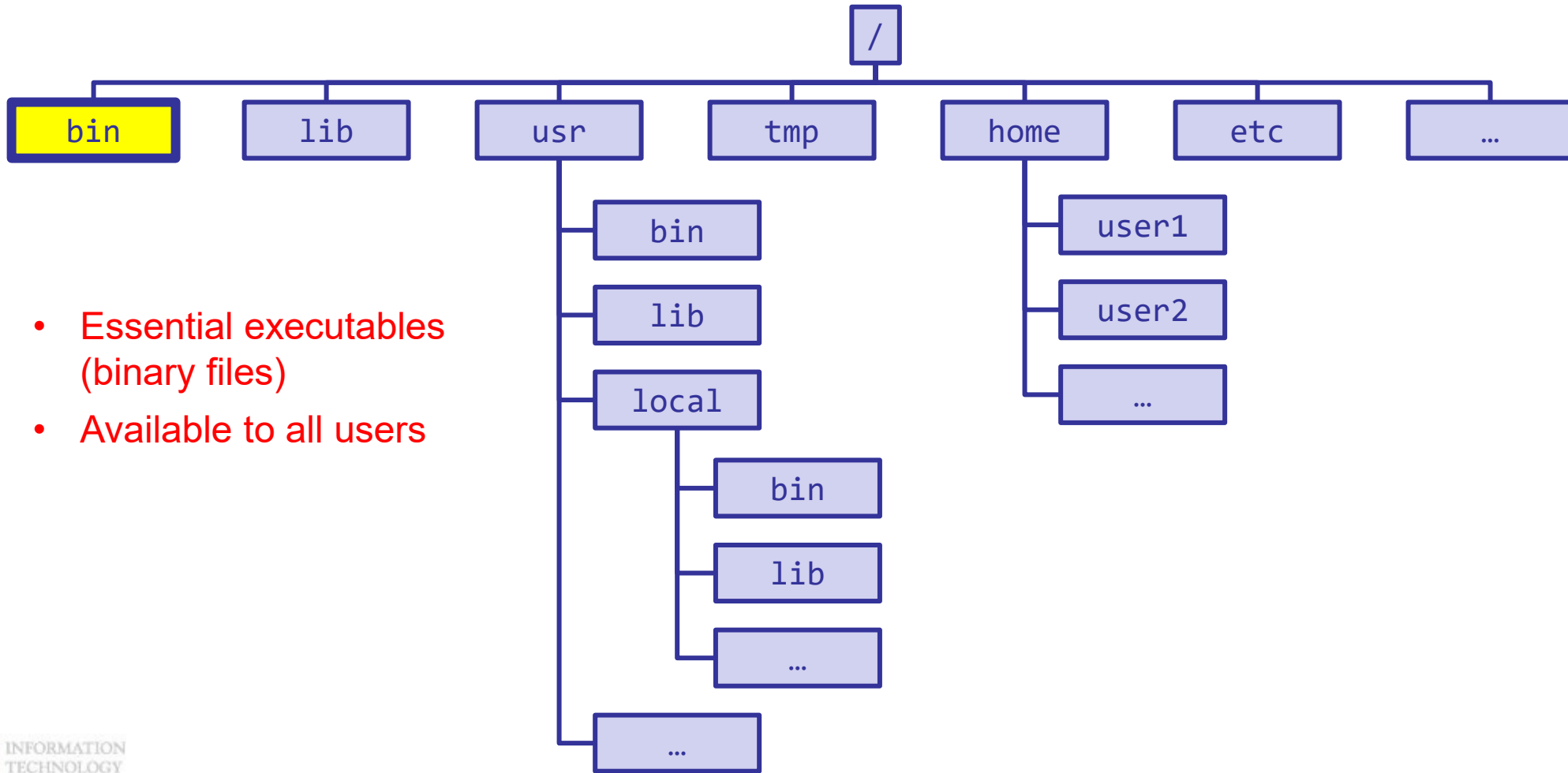


## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



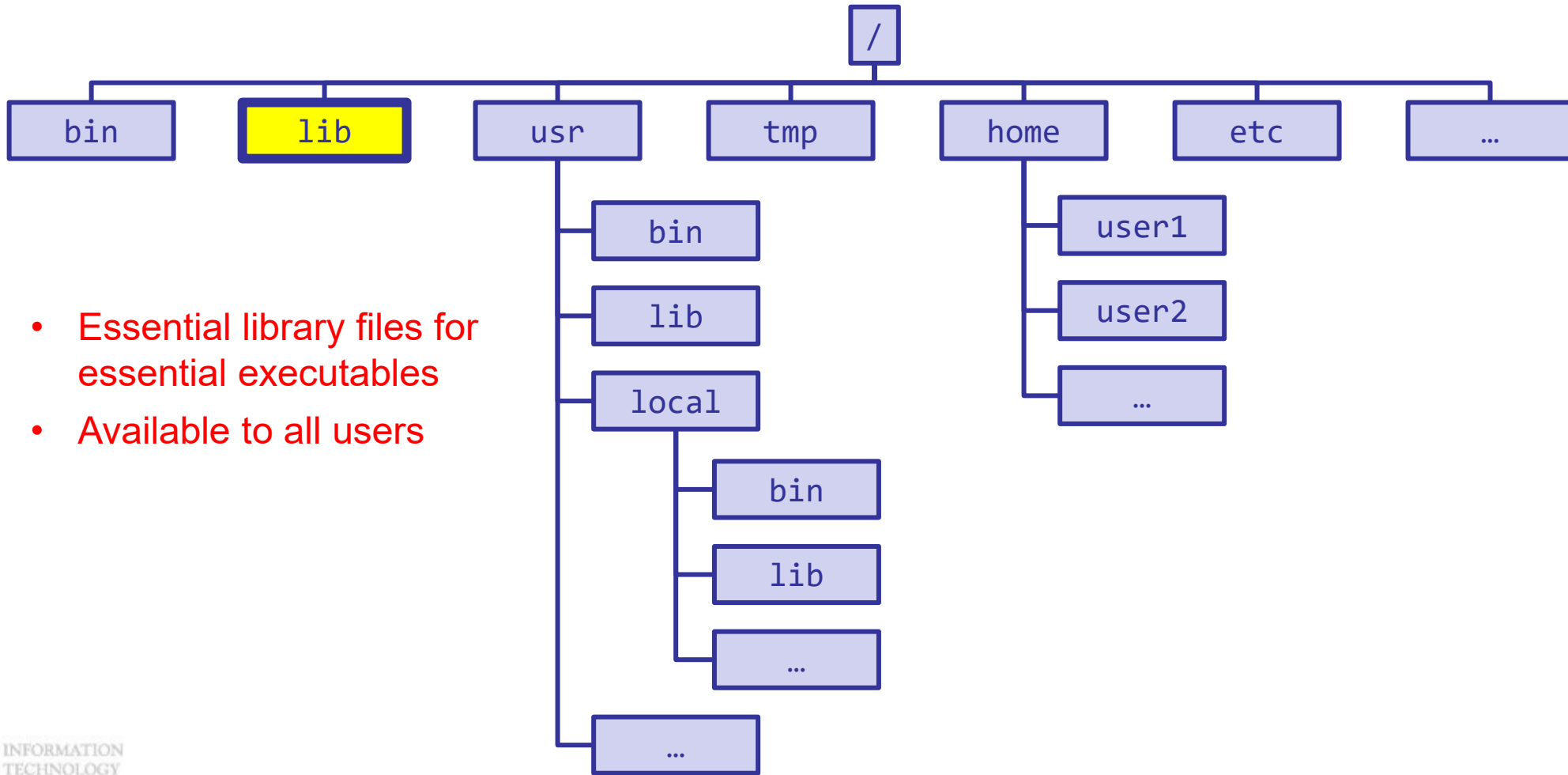
- Essential binary files
- Available to all users

## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



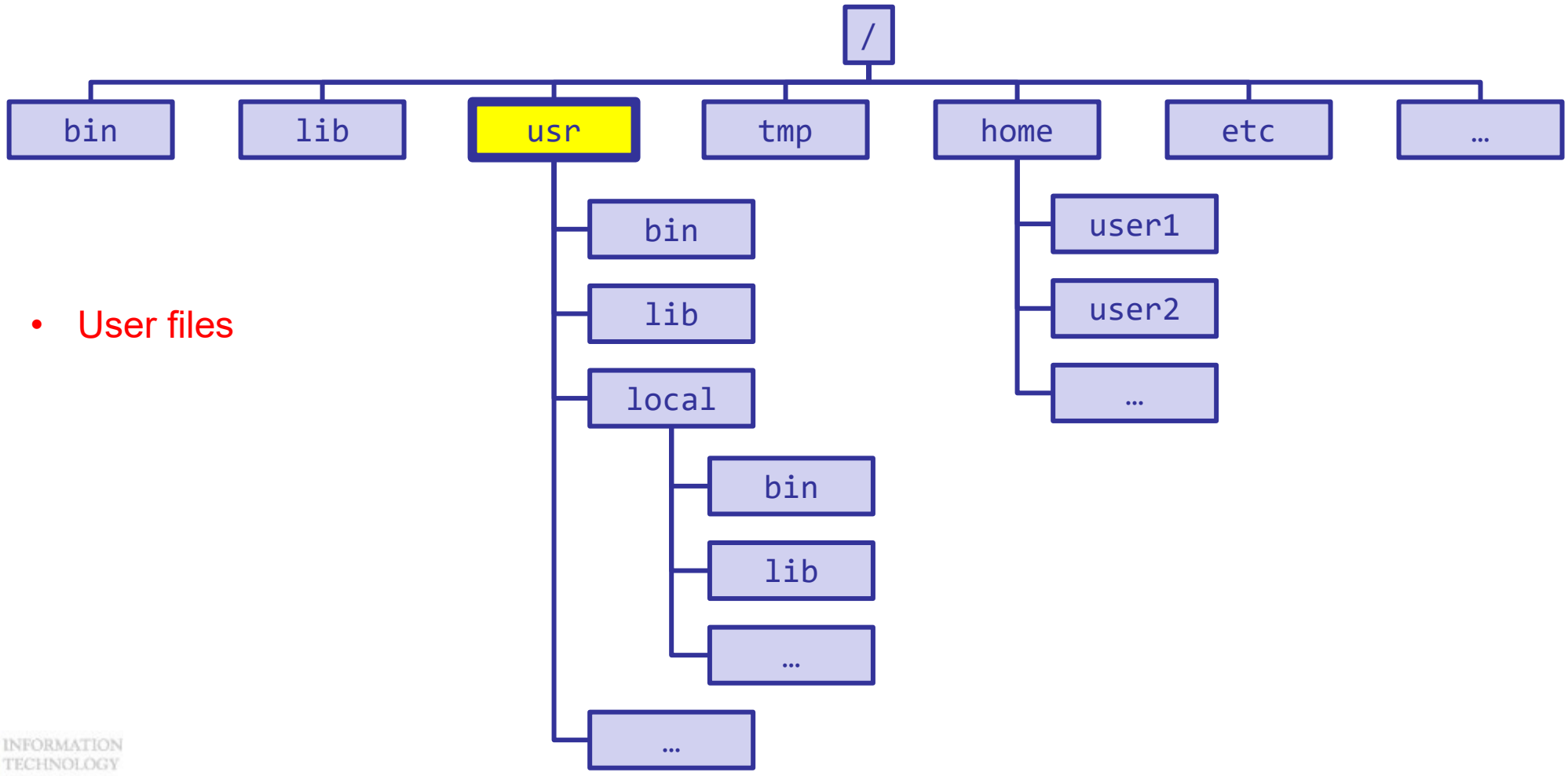
- Essential executables (binary files)
- Available to all users

## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



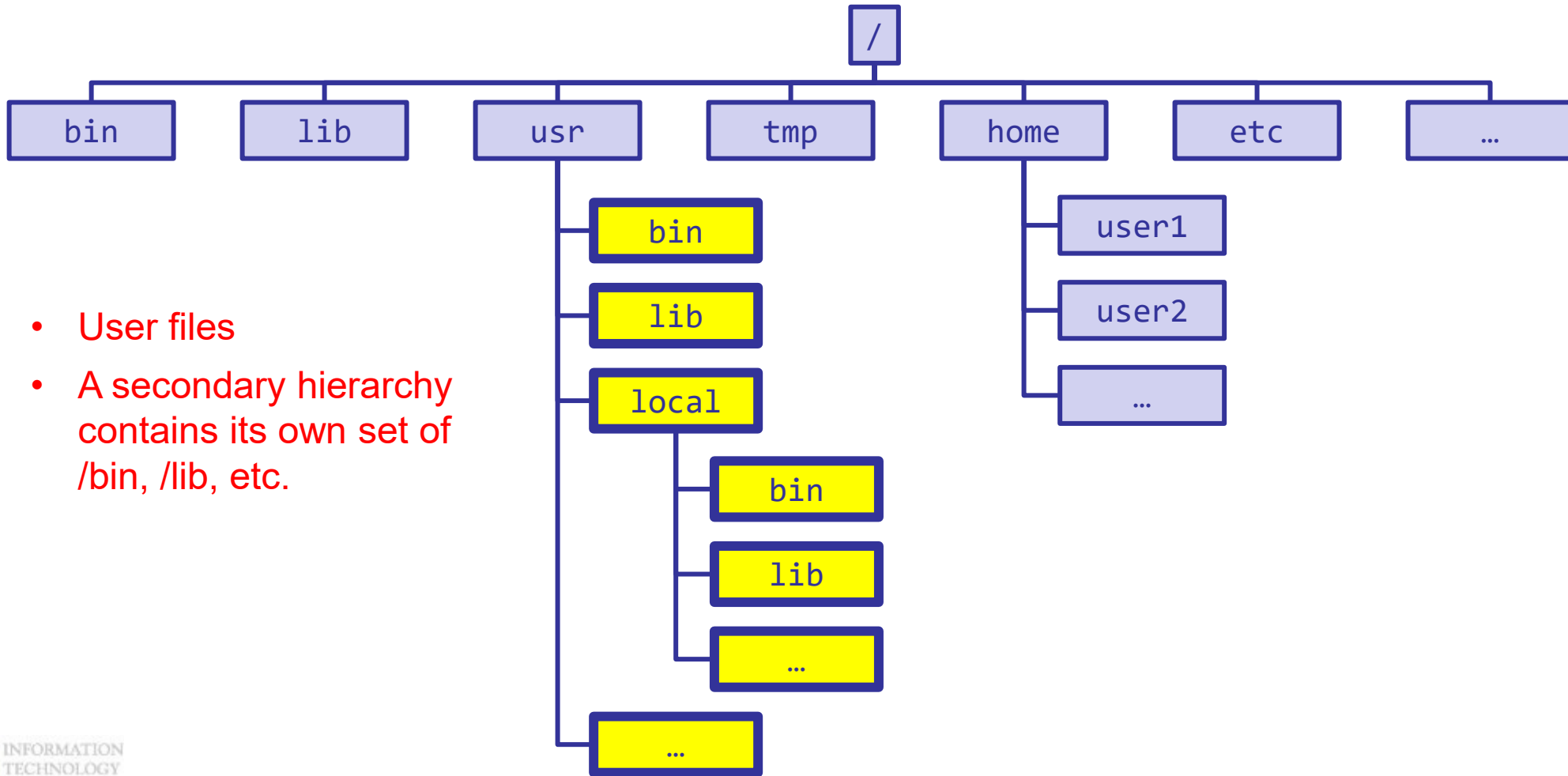
- Essential library files for essential executables
- Available to all users

## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



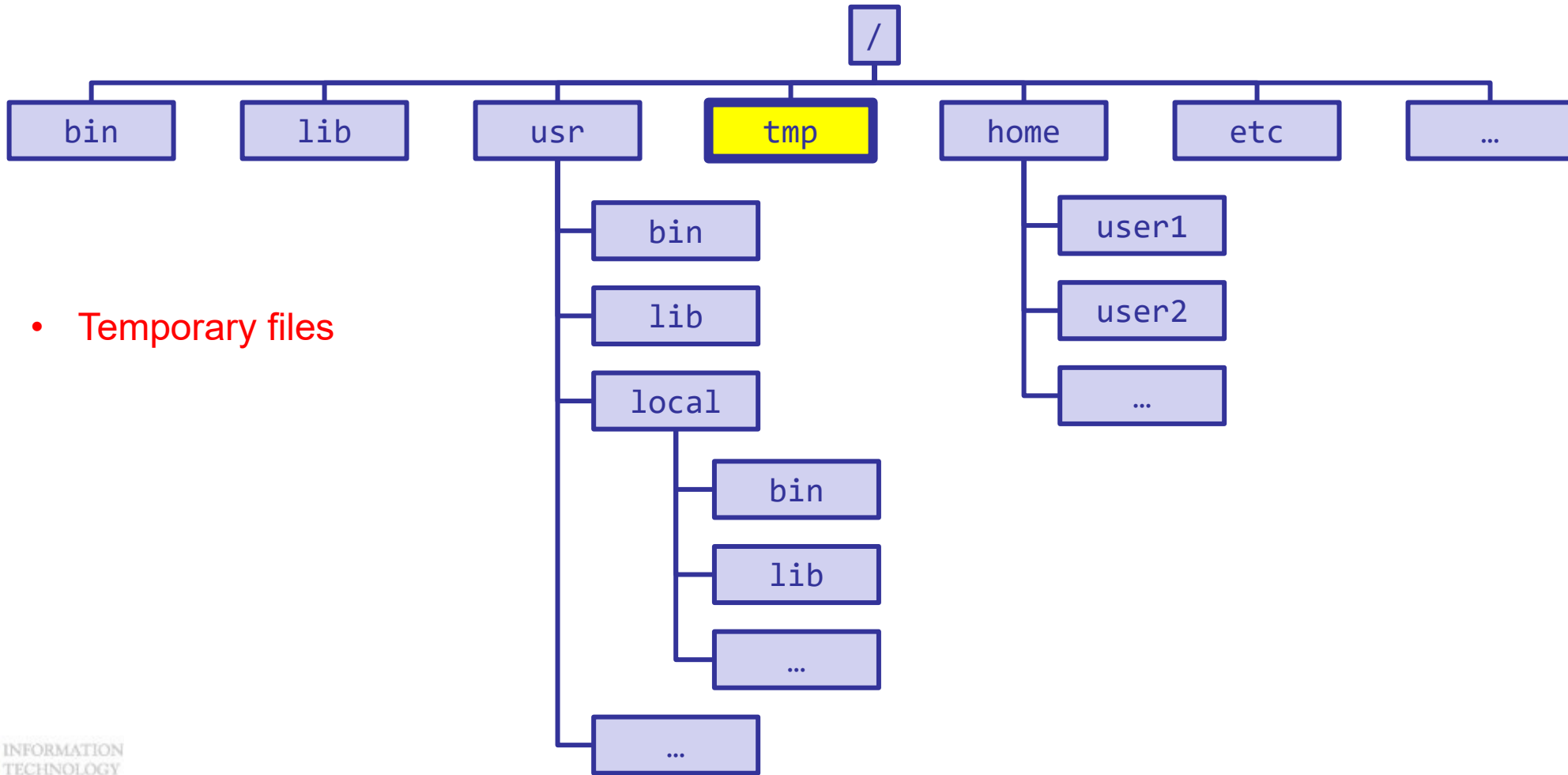
- User files

## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



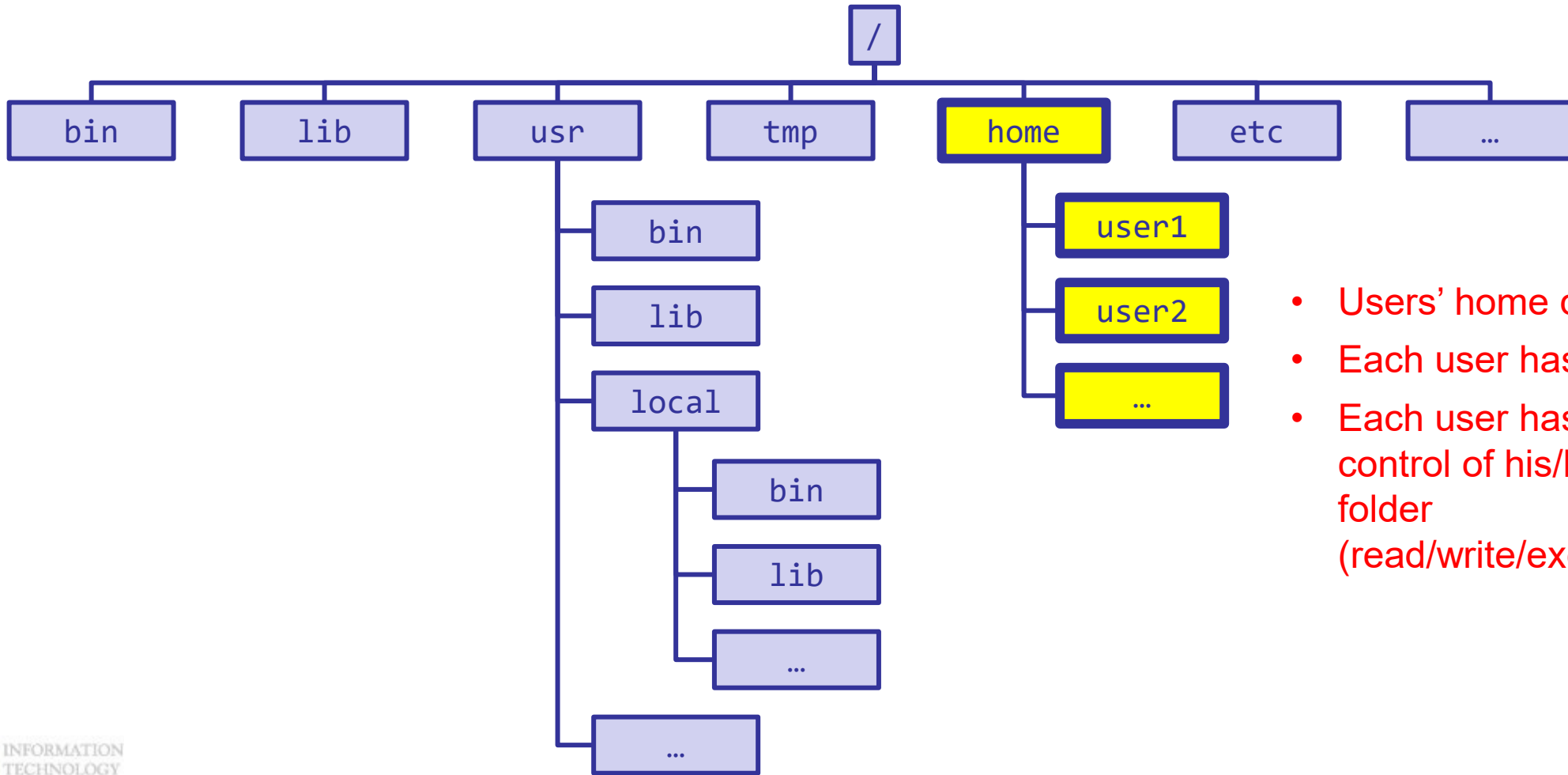
- User files
- A secondary hierarchy contains its own set of /bin, /lib, etc.

## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



- Temporary files

## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



- Users' home directory
- Each user has one
- Each user has full control of his/her own folder (read/write/execute)

## a) Filesystem Hierarchy Standard (FHS) : An inverted tree



- System configuration

## b) Two ways to use file path

- Path: Position of a file in the hierarchy
- **Case sensitive!** (“TeSt” ≠ “tEsT”)

### Absolute Path

- Start with “/”
- E.g., `/home/jasonli3/IntroToLinux/test.sh`
- Unique, does not depend on where you are

### Relative Path

- Does **not** start with “/”
- E.g., `IntroToLinux/test.py`
- Not unique, depends on where you are

## c) Special symbols

\* Assuming currently in: `/home/jasonli3/IntroToLinux/myfolder`

Path	Meaning	Example
<code>.</code>	Current path	<code>/home/jasonli3/IntroToLinux/myfolder</code>
<code>..</code>	Parent path	<code>/home/jasonli3/IntroToLinux</code>
<code>~</code>	User's home directory	<code>/home/jasonli3/</code>

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

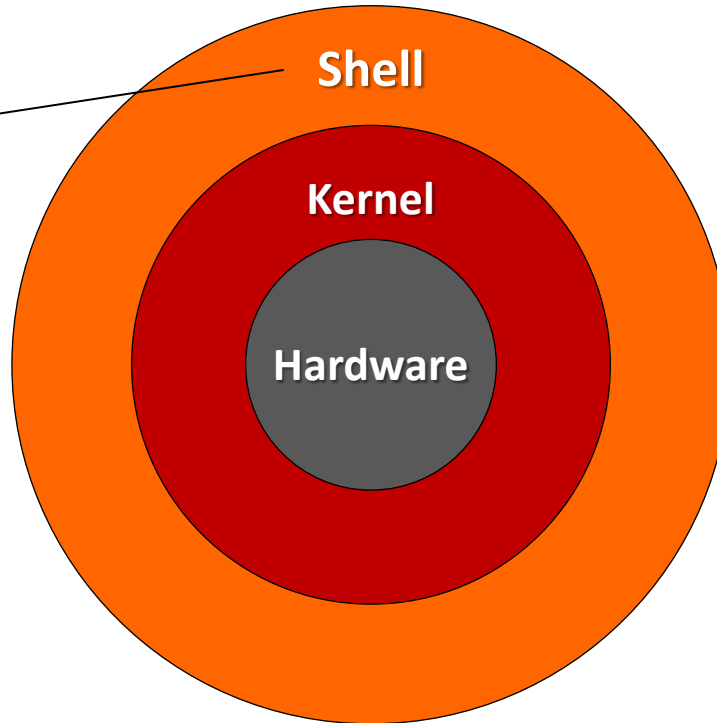
- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

- Before meeting your terminal

- Contains (**higher-level**):
  - Command execution
  - Scripting
  - ...
- **GNU Project** comes in



- **Before meeting your terminal**

- Different Shell, different terminal

- **sh** (Original Bourne Shell)
- **bash** (Bourne Again Shell)
- **cs**h (C Shell)
- **tcs**h (TENEX C Shell, more features)
- **ksh** (KornShell)

- **zsh** (Z Shell)
- **dash** (Debian Almquist Shell)
- **fish** (Friendly Interactive Shell)
- ...

- Supported on our HPC
- Feel free to use whichever you like!
- Can set your own default Shell

- **Before meeting your terminal**

- Different Shell, different terminal
  - **sh** (Original Bourne Shell)
  - **bash** (Bourne Again Shell)
  - **csh** (C Shell)
  - **tcsch** (TENEX C Shell, more features)
  - **ksh** (KornShell)
  - **zsh** (Z Shell)
  - **dash** (Debian Almquist Shell)
  - **fish** (Friendly Interactive Shell)
  - ...

- Default for lots of Linux distro
- Will only talk about it today
- Customizable, will only talk about default look

## 2) Meet You Terminal

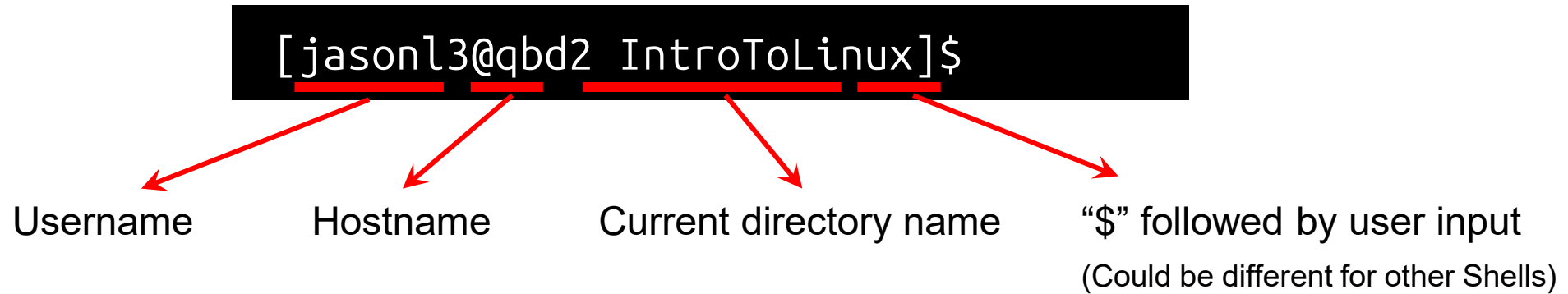
- What Terminal looks like

```
[jasonl3@qbd2 IntroToLinux]$
```

**“Prompt”**



- What Terminal looks like



\* Join us next week for **HPC User Environment 1** !

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

- **What to expect in this part:**
  - The most basic commands you would use daily to navigate through Linux

- Common Linux command format

command [options] arguments

- Regulates command behavior
- Optional
- Usually like “-X” or “--xxxx”

- May or may not required
- E.g., a string, a file path, ...

- Common “universal” options

Option	Description
-h --help	Help information
-v --version	Software version

# 3) Basic Commands

- a) **Navigation**
- b) **File operations**
- c) **Content display**
- d) **Other essentials**

## a) Navigation

Command	Option	Augument	Description
ls ("list")		[PATH]	List given files, or content of a given directory (if [PATH] is a directory)
			List current directory
	-a	[PATH]	Show hidden files (start with ".")
	-l	[PATH]	Show details
	-h	[PATH]	Show human-readable file size
	-r	[PATH]	Reverse order
	-t	[PATH]	Sort by modification time (newest first)

## a) Navigation

Command	Option	Augument	Description
<b>cd</b> ("change directory")		[DIR]	Change current working directory to given destination (absolute or relative)
			Change to user's home directory
<b>pwd</b> ("print working directory")			Print current directory in standard output.

## b) File operations

Command	Option	Augument	Description
<b>cp</b> ("copy")		SOURCE DEST	Copy source file(s) to destination. <ul style="list-style-type: none"><li>• If DEST is an existing directory: Copy SOURCE under it</li><li>• If SOURCE is single file: Can use DEST to rename file after copying</li><li>• If SOURCE has multiple files: DEST must be an existing directory to move under</li></ul>
	-r	SOURCE DEST	Copy directory recursively
	-i	SOURCE DEST	Prompt before overwritten

## b) File operations

Command	Option	Augument	Description
<b>mv</b> ("move")		SOURCE DEST	Move source file(s) to destination. <ul style="list-style-type: none"><li>• If DEST is an existing directory: Move SOURCE under it</li><li>• If SOURCE is single file: Can use DEST to rename file after copying (<b>This is how you rename file in Linux!</b>)</li><li>• If SOURCE has multiple files: DEST must be an existing directory to move under</li></ul>
	-i	SOURCE DEST	Prompt before overwritten

## b) File operations

Command	Option	Augument	Description
<code>rm</code> ("remove")		FILE	Remove file(s)
	<code>-r</code>	FILE	Copy directory recursively
	<code>-i</code>	FILE	Prompt before removal
	<code>-f</code>	FILE	Force removal (ignore prompt). <b>DANGER!</b>

- **CAUTION!** Files removed **CANNOT** be recovered!
- Be extremely careful whenever you do `rm -rf` !!!!

## b) File operations

Command	Option	Augument	Description
<b>mkdir</b> ("make directory")		DIR	Create a directory under an <b>existing</b> parent directory
	-p	DIR	No error. Attempt to create parent directory(s) if needed

## c) Content display

Command	Option	Augument	Description
<b>cat</b> ("Concatenate")		[FILE]	Print file content to standard output
<b>head</b> / <b>tail</b>		[FILE]	Print first / last <b>10 lines</b> of a file
	-n[N]	[FILE]	Print first / last [N] <b>lines</b> of a file
	-c[N]	[FILE]	Print first / last [N] <b>bytes</b> of a file

- `head / tail` only loads given portion, extremely fast for large files

## c) Content display

Command	Option	Augument	Description
<code>more/less</code>		FILE	Display a file

### `more`

- Loads a chunk at a time
- Forward scrolling only
- Limited search function

→ Fast

### `less`

- Loads entire file
- Forward / backward scrolling
- Powerful search function

→ Versatile

## d) Other essentials

Command	Option	Augument	Description
<code>echo</code>		[STRING]	Print [STRING] to standard output
<code>date</code>			Print current time stamp
<code>alias</code>		CMD=STRING	Create CMD command as alias to STRING • E.g., <code>alias work="cd /work/\$USER"</code>
<code>man</code>		CMD	Display manual of a command (if supported)

- Cheat sheet

Command		Description
Navigation	<code>cd</code>	Change directory.
	<code>pwd</code>	Print current directory in standard output.
	<code>ls</code>	List files at a given location .
File Operations	<code>cp / mv</code>	Copy / Move files.
	<code>rm</code>	Remove files.
	<code>mkdir</code>	Create a directory.
Display	<code>cat</code>	Print out an entire file in standard output.
	<code>head / tail</code>	Show first / last several lines of a file.
	<code>more / less</code>	Display file one page at a time.
Others	<code>echo</code>	Print out strings in standard output.
	<code>date</code>	Print out current date & time in standard output.
	<code>alias</code>	Create alias to a command
	<code>man</code>	Display manual of a command (if supported)
		...

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

# 4) Terminal Like a Pro

- A few tips to help you use terminal more efficiently

## a) Auto completion

- Press **TAB**
- Automatically complete file paths & command names based on entered text
- **Example:** Current directory contains “Documents”, “Downloads”, “Desktop”
  - Enter “**ls D**”, then press TAB
  - Enter “**ls Do**”, then press TAB
  - Enter “**ls Dow**”, then press TAB

## b) Navigate through past commands

- Imaging I am debugging a very lengthy command...

```
python3 -m torch.distributed.launch --nproc_per_node=8 --node_rank=0 --master_addr=192.168.1.1 \
master_port=29500 \
train.py \
--model "vit_huge_patch14_clip_224" --dataset "imagenet_1k" --data_path \
"/datasets/imagenet" \
--batch_size 128 --accumulation_steps 4 --epochs 100 --warmup_epochs 5 \
--opt "adamw" --lr 1.5e-4 --weight_decay 0.05 --clip_grad 1.0 \
--sched "cosine" --min_lr 1e-6 --reprob 0.25 --mixup 0.8 --cutmix 1.0 \
--precision "bf16" --dist_backend "nccl" --use_checkpointing \
--output_dir "./experiments/run_alpha_v2" --log_interval 10 --save_freq 5 \
--wandb_project "vision_transformer_scaling" --wandb_run_name "node0_h100_cluster" \
--distributed --sync_bn --pin_mem --num_workers 16 \
```

## b) Navigate through past commands

### i. Arrow keys (simplest)

- Use ↑ / ↓ keys to navigate through past commands

## b) Navigate through past commands

### ii. history command

- A comprehensive list of all previous commands

## b) Navigate through past commands

### iii. Reverse search (most advanced)

- Press `Ctrl + R`
- Type any phrases you want to search (keep pressing `Ctrl + R` until found)
- Press `TAB` to select, or `ENTER` to execute directly

## c) Wildcards

- Used to replace part(s) of the paths

Wildcard	Description	Example
*	Replace any number of any character	<pre>ls test*</pre> <ul style="list-style-type: none"><li>- List any file start with “test”</li></ul>
?	Replace any single character	<pre>ls test.??</pre> <ul style="list-style-type: none"><li>- List any file start with “test”, with 2-character extension</li></ul>
[ ]	Replace a single of a specific range	<pre>ls test.[0-9]</pre> <ul style="list-style-type: none"><li>- List any file start with “test”, with a single digit as extension</li></ul>

## d) Shortcuts

Shortcut	Description
<code>Ctrl + C</code>	Interrupt current process
<code>Ctrl + D</code>	Exit terminal or session
<code>Ctrl + L</code> (or <code>clear</code> command)	Clear screen

## 1) File System

- A glimpse of how Linux organizes its files

## 2) Meet Your Terminal

- What command prompt looks like

## 3) Basic Commands

- A very brief list of commands that you will most likely use daily

## 4) Terminal Like a Pro

- A few tips to help you use terminal more efficiently

- **Get some water**
- **Use restroom**
- **Ask questions**
  
- **Don't forget, the recording will be available at:**
  - **LSU HPC:** <https://www.hpc.lsu.edu/training/archive/tutorials.php>
  - **LONI HPC:** <https://hpc.loni.org/training/archive/tutorials.php>

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

## a) Why permission is important?

**Windows:**



User

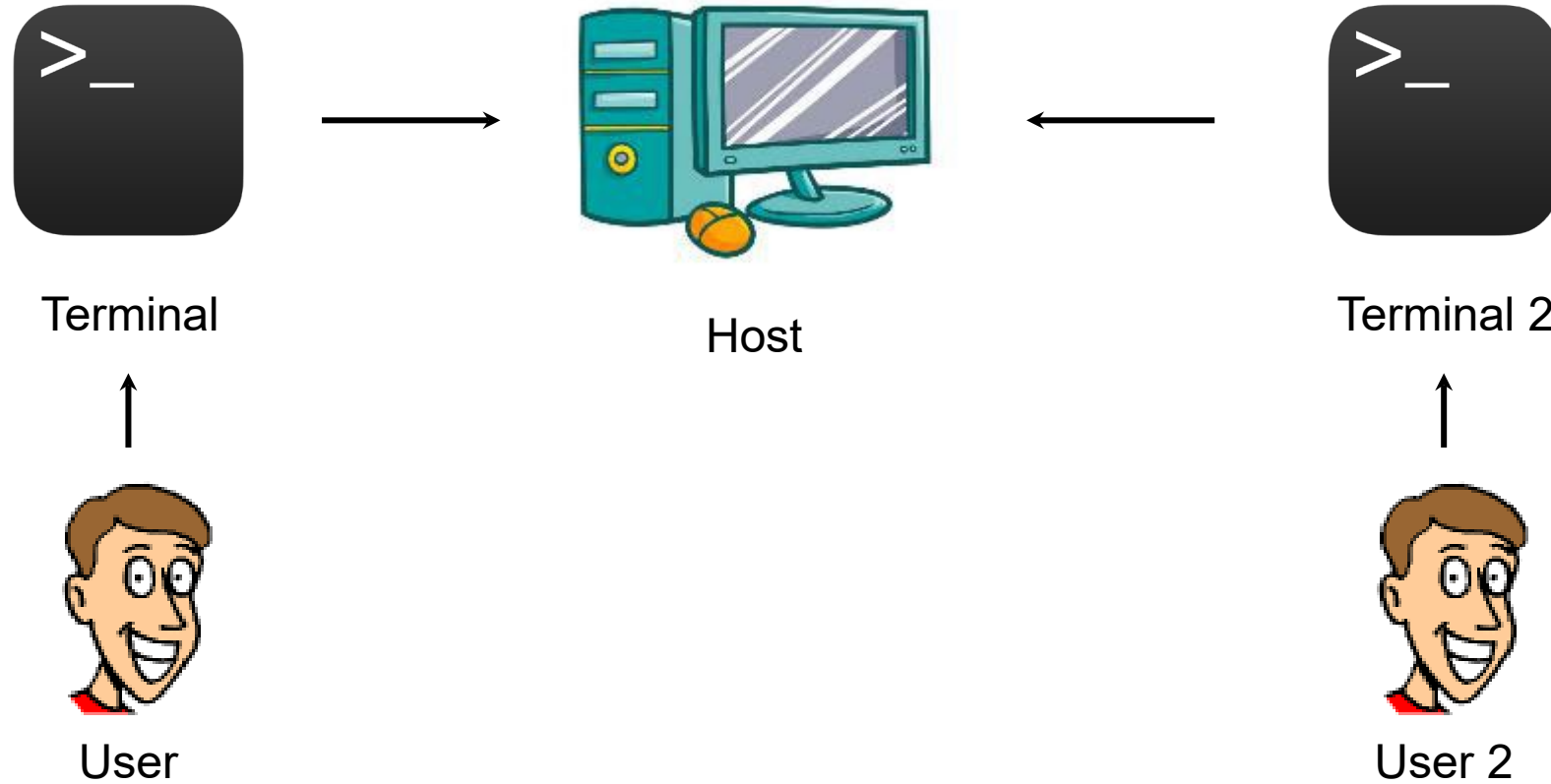


Machine

# 1) Understand Permissions

## a) Why permission is important?

Linux:



## b) Linux permission system

### i. User groups



User

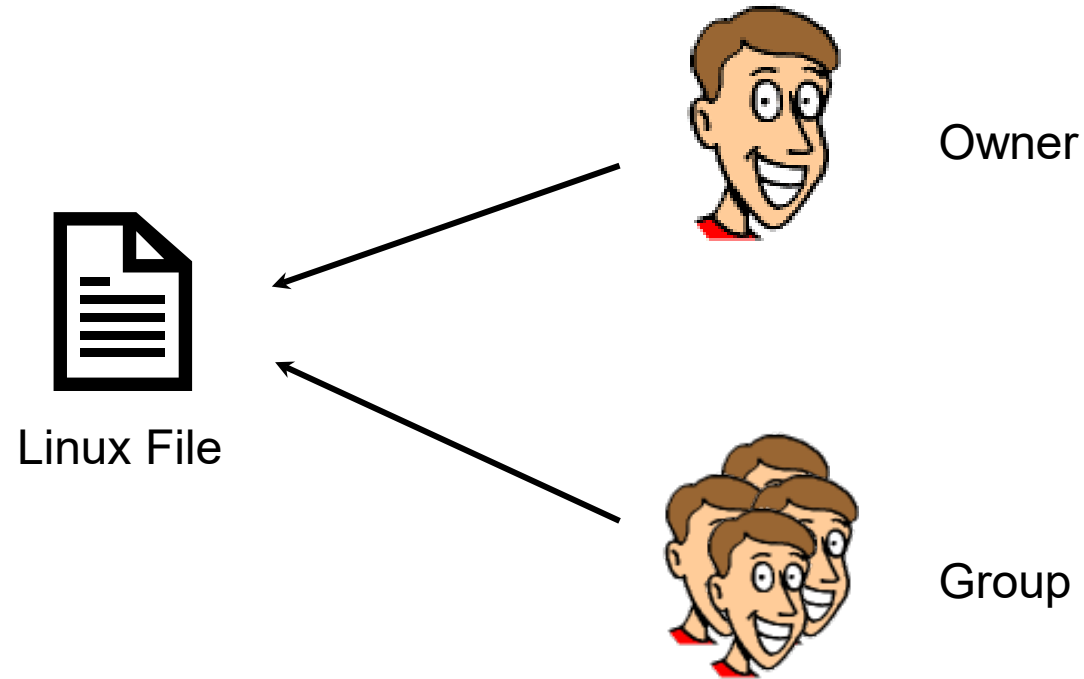


User Group

- Created by admins
- For file sharing among group members
- Find your groups: **groups**

## b) Linux permission system

### ii. File ownership



## b) Linux permission system

### iii. Permission matrix

	User (u)	Group (g)	Others (o)
Read (r)			
Write (w)			
Execute (x)			

## b) Linux permission system

### iii. Permission matrix

\* A typical non-executable plain text file

	User (u)	Group (g)	Others (o)
Read (r)			
Write (w)			
Execute (x)			

## b) Linux permission system

### iii. Permission matrix

\* A typical executable

	User (u)	Group (g)	Others (o)
Read (r)			
Write (w)			
Execute (x)			

## b) Linux permission system

iv. A slight difference b/w files and directories

Permission	File	Directory
Read (r)	Read the file content	ls files under the directory
Write (w)	Write to the file	Create / delete / move files or directories under the directory
Execute (x)	Execute the file (if executable)	cd into the directory

# 1) Understand Permissions

## b) Linux permission system

v. View permissions: `ls -l`

```
[jasonli3@qbd4 ollamaondemand]$ ls -l
total 140
-rw-r----- 1 jasonli3 sa_jasonli3 2432 Sep 10 11:30 arg.py
-rw-r----- 1 jasonli3 sa_jasonli3 2488 Jul 22  2025 chatsessions.py
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Sep  1 13:30 container
-rw-r----- 1 jasonli3 sa_jasonli3 6801 Oct 29 17:21 grblocks.css
-rw-r----- 1 jasonli3 sa_jasonli3 6361 Sep  2 19:03 head.html
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Aug 19 11:32 images
-rw-r----- 1 jasonli3 sa_jasonli3 1067 Jul 22  2025 LICENSE
-rwxr-xr-x 1 jasonli3 sa_jasonli3 71578 Dec  9 17:51 main.py
-rw-r----- 1 jasonli3 sa_jasonli3 7693 Dec  9 18:12 multimodal.py
```

Owner      Group

# 1) Understand Permissions

## b) Linux permission system

v. View permissions: `ls -l`

```
[jasonli3@qbd4 ollamaondemand]$ ls -l
total 140
-rw-r----- 1 jasonli3 sa_jasonli3 2432 Sep 10 11:30 arg.py
-rw-r----- 1 jasonli3 sa_jasonli3 2488 Jul 22 2025 chatsessions.py
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Sep 1 13:30 container
-rw-r----- 1 jasonli3 sa_jasonli3 6801 Oct 29 17:21 grblocks.css
-rw-r----- 1 jasonli3 sa_jasonli3 6361 Sep 2 19:03 head.html
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Aug 19 11:32 images
-rw-r----- 1 jasonli3 sa_jasonli3 1067 Jul 22 2025 LICENSE
-rwxr-xr-x 1 jasonli3 sa_jasonli3 71578 Dec 9 17:51 main.py
-rw-r----- 1 jasonli3 sa_jasonli3 7693 Dec 9 18:12 multimodal.py
```

Permissions

# 1) Understand Permissions

## b) Linux permission system

v. View permissions: `ls -l`

```
[jasonli3@qbd4 ollamaondemand]$ ls -l
total 140
-rw-r----- 1 jasonli3 sa_jasonli3 2432 Sep 10 11:30 arg.py
-rw-r----- 1 jasonli3 sa_jasonli3 2488 Jul 22  2025 chatsessions.py
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Sep  1 13:30 container
-rw-r----- 1 jasonli3 sa_jasonli3 6801 Oct 29 17:21 grblocks.css
-rw-r----- 1 jasonli3 sa_jasonli3 6361 Sep  2 19:03 head.html
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Aug 19 11:32 images
-rw-r----- 1 jasonli3 sa_jasonli3 1067 Jul 22  2025 LICENSE
-rwxr-xr-x 1 jasonli3 sa_jasonli3 71578 Dec  9 17:51 main.py
-rw-r----- 1 jasonli3 sa_jasonli3 7693 Dec  9 18:12 multimodal.py
```

Owner  
permissions

# 1) Understand Permissions

## b) Linux permission system

v. View permissions: `ls -l`

```
[jasonli3@qbd4 ollamaondemand]$ ls -l
total 140
-rw-r--r-- 1 jasonli3 sa_jasonli3 2432 Sep 10 11:30 arg.py
-rw-r--r-- 1 jasonli3 sa_jasonli3 2488 Jul 22  2025 chatsessions.py
drwxr-xr-x 2 jasonli3 sa_jasonli3 4096 Sep  1 13:30 container
-rw-r--r-- 1 jasonli3 sa_jasonli3 6801 Oct 29 17:21 grblocks.css
-rw-r--r-- 1 jasonli3 sa_jasonli3 6361 Sep  2 19:03 head.html
drwxr-xr-x 2 jasonli3 sa_jasonli3 4096 Aug 19 11:32 images
-rw-r--r-- 1 jasonli3 sa_jasonli3 1067 Jul 22  2025 LICENSE
-rwxr-xr-x 1 jasonli3 sa_jasonli3 71578 Dec  9 17:51 main.py
-rw-r--r-- 1 jasonli3 sa_jasonli3 7693 Dec  9 18:12 multimodal.py
```

Group  
permissions

# 1) Understand Permissions

## b) Linux permission system

v. View permissions: `ls -l`

```
[jasonli3@qbd4 ollamaondemand]$ ls -l
total 140
-rw-r----- 1 jasonli3 sa_jasonli3 2432 Sep 10 11:30 arg.py
-rw-r----- 1 jasonli3 sa_jasonli3 2488 Jul 22  2025 chatsessions.py
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Sep  1 13:30 container
-rw-r----- 1 jasonli3 sa_jasonli3 6801 Oct 29 17:21 grblocks.css
-rw-r----- 1 jasonli3 sa_jasonli3 6361 Sep  2 19:03 head.html
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Aug 19 11:32 images
-rw-r----- 1 jasonli3 sa_jasonli3 1067 Jul 22  2025 LICENSE
-rwxr-xr-x 1 jasonli3 sa_jasonli3 71578 Dec  9 17:51 main.py
-rw-r----- 1 jasonli3 sa_jasonli3 7693 Dec  9 18:12 multimodal.py
```

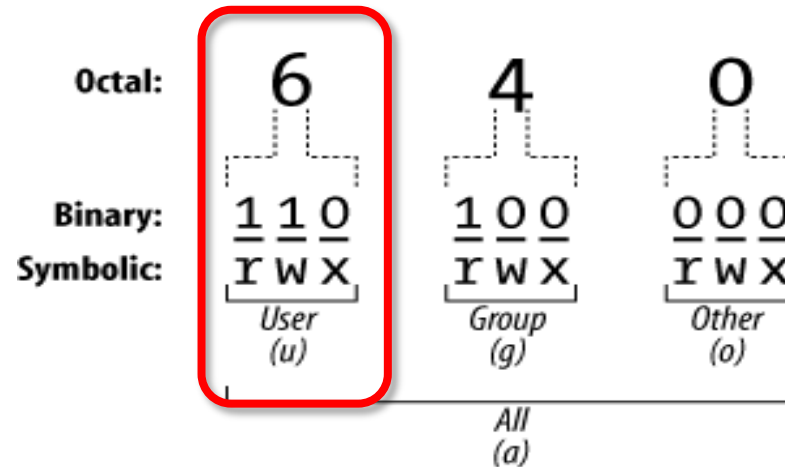
Others'  
permissions

# 1) Understand Permissions

## b) Linux permission system

v. View permissions: `ls -l`

Octal notation:

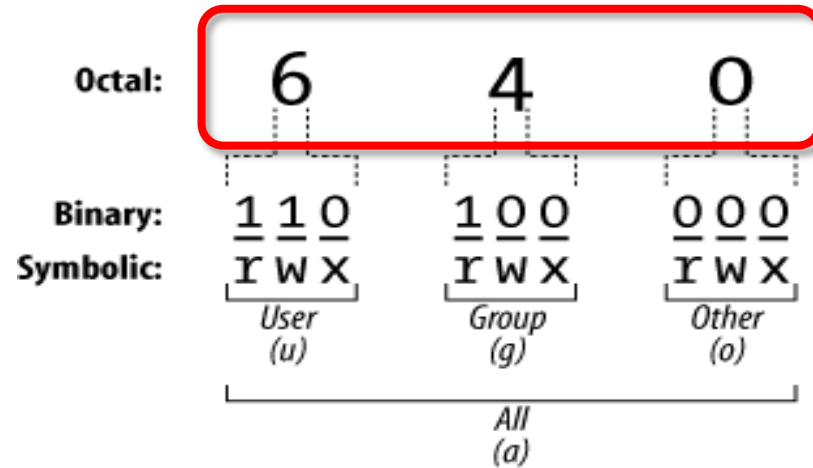


# 1) Understand Permissions

## b) Linux permission system

v. View permissions: `ls -l`

Octal notation:



# 1) Understand Permissions

## b) Linux permission system

v. View permissions: `ls -l`

```
[jasonli3@qbd4 ollamaondemand]$ ls -l
total 140
-rw-r----- 1 jasonli3 sa_jasonli3 2432 Sep 10 11:30 arg.py
-rw-r----- 1 jasonli3 sa_jasonli3 2488 Jul 22 2025 chatsessions.py
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Sep 1 13:30 container
-rw-r----- 1 jasonli3 sa_jasonli3 6801 Oct 29 17:21 grblocks.css
-rw-r----- 1 jasonli3 sa_jasonli3 6361 Sep 2 19:03 head.html
drwxr-x--- 2 jasonli3 sa_jasonli3 4096 Aug 19 11:32 images
-rw-r----- 1 jasonli3 sa_jasonli3 1067 Jul 22 2025 LICENSE
-rwxr-xr-x 1 jasonli3 sa_jasonli3 71578 Dec 9 17:51 main.py
-rw-r----- 1 jasonli3 sa_jasonli3 7693 Dec 9 18:12 multimodal.py
```

What's the octal code of this file?

**755**

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

## 2) Change Permissions

- `chmod` (“change mode”)

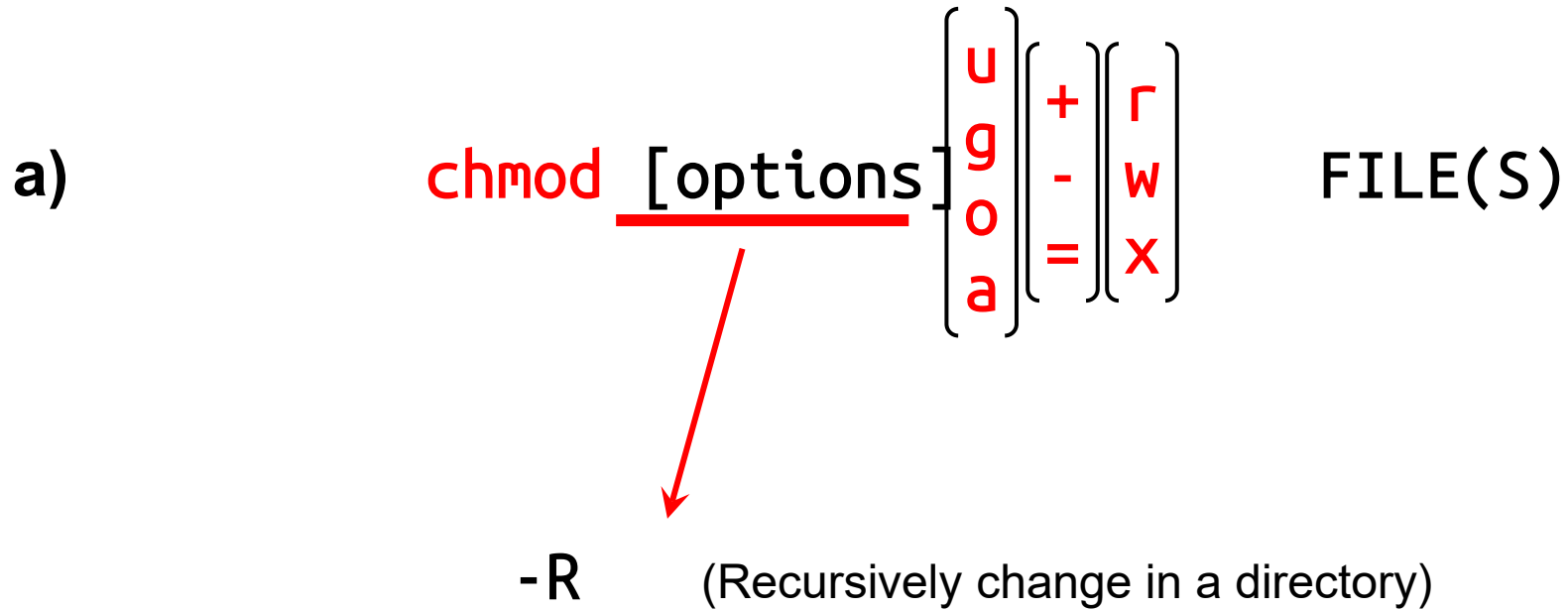
a) `chmod` [options]  $\begin{pmatrix} u \\ g \\ o \\ a \end{pmatrix} \begin{pmatrix} + \\ - \\ = \end{pmatrix} \begin{pmatrix} r \\ w \\ x \end{pmatrix}$  FILE(S)

E.g.,

- `chmod u+x FILE` (for user, add execute permission)
- `chmod g-w FILE` (for group, remove write permission)
- `chmod o= FILE` (for others, set permission to none)
- `chmod a-x FILE` (for **all**, remove execute permission)

# 2) Change Permissions

- `chmod` (“change mode”)



- `chmod` (“change mode”)

b) `chmod [options] OCTAL-CODE FILE(S)`

E.g.,

– `chmod 755 FILE` (set mode to 755)

## 1) Understand Permissions

– Permission matrix:

	User (u)	Group (g)	Others (o)
Read (r)	✓	✓	✗
Write (w)	✓	✗	✗
Execute (x)	✓	✓	✗

## 2) Change Permissions

– Multiple ways of using **chmod**

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

- **This is HPC training, after all...**
  - 1) Tools You Can Use on HPC
  - 2) Tools You Cannot Use on HPC

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

# 1) Those You Can Use on HPC

## a) Misc commands

Command	Option	Augument	Description
<b>du</b> ("disk usage")		FILE(S)	Calculate disk usage
	-h	FILE(S)	Show human-readable file size
	-s	FILE(S)	Only summarize total usage
<b>top</b>			Dynamically display system summary and processes
<b>ln</b> ("link")	-s	TARGET LINK	Create symbolic link (symlink), where LINK is pointing to TARGET
<b>wget</b> ("www get")		URL	Download file to current directory

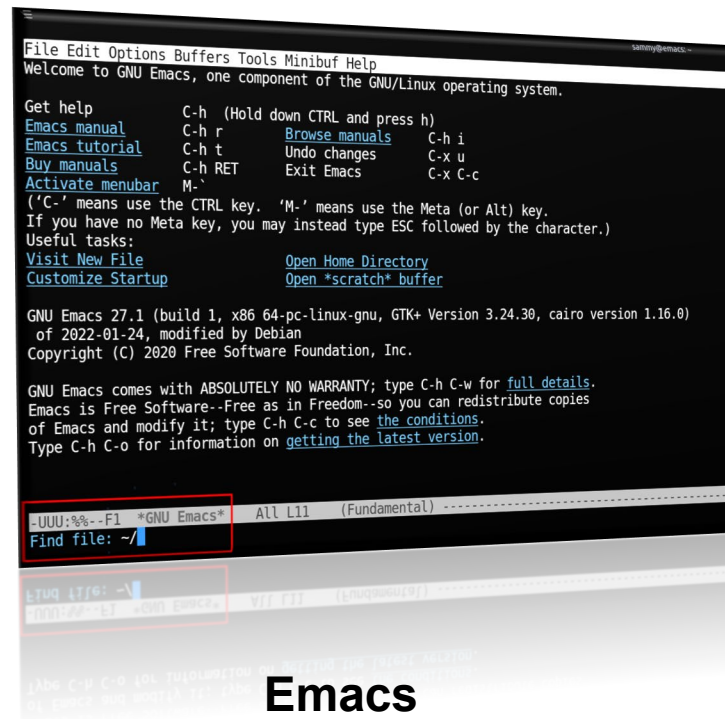
# 1) Those You Can Use on HPC

## b) File editors

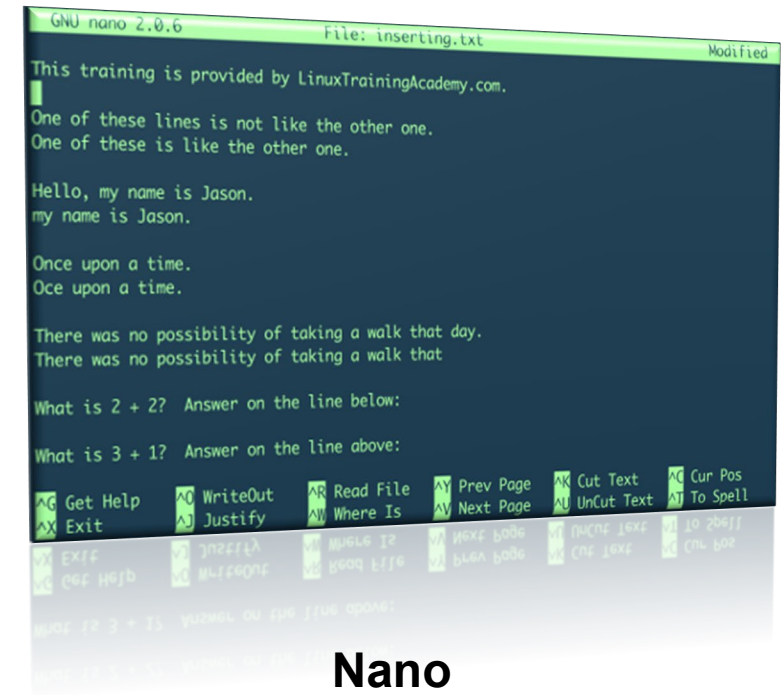
- You may be used to VS Code or Notepad++.
- But with terminal access to HPC, you might as well get used to at least one terminal-based editor



VIM (VI)



Emacs



Nano

## c) Compress / Decompress

Command	Option	Augument	Description
tar	-cf	ARCHIVE FILE(S)	Compress FILE(S) to ARCHIVE • Choose any archive format . (Popular choice: .tar.gz)
	-xf	ARCHIVE	Extract from ARCHIVE
	-tf	ARCHIVE	Inspect files in ARCHIVE
zip		ARCHIVE FILE(S)	Compress FILE(S) to a zip ARCHIVE
unzip		ARCHIVE	Extract from a zip ARCHIVE

## d) `~/ .bashrc`

- Automatically executed every time you start a Shell session
- Used for automatically set up environment upon login
  - Set environmental variables
  - Set alias
  - Execute scripts
  - Customize prompt
  - ...
- Other Shell types uses different files
  - E.g., `~/ .tcshrc` for `tcsh`

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

## 2) Those You Cannot Use on HPC

- **Wait a minute... Why bother introduce them?**
- **Reasons:**
  - i. Next time you search for a solution and some one suggest these answers, you know you cannot use them on HPC!
  - ii. You can still use them on your own Linux machine
  - iii. Most importantly, though you cannot use them directly on HPC, there are tools allow you to effectively use them, such as **containers** !

### a) sudo

- Run command as **root** user (“superuser do”)
- E.g.,

```
mkdir /usr/local/test
```



```
sudo mkdir /usr/local/test
```

- *Will fail (Permission denied)*
- **Problem:** HPC users do **NOT** have root privilege!

- *Will succeed (if you are a root user)*

### b) Certain package management

- apt / apt-get / dnf / yum / ...
- Convenient & hassle free
- **Problem:** Usually needs to write into system paths and run with `sudo`

```
sudo apt install  
intree
```

- Even though you cannot use them directly on HPC, you can use them in **containers** and run containers on our HPC!

\*View the latest  
**Magic Tools to Install - Manage Software Part 1: Singularity Container**  
at <https://www.hpc.lsu.edu/training/tutorials/past/>  
or <https://hpc.loni.org/training/tutorials/past/>

## 1. Meet Linux

- 1) What's Linux?
- 2) Many Faces of Linux
- 3) Get Linux

## 2. Basic Operations

- 1) File System
- 2) Meet Your Terminal
- 3) Basic Commands
- 4) Terminal Like a Pro

## 3. File Permissions

- 1) Understand Permissions
- 2) Change Permissions

## 4. Useful Tools

- 1) Those You Can Use on HPC
- 2) Those You Cannot Use on HPC

- **Contact user services**

- Email Help Ticket:

- [sys-help@hpc.lsu.edu](mailto:sys-help@hpc.lsu.edu)

- [sys-help@loni.org](mailto:sys-help@loni.org)

- Telephone Help Desk: +1 (225) 578-0900