Introduction to Linux

Jason Li

HPC User Services

LSU HPC / LONI

sys-help@loni.org

Louisiana State University, Baton Rouge

Jan 21, 2026

Before we start…

- **Recordings 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

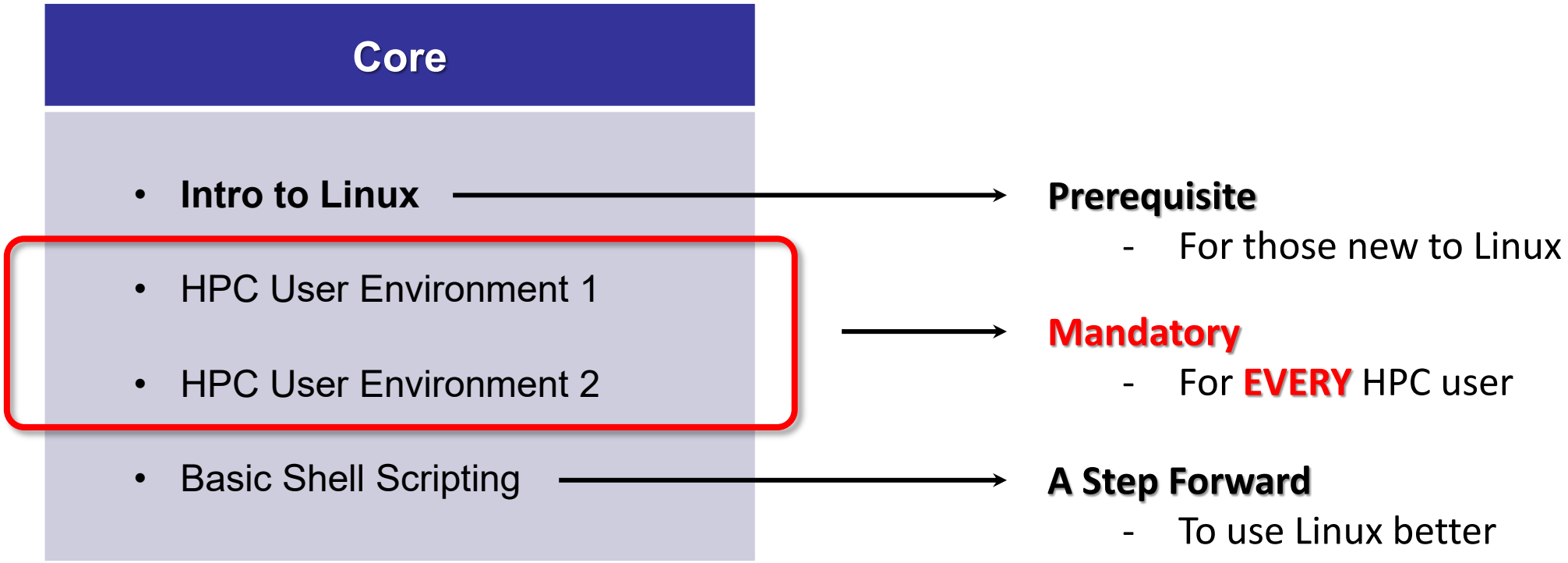
Prologue

- **Research Computing Trainings**

Core	Interested-based
• **Intro to Linux** • HPC User Environment 1 • HPC User Environment 2 • Basic Shell Scripting	• Open OnDemand • Intro to Python • Magic Tools 1: Singularity Container • Magic Tools 2: Conda Environment • GNU Parallel • Molecular Dynamics • …

Prologue

- **Research Computing Trainings**



Prologue

- **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 

Outlines

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

Outlines

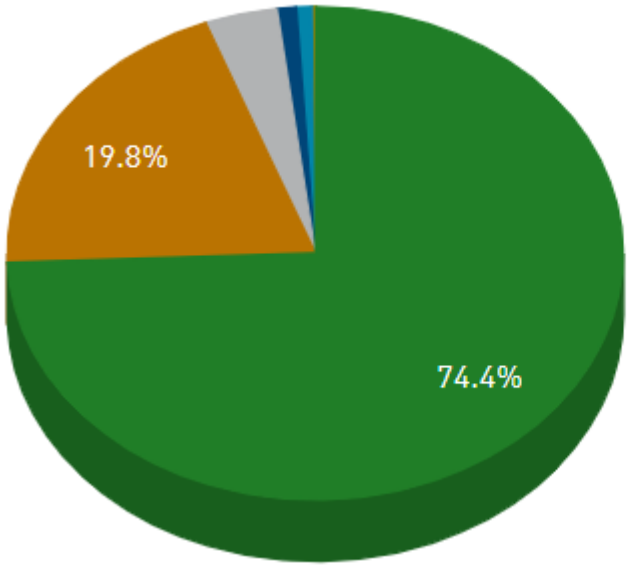
1. **Meet Linux**
   1) What's Linux?
   2) Many Faces of Linux
   3) Get Linux
2. **Basic Operations**
   1) File System
   2) Meet Your Terminal
   3) Basic Commands
   4) Terminal Like a Pro
3. **File Permissions**
   1) Understand Permissions
   2) Change Permissions
4. **Useful Tools**
   1) Those You Can Use on HPC
   2) Those You Cannot Use on HPC

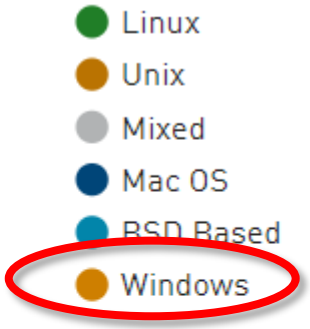
1) What's Linux?

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

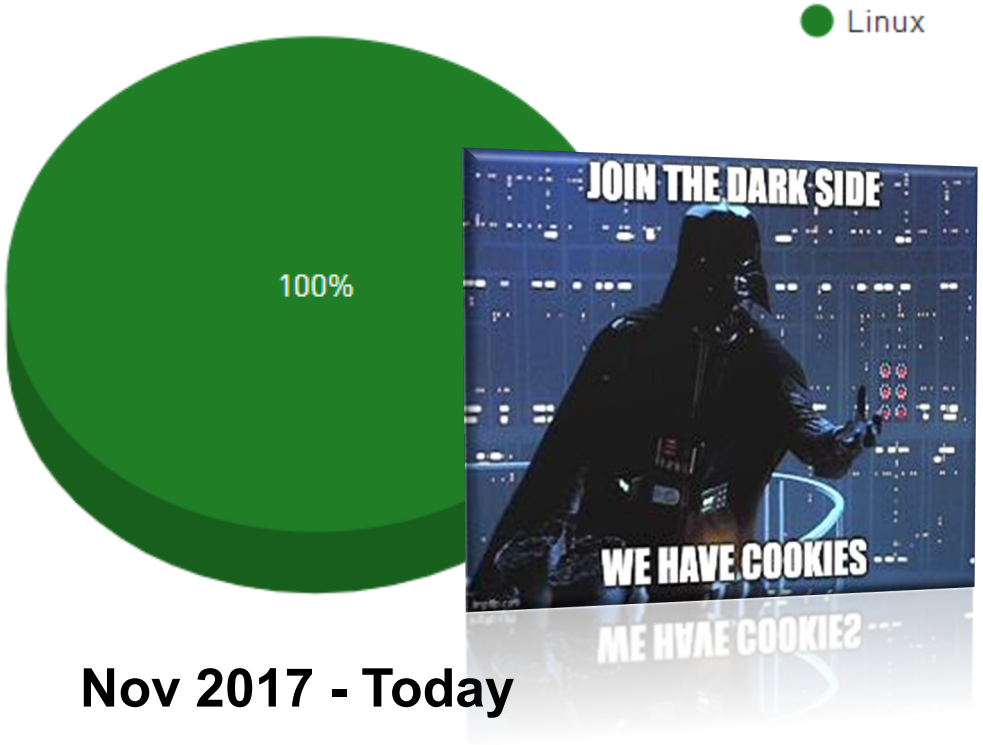
1) What's Linux?

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





Windows' swan song!

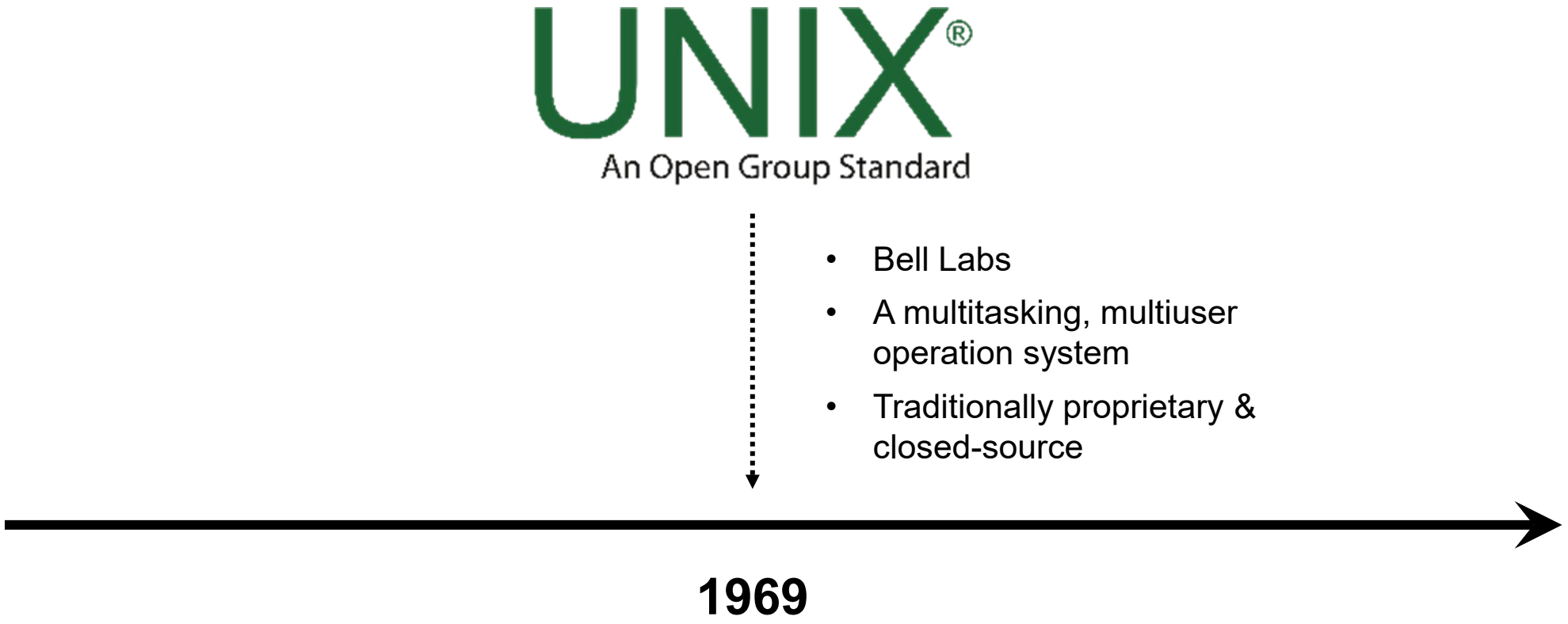


Nov 2005

Nov 2017 - Today

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

- **Linux – A journey to complete freedom!**



1) What's Linux?

- **Linux – A journey to complete freedom!**

UNIX®
An Open Group Standard

- Richard Stallman, "***The GNU Manifesto***"
  - "**GNU**" = "Gnu's Not Unix!"
- Create a "*complete Unix-compatible software system*" and "*give it away free to everyone*"
- Only missing good, free, **lower-level** OS components

1969 **1983**

[1] https://www.gnu.org/gnu/manifesto.html

1) What's Linux?

- **Linux – A journey to complete freedom!**

UNIX®
An Open Group Standard

- **Linus Torvald** (THE MAN, THE LEGEND!)
- Developed and released his own OS, which later became **Linux kernel**
- **GPL license** (free to download, edit, and redistribute)

1969 **1983** **1991**

1) What's Linux?

- **Linux – A journey to complete freedom!**

UNIX®
An Open Group Standard

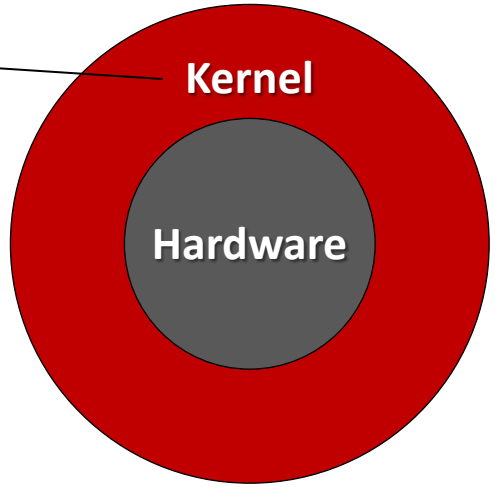
GNU Project's free software + **Linux kernel** = "Linux"

1969 **1983** **1991**

1) What's Linux?

- **A closer look at Linux architecture**

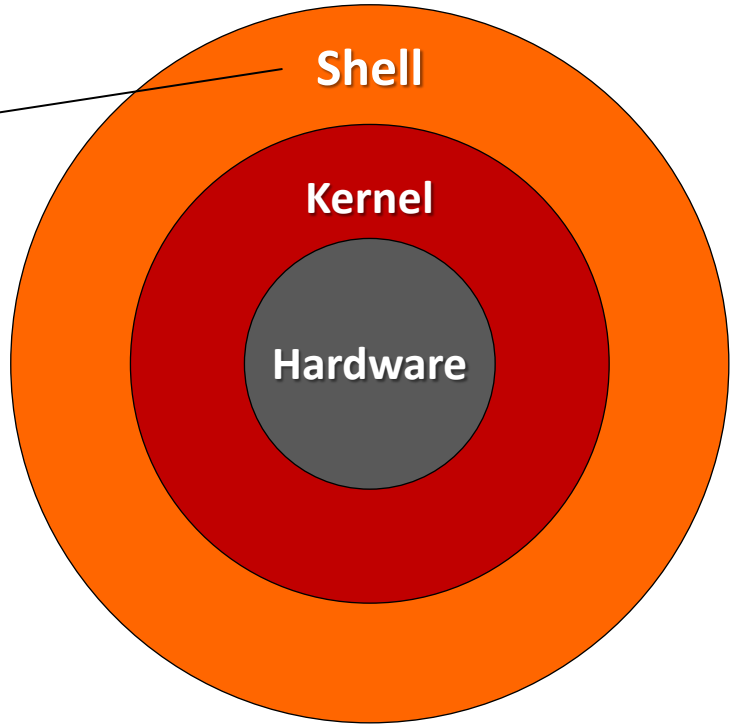
- Contains (**low-level**):
  - Resource management
  - Process management
  - Device Drivers
  - System Calls
  - …
- **Linux Kernel**, the part Linus Torvald developed and released for free



1) What's Linux?

- **A closer look at Linux architecture**

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

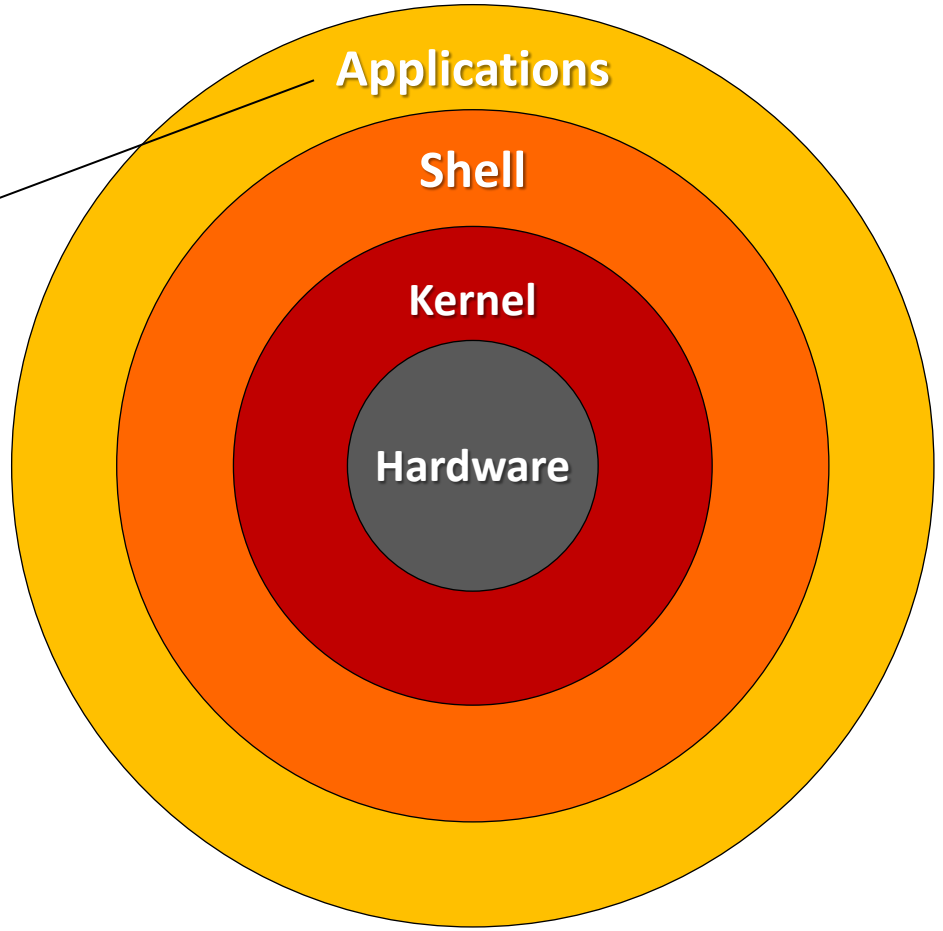


*Join us on 2/11/2026 for **Basic Shell Scripting** !

1) What's Linux?

- **A closer look at Linux architecture**

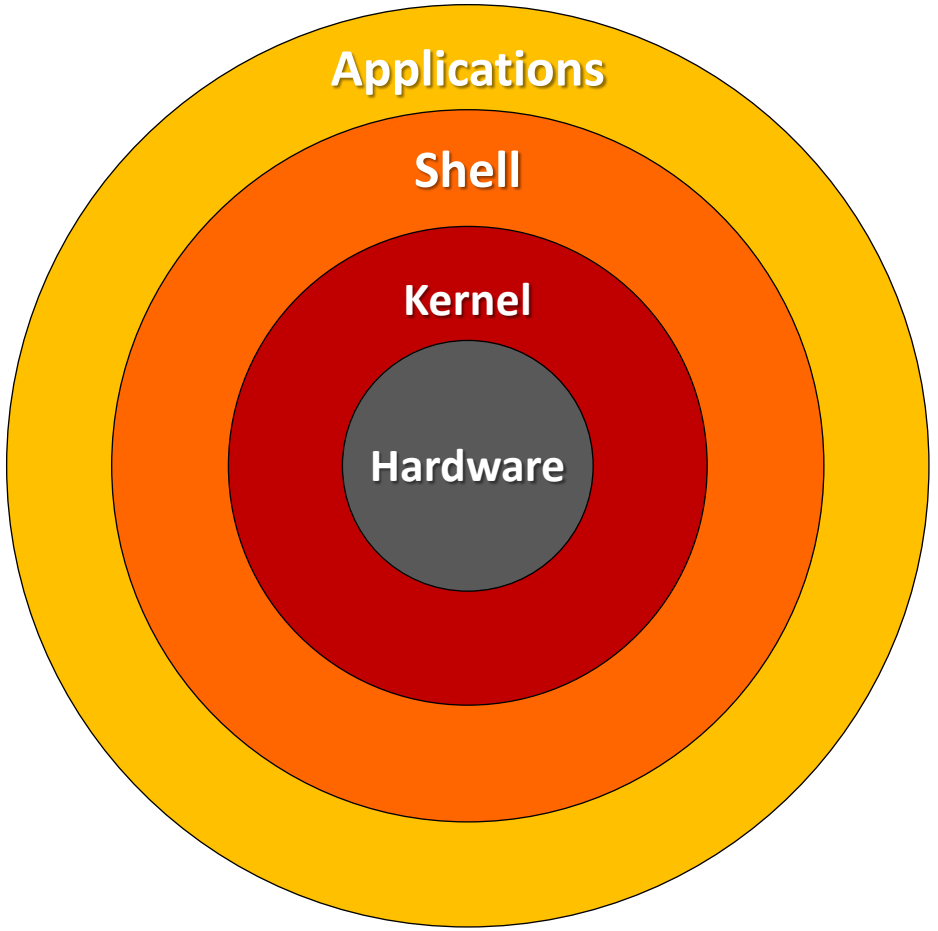
  - Contains (**highest-level**):
    - File browser
    - Text editor
    - Multimedia
    - Internet browser
    - Office suites
    - …
  - **GNU Project** & beyond



1) What's Linux?

- **A closer look at Linux architecture**

"Linux"



1) What's Linux?

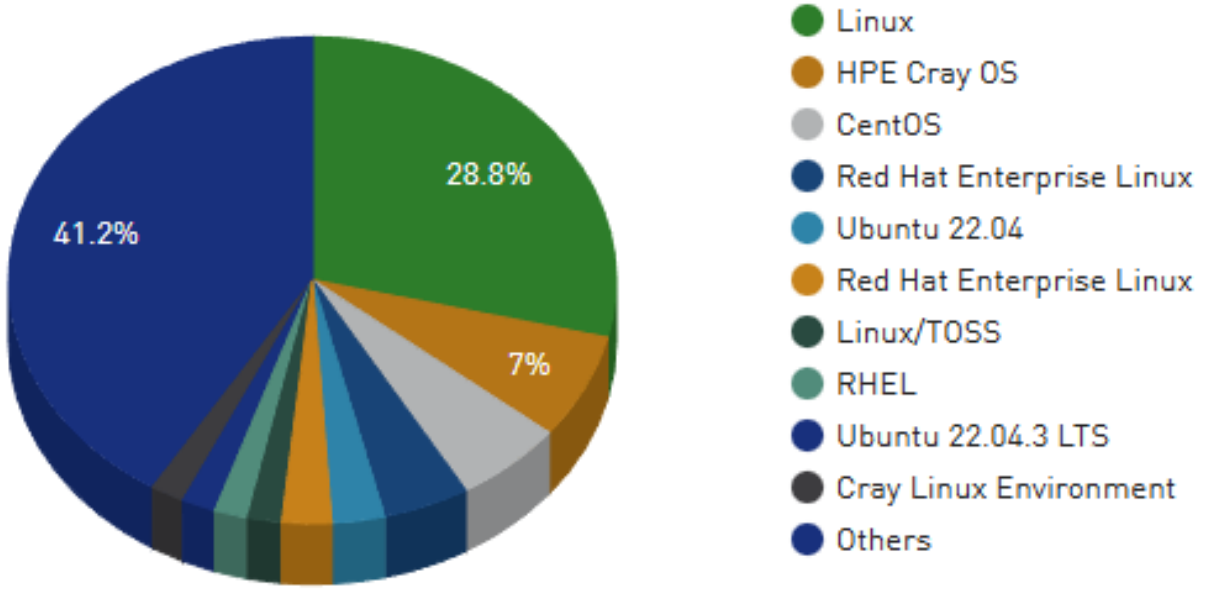
- **Key to a legendary project**

	Visionary leader (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

Outlines

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/

a) Distributions (Distro)

UNIX®
An Open Group Standard

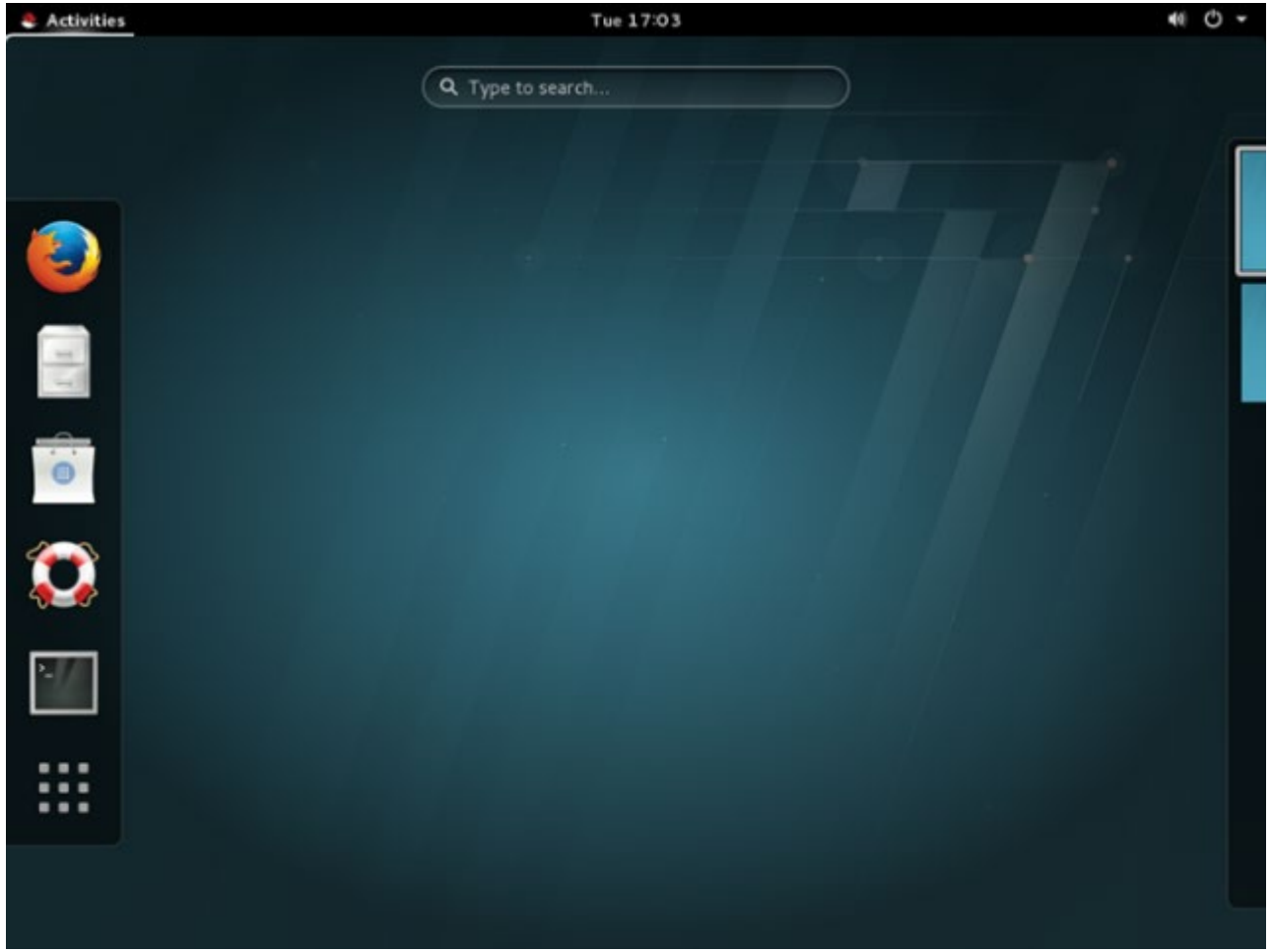
GNU Project's free software + Linux kernel = "Linux"

1969 1983 1991

2) Many Faces of Linux

a) Distributions (Distro)

Red Hat
(one of the industrial favorites)



2) Many Faces of Linux

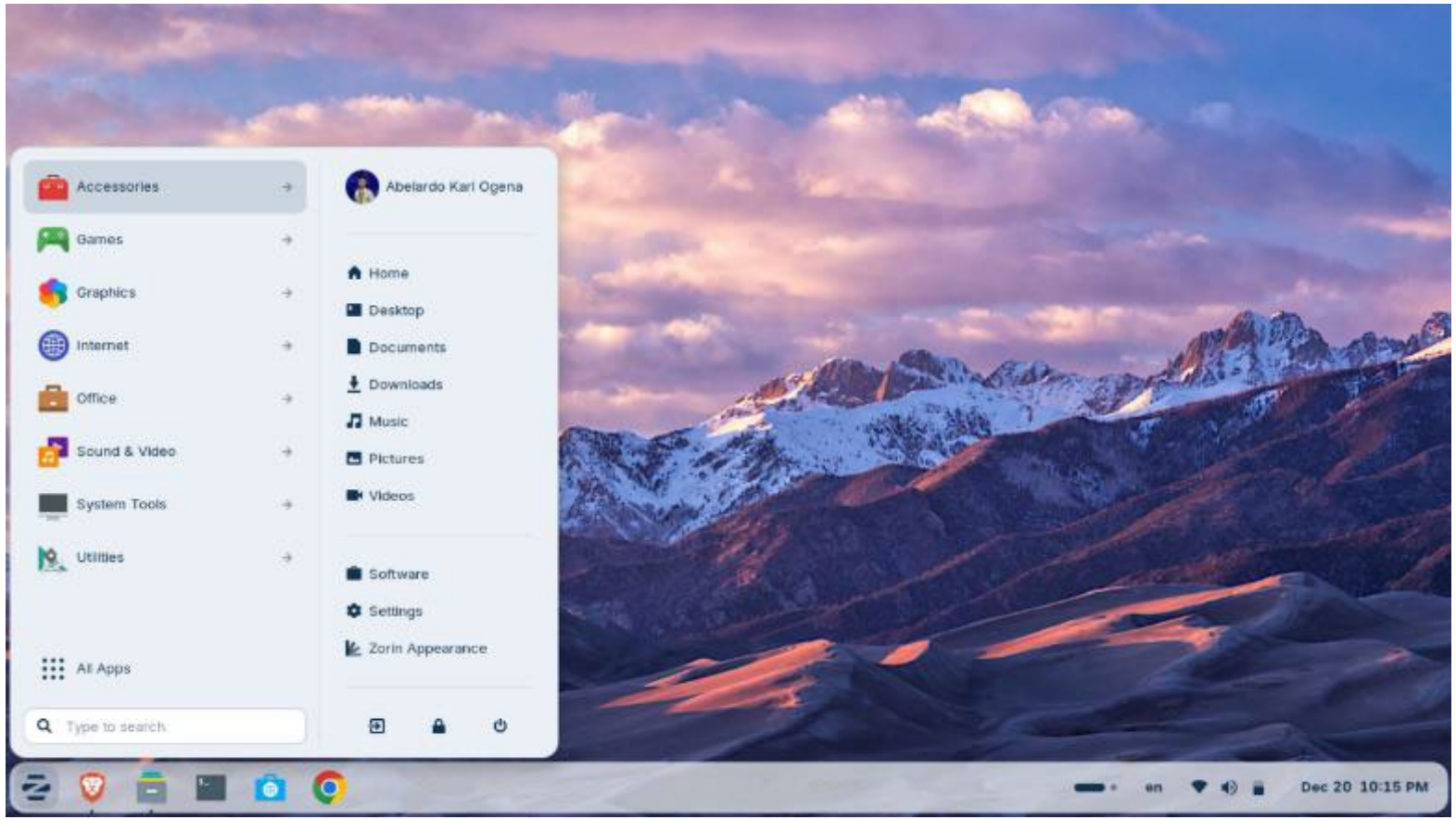
a) Distributions (Distro)

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)

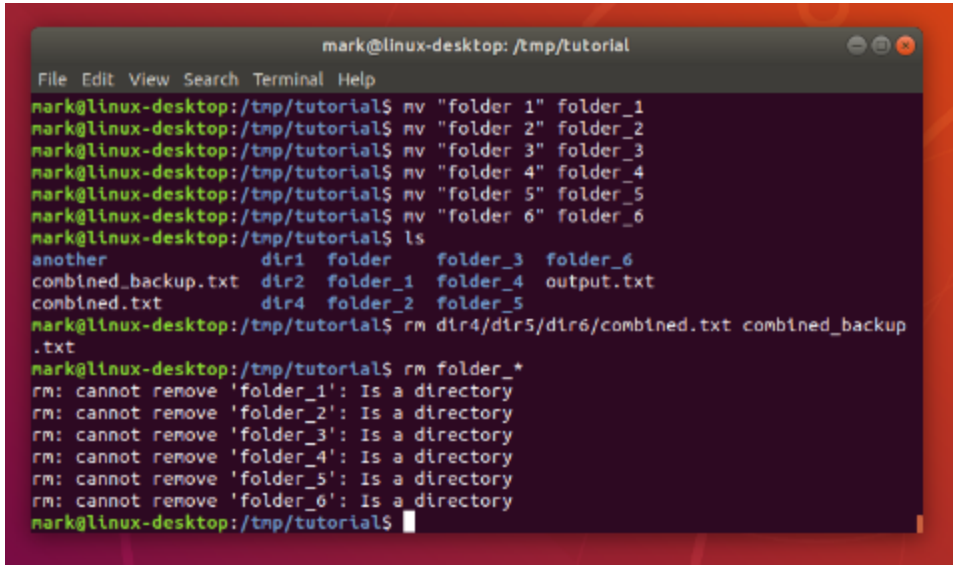
Question: Do you know what distro LSU / LONI HPC use?

Red Hat Enterprise Linux (RHEL)

2) Many Faces of Linux

b) Desktop vs Terminal

Desktop (GUI)



Terminal (Command-line)

b) Desktop vs Terminal

We will only focus on terminal!

Outlines

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:

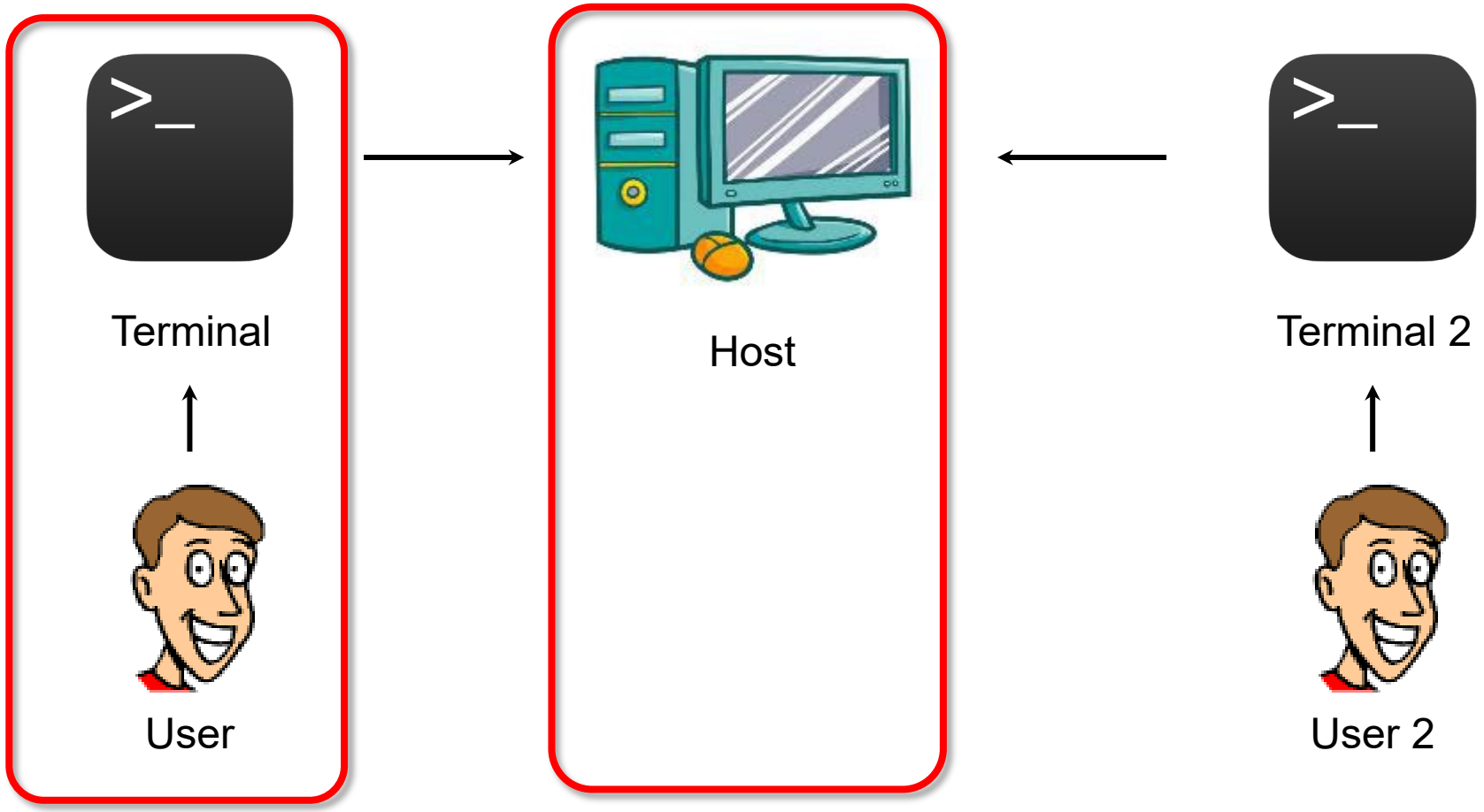
User

Machine

3) Get Linux

a) **You do NOT have to!!**

Linux:



3) Get Linux

a) **You do NOT have to!!**

A terminal is all you need!

Your OS …	Your choices…
Linux / Mac / ChromeOS	Terminal
Windows	**MobaXterm**, Putty, …
iOS / iPadOS	Termius, Blink Shell, …
Android	Termux, …

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

3) Get Linux

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
```

Summary

- **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!

Outlines

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

Outlines

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

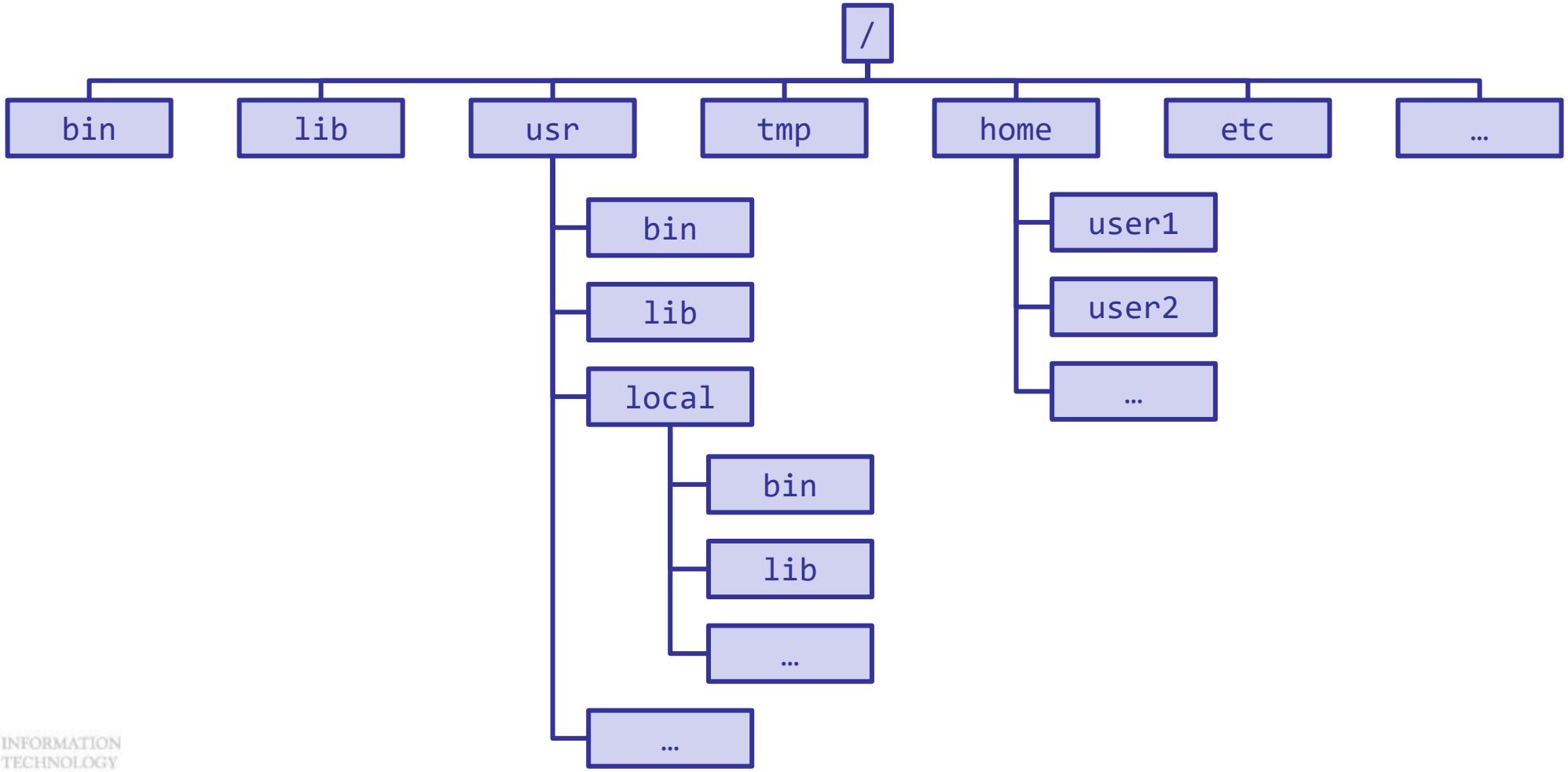
2. **Basic Operations**
   1) File System
   2) Meet Your Terminal
   3) Basic Commands
   4) Terminal Like a Pro

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

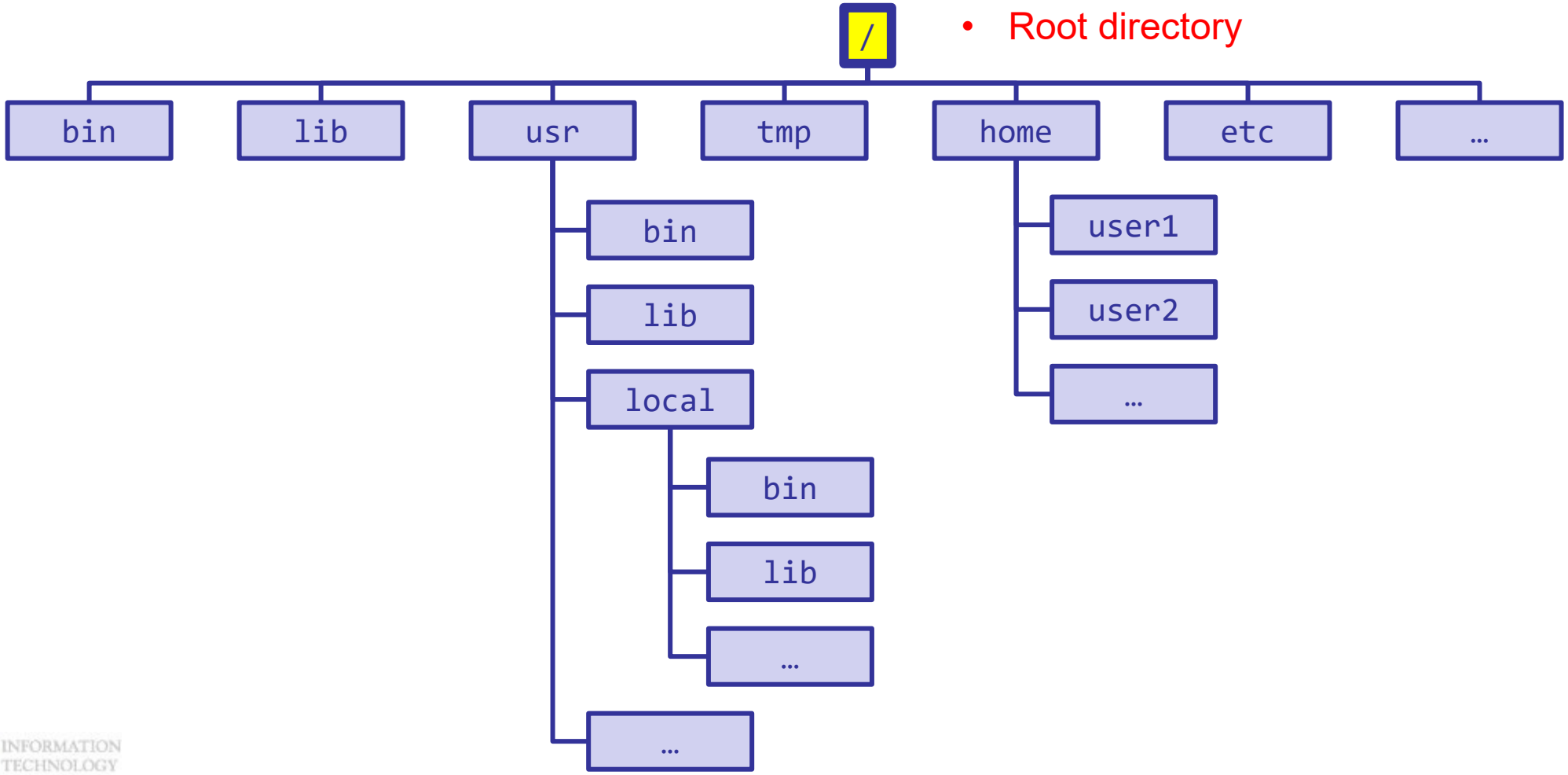
1) File System

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

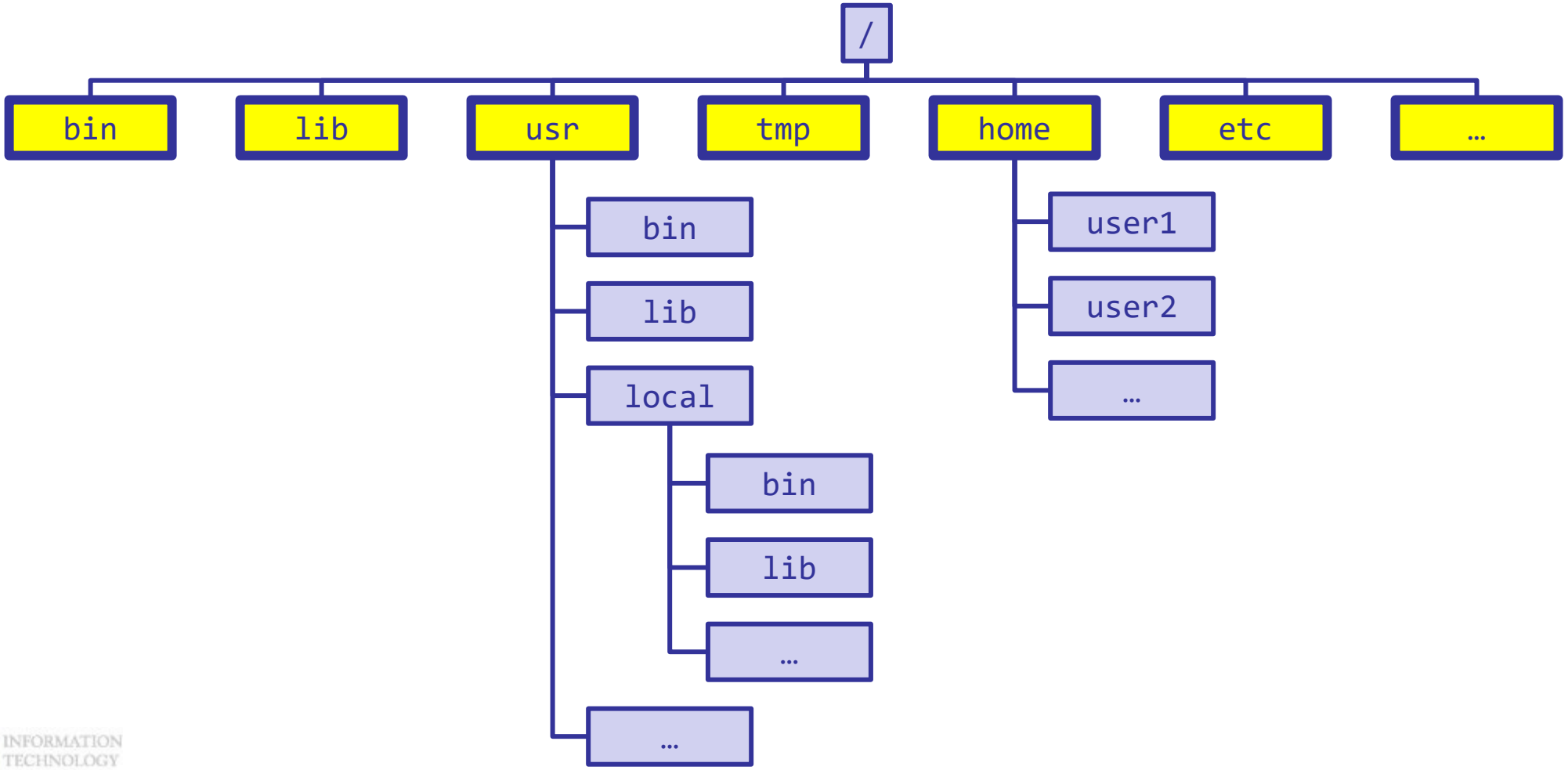


1) File System

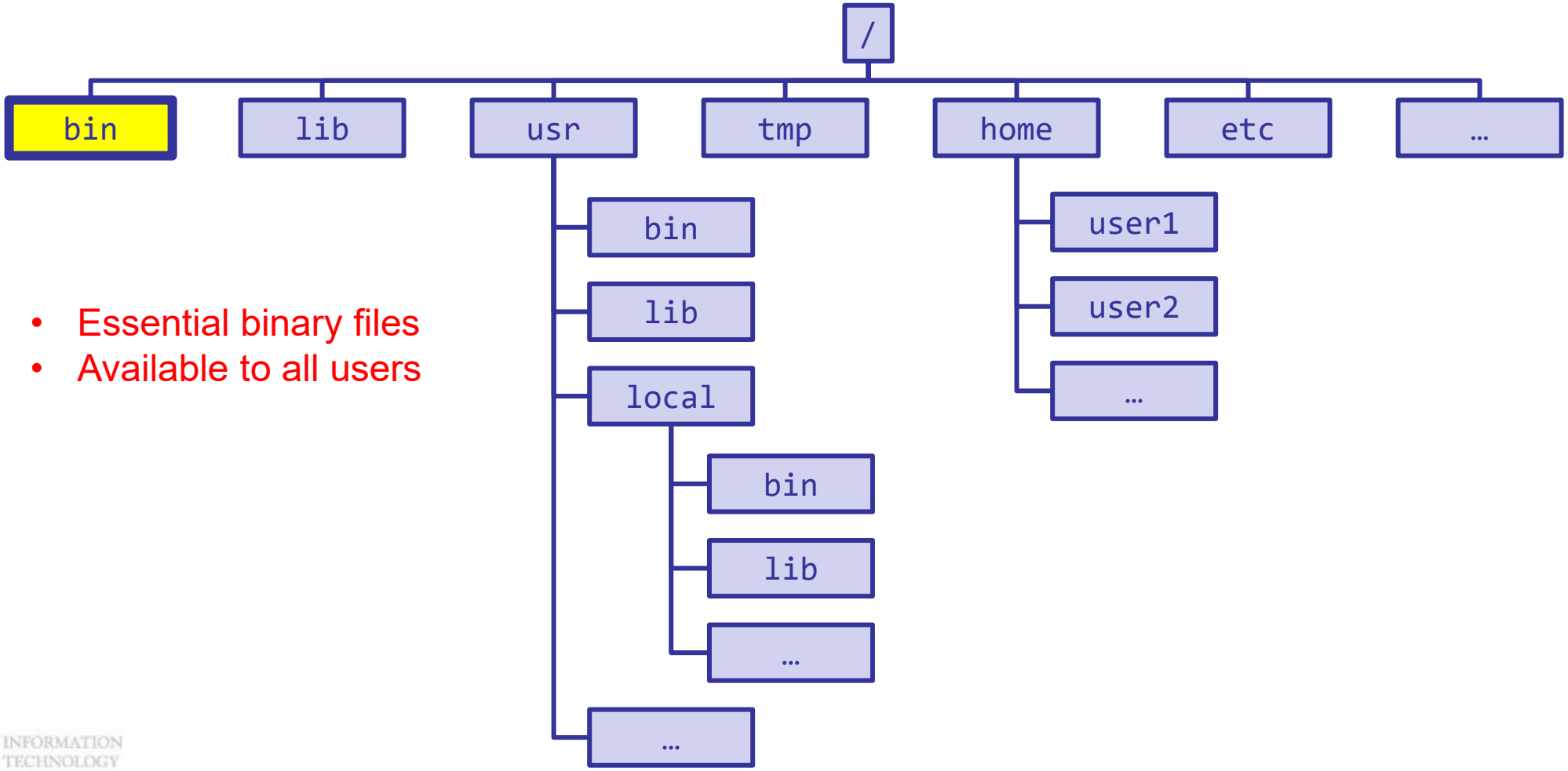
a) Filesystem Hierarchy Standard (FHS) : An inverted tree



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



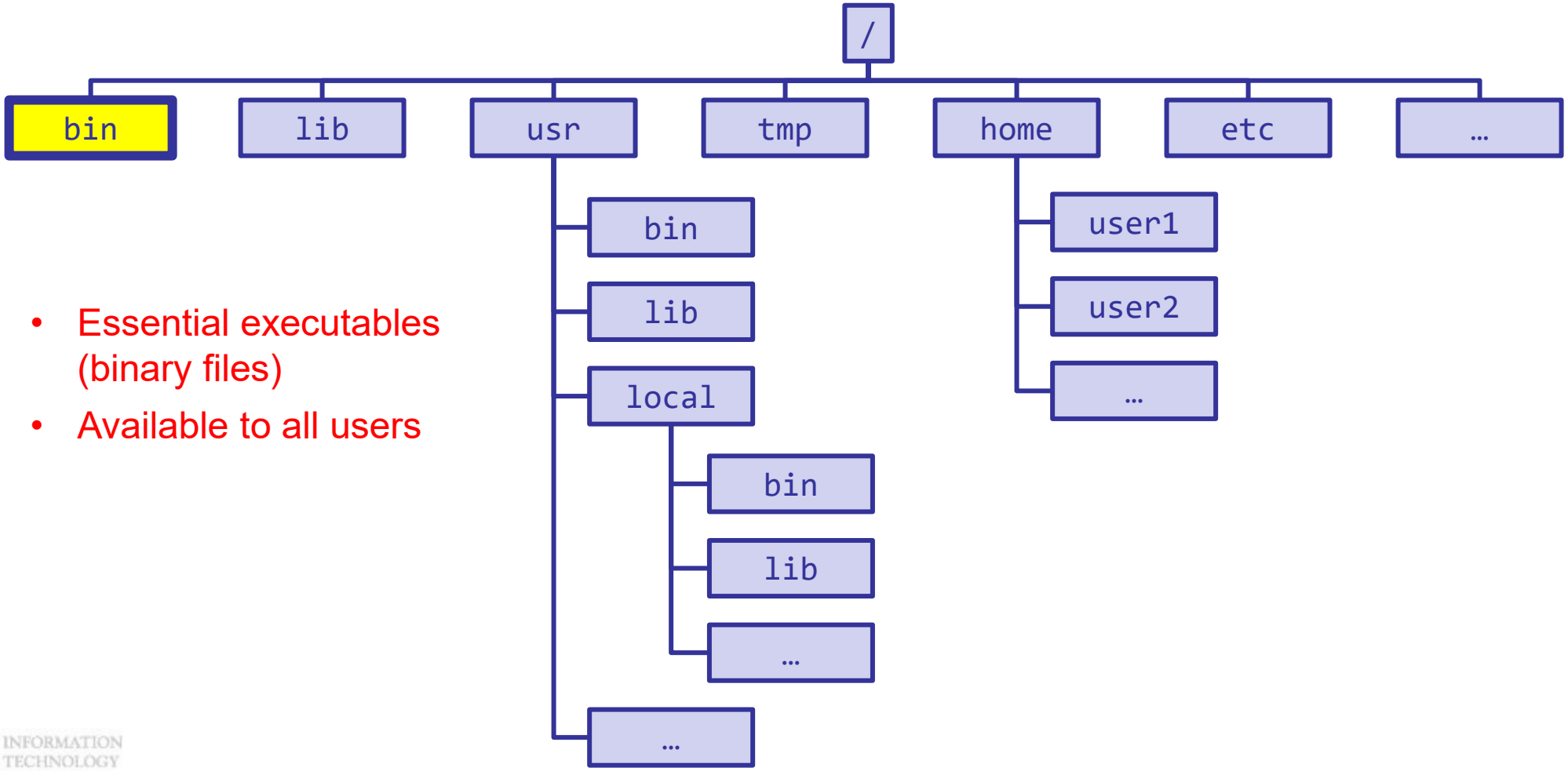
a) Filesystem Hierarchy Standard (FHS) : An inverted tree



- Essential binary files
- Available to all users

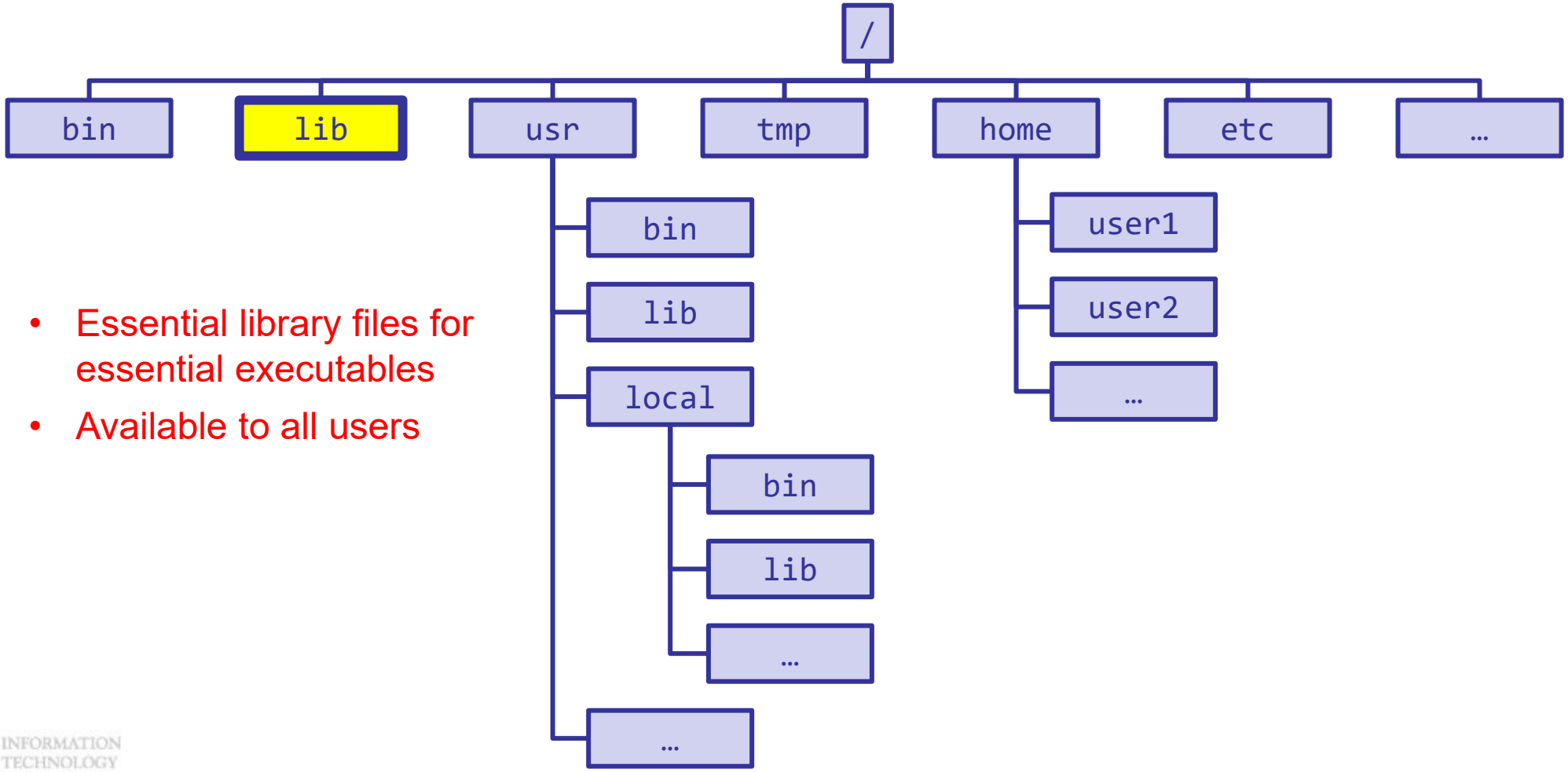
1) File System

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



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

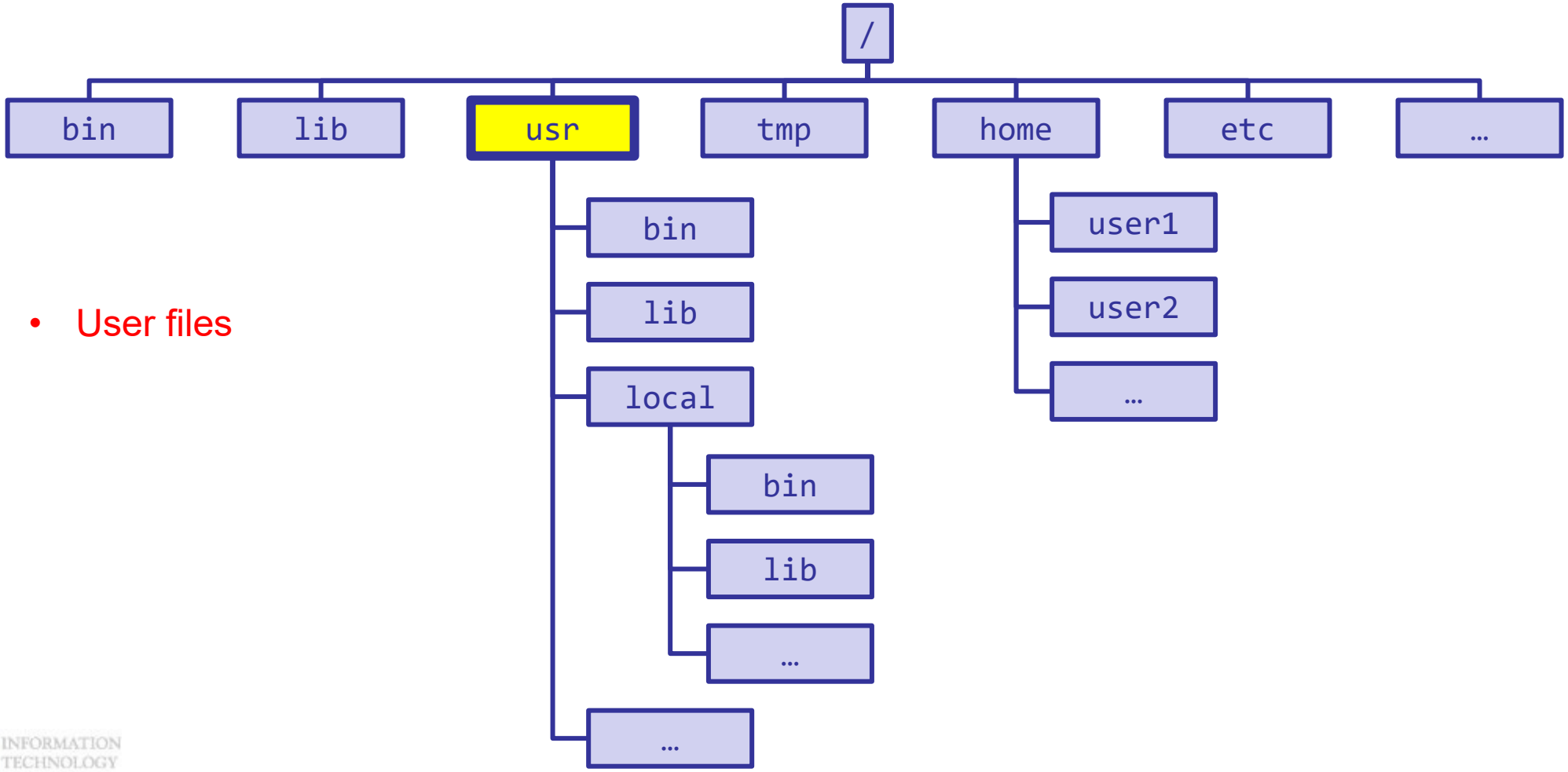
a) Filesystem Hierarchy Standard (FHS) : An inverted tree



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

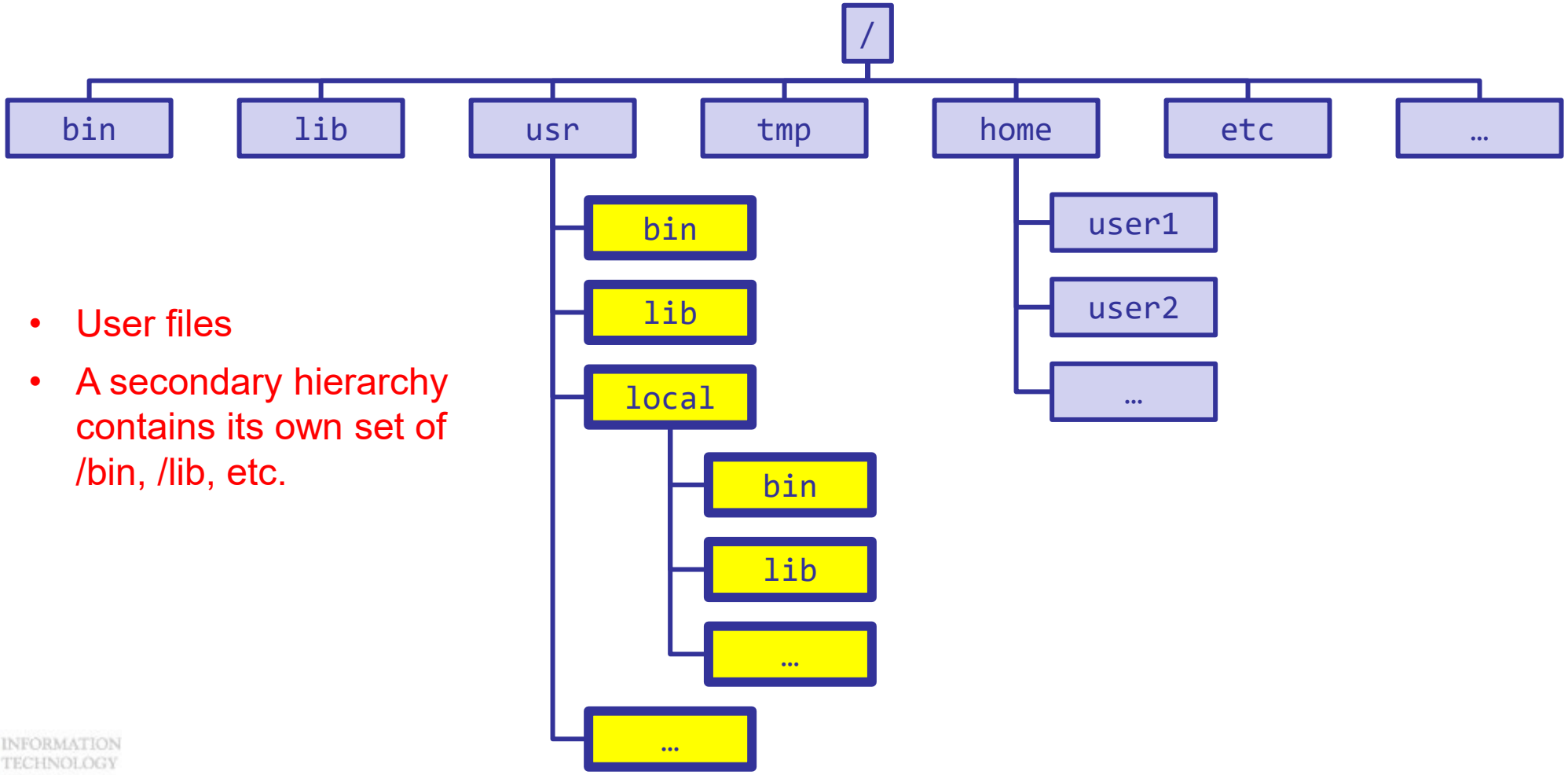
1) File System

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



- User files

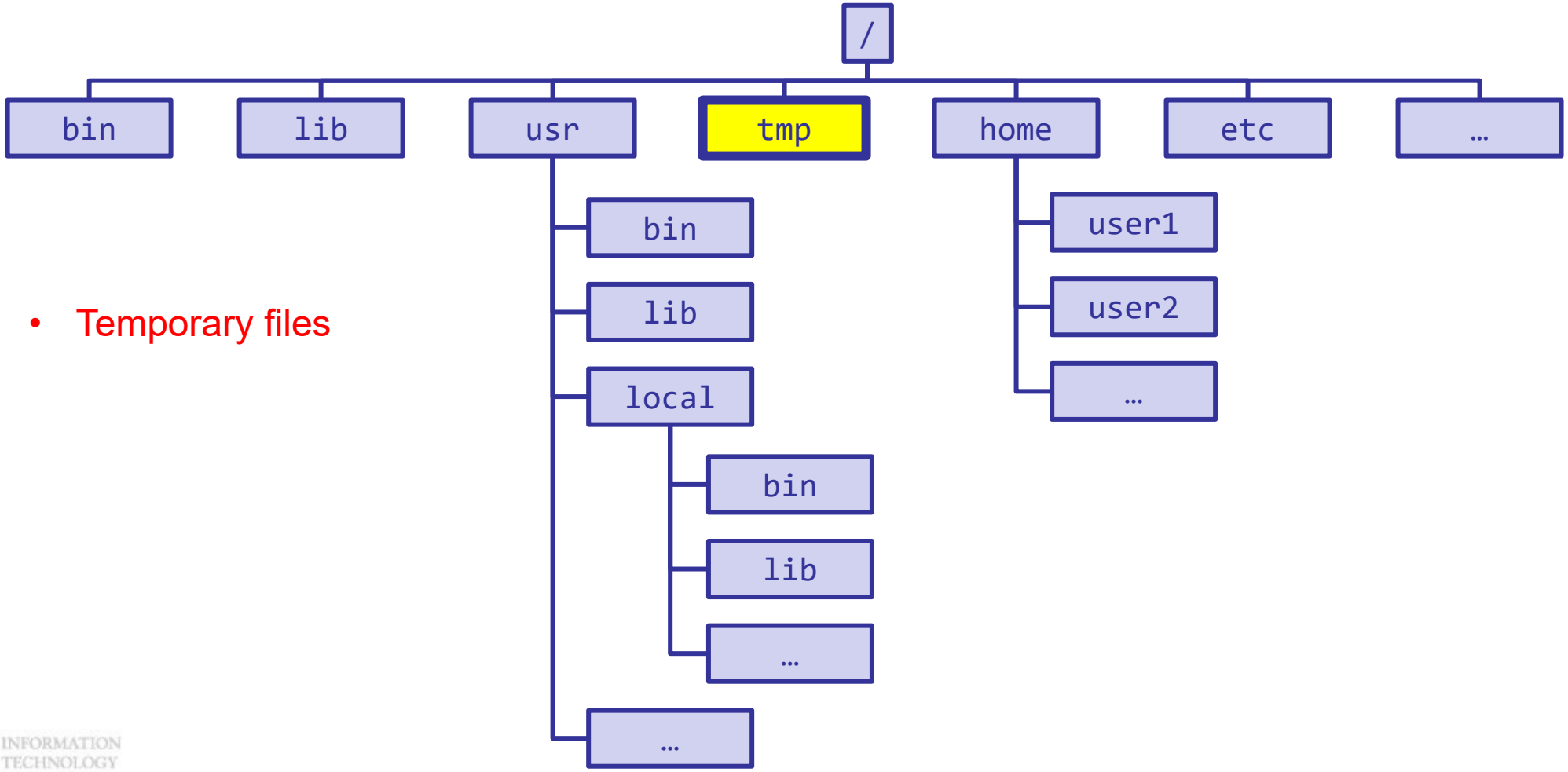
a) Filesystem Hierarchy Standard (FHS) : An inverted tree



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

1) File System

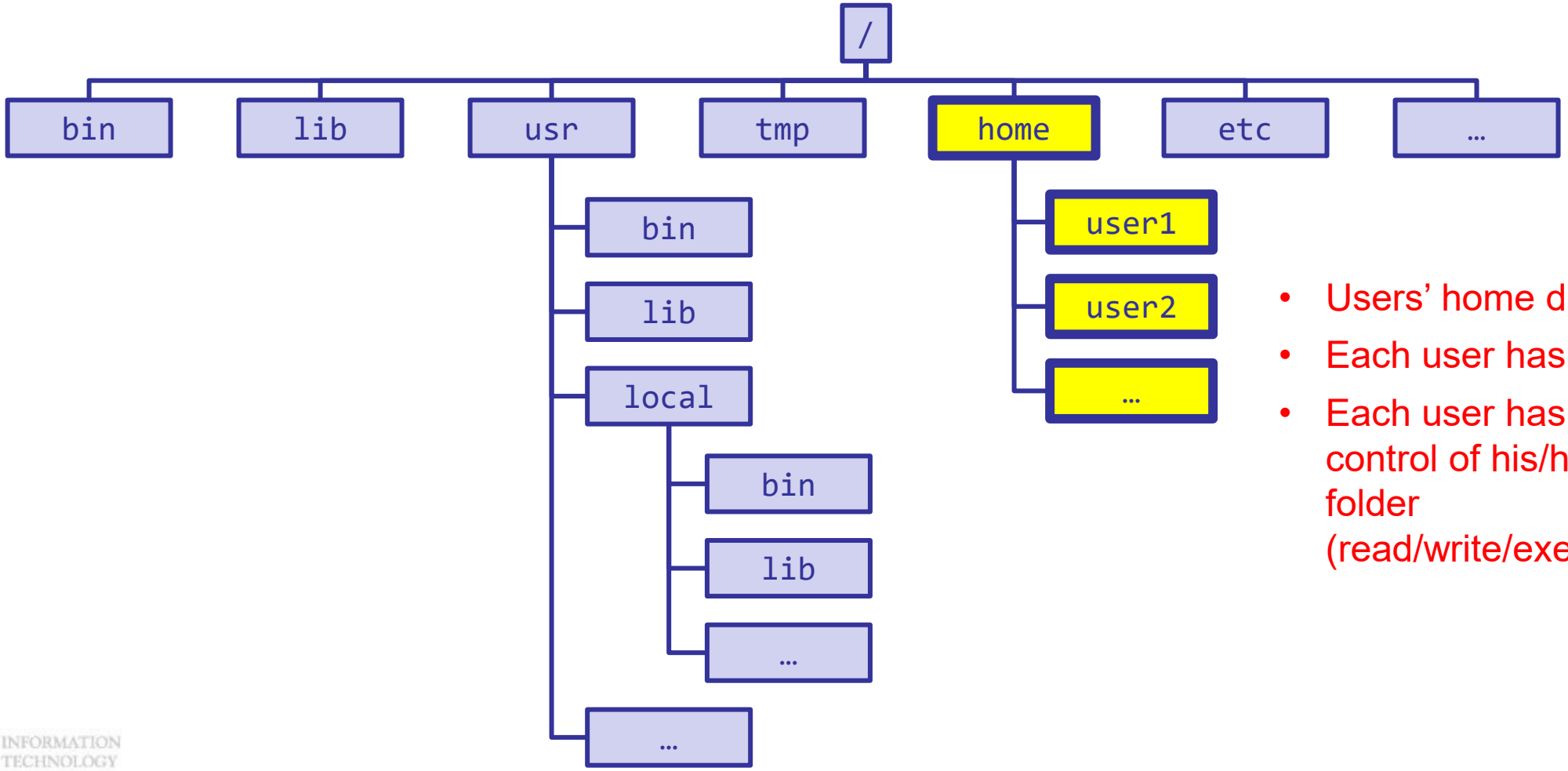
a) Filesystem Hierarchy Standard (FHS) : An inverted tree



- Temporary files

1) File System

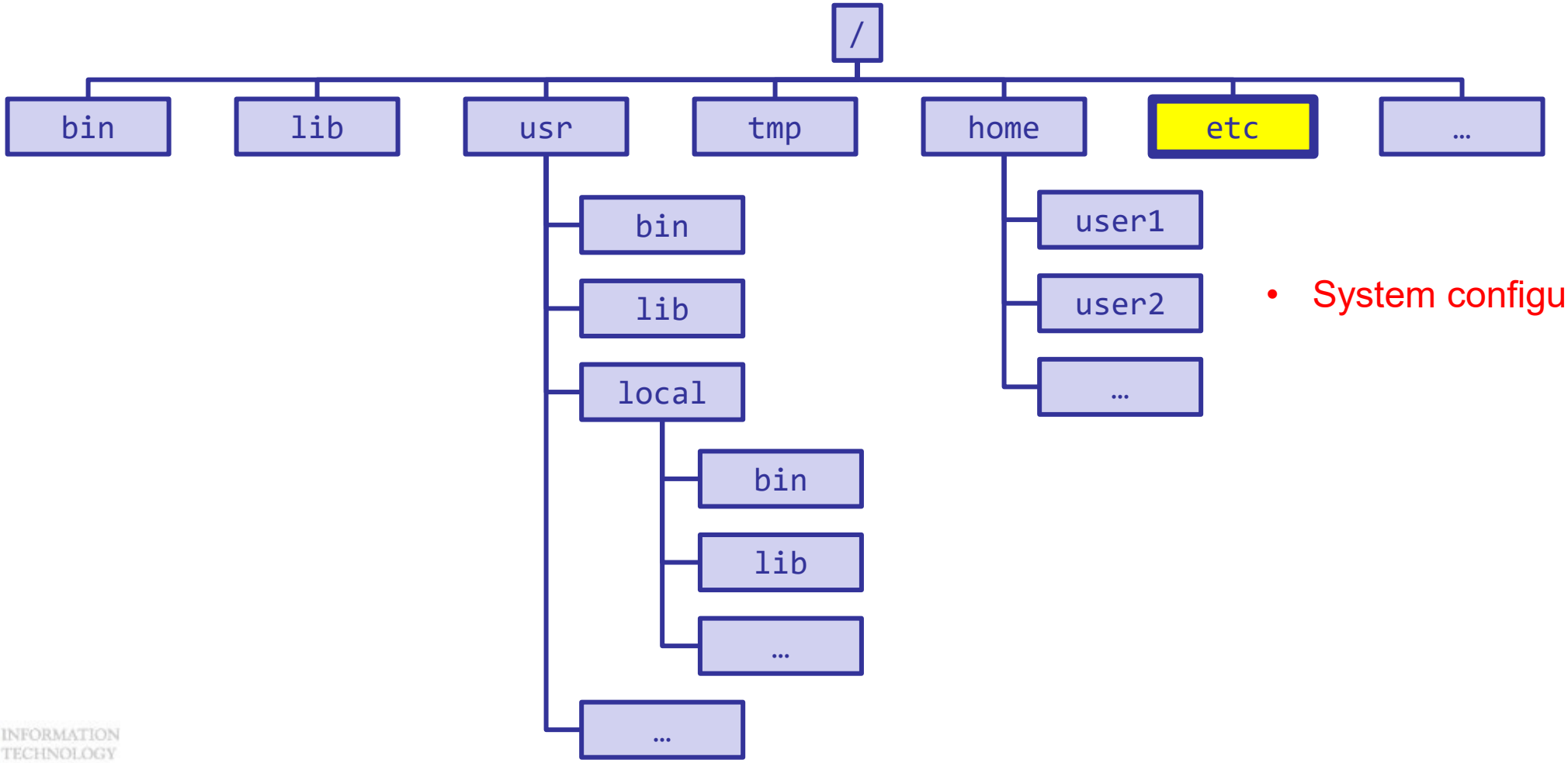
a) Filesystem Hierarchy Standard (FHS) : An inverted tree



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

1) File System

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
.	Current path	`/home/jasonli3/IntroToLinux/myfolder`
..	Parent path	`/home/jasonli3/IntroToLinux`
~	User's home directory	`/home/jasonli3/`

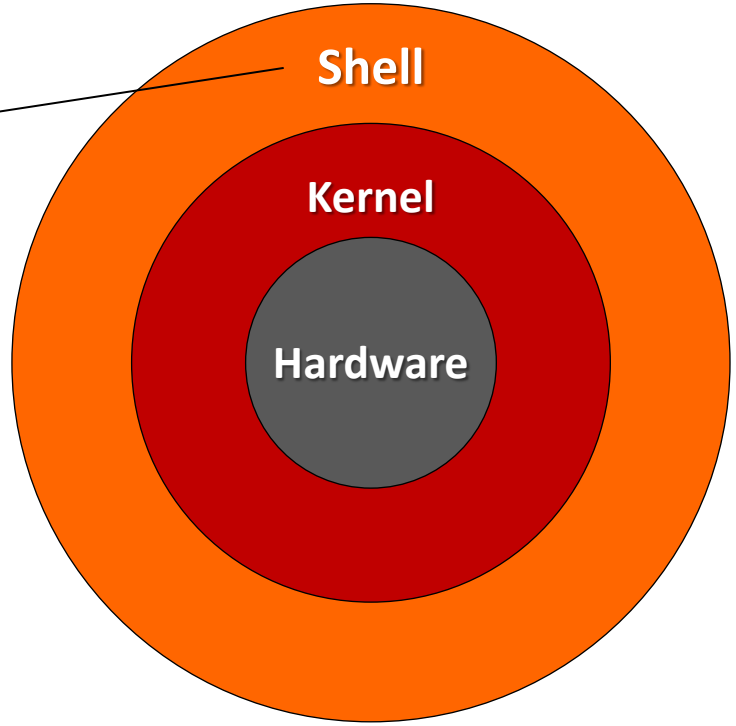
Outlines

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) Meet You Terminal

- **Before meeting your terminal**

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



2) Meet You Terminal

- **Before meeting your terminal**
  - Different Shell, different terminal
    - **sh** (Original Bourne Shell)
    - **bash** (Bourne Again Shell)
    - **csh** (C Shell)
    - **tcsh** (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

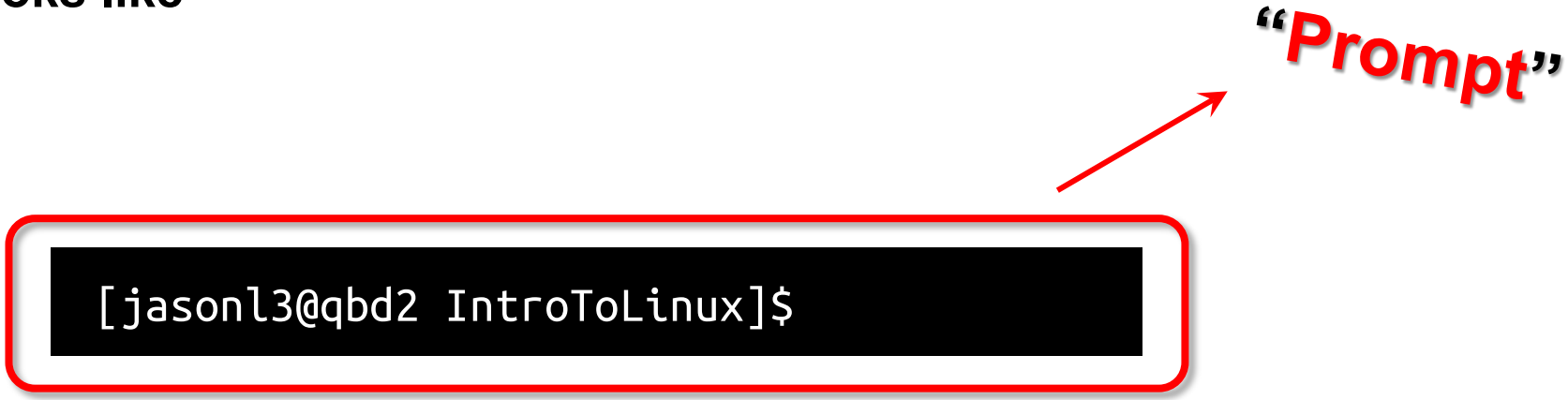
2) Meet You Terminal

- **Before meeting your terminal**
  - Different Shell, different terminal
    - **sh** (Original Bourne Shell)
    - **bash** (Bourne Again Shell)
    - **csh** (C Shell)
    - **tcsh** (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

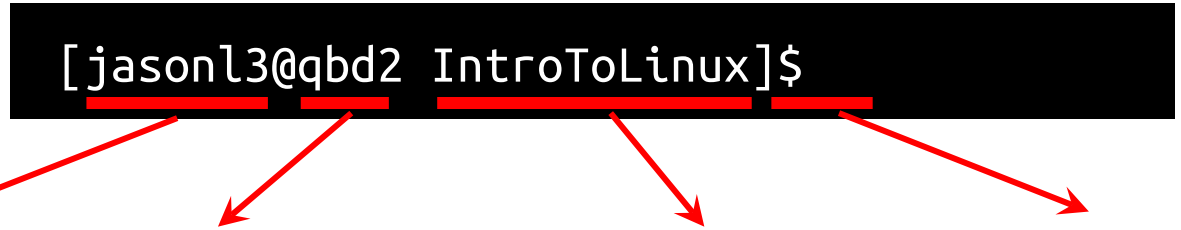
2) Meet You Terminal

- **What Terminal looks like**



2) Meet You Terminal

- **What Terminal looks like**



Username Hostname Current directory name "$" followed by user input

(Could be different for other Shells)

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

Outlines

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) Basic Commands

- **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, …

3) Basic Commands

- **Common “universal” options**

Option	Discription
`-h` `--help`	Help information
`-v` `--version`	Software version

3) Basic Commands

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

3) Basic Commands

a) Navigation

Command	Option	Augument	Description
cd (“**c**hange **d**irectory”)		`[DIR]`	Change current working directory to given destination (absolute or relative)
			Change to user’s home directory
pwd (“**p**rint **w**orking **d**irectory”)			Print current directory in standard output.

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)

b) File operations

Command	Option	Augument	Description
cp ("copy")		**SOURCE DEST**	Copy source file(s) to destination. • If **DEST** is an existing directory: Copy **SOURCE** under it • If **SOURCE** is single file: Can use **DEST** to rename file after copying • If **SOURCE** has multiple files: **DEST** must be an existing directory to move under
	-r	SOURCE DEST	Copy directory recursively
	-i	SOURCE DEST	Prompt before overwritten

b) File operations

Command	Option	Augument	Description
mv ("**m**ove")		**SOURCE DEST**	Move source file(s) to destination. • If **DEST** is an existing directory: Move **SOURCE** under it • If **SOURCE** is single file: Can use **DEST** to rename file after copying (**This is how you rename file in Linux!**) • If **SOURCE** has multiple files: **DEST** must be an existing directory to move under
	-i	SOURCE DEST	Prompt before overwritten

b) File operations

Command	Option	Augument	Description
`rm` ("**rem**ove")		**`FILE`**	Remove file(s)
	`-r`	`FILE`	Copy directory recursively
	`-i`	`FILE`	Prompt before removal
	`-f`	`FILE`	Force removal (ignore prompt). **DANGER!**

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

b) File operations

Command	Option	Augument	Description
mkdir (“**mak**e **dir**ectory”)		**DIR**	Create a directory under an **existing** parent directory
	-p	DIR	No error. Attempt to create parent directory(s) if needed

c) Content display

Command	Option	Augument	Description
`cat` (“Con**cat**enate”)		**`[FILE]`**	Print file content to standard output
`head` / **`tail`**		**`[FILE]`**	Print first / last **10 lines** of a file
	`-n[N]`	`[FILE]`	Print first / last **`[N]` lines** of a file
	`-c[N]`	`[FILE]`	Print first / last **`[N]` bytes** of a file

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

c) Content display

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

3) Basic Commands

- **Cheat sheet**

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

Outlines

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**

4) Terminal Like a Pro

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

4) Terminal Like a Pro

b) Navigate through past commands

- Imaging I am debugging a very lengthy command…

```
torchrun --nproc_per_node=8 --nnodes=4 --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 \
    --fsdp_policy "transformer_layer" --sharding_strategy "full_shard"
```

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	**ls test*** - List any file start with “test”
?	Replace any single character	**ls test.??** - List any file start with “test”, with 2-character extension
[ ]	Replace a single of a specific range	**ls test.[0-9]** - List any file start with “test”, with a single digit as extension

d) Shortcuts

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

Summary

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

Break

- 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

Outlines

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

Outlines

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) Understand Permissions

a) Why permission is important?

Windows:

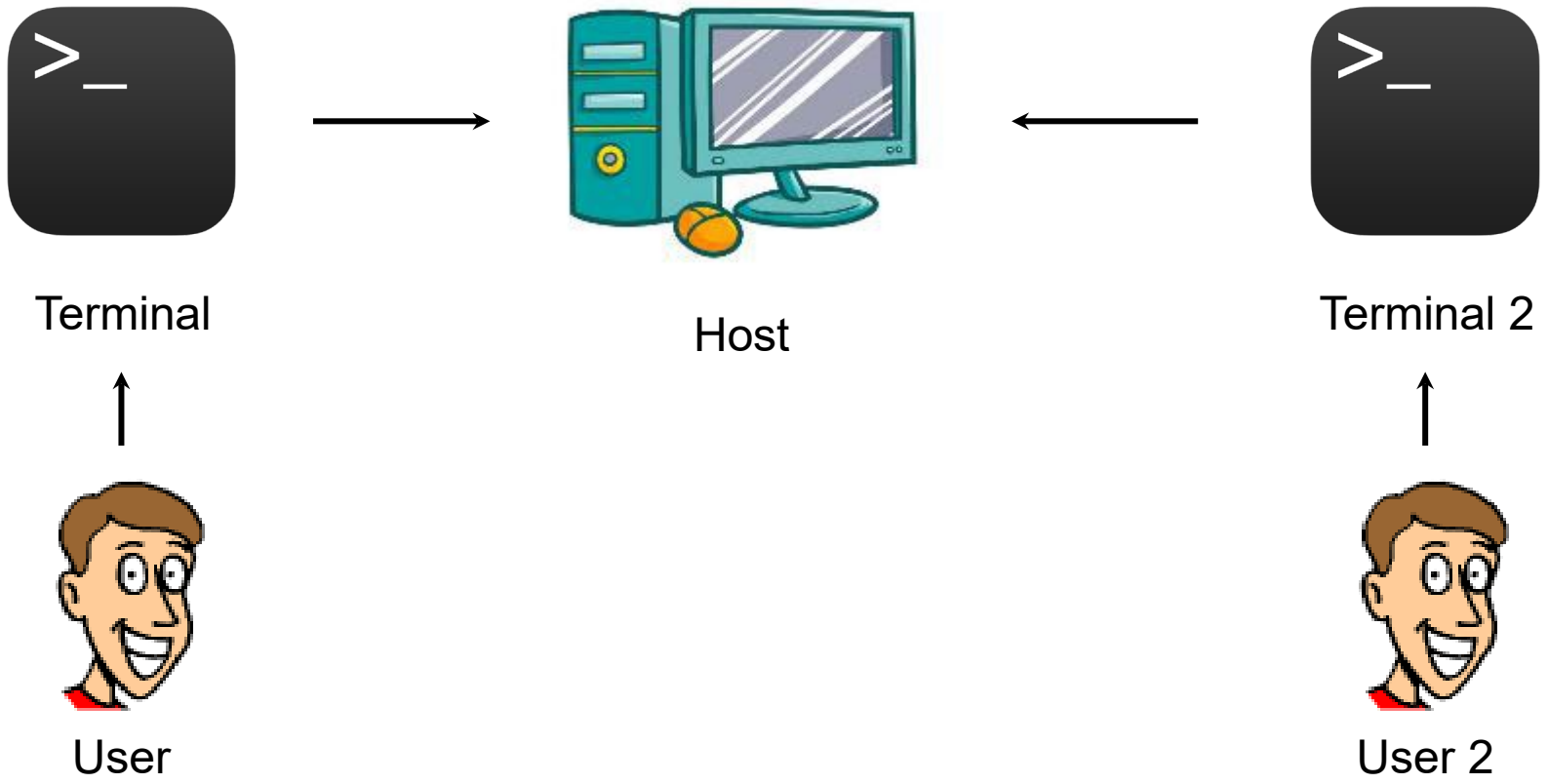
User

Machine

1) Understand Permissions

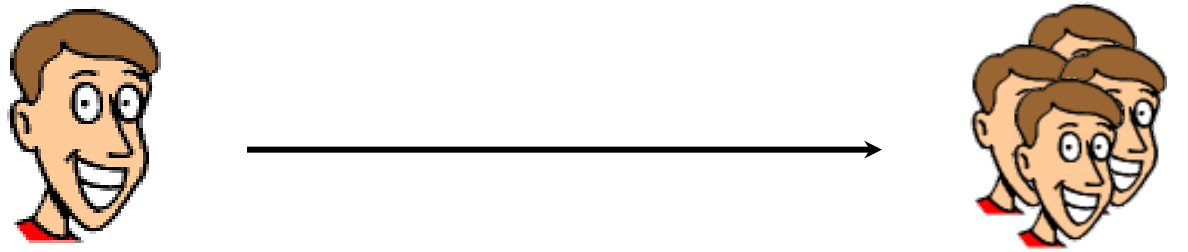
a) Why permission is important?

Linux:



b) Linux permission system

i. User groups



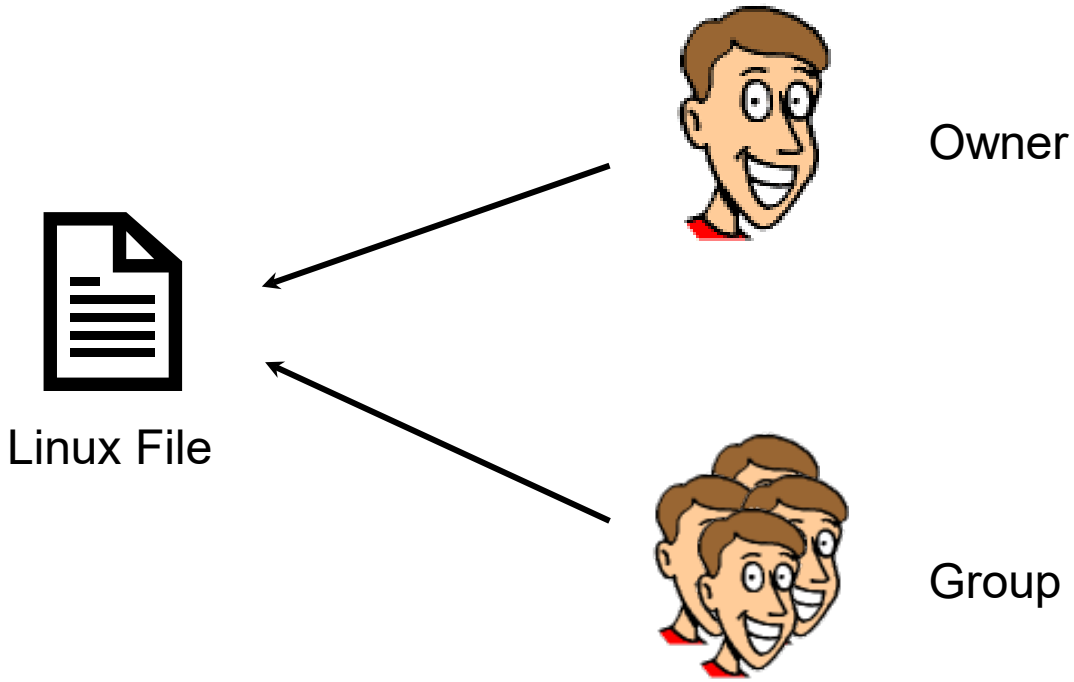
User

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----- 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
```

Group permissions

1) Understand Permissions

b) Linux permission system

v. View permissions: `ls -l`

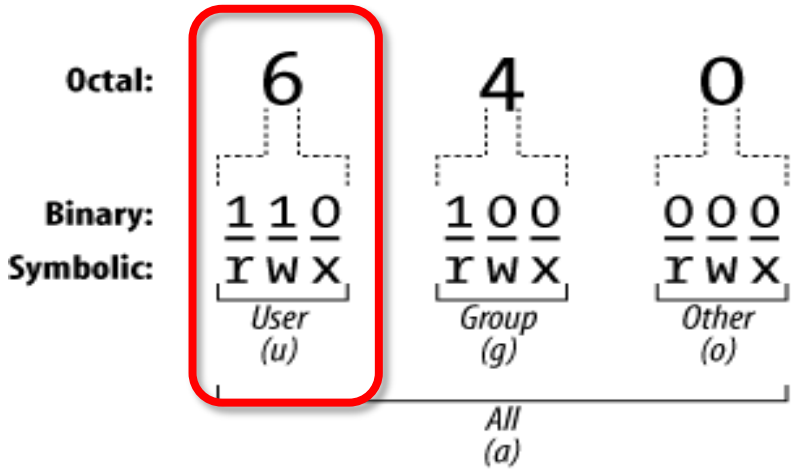
```
[jasonli3@qbd4 ollamaondemand]$ ls -l
total 140
-rw-r-----  1 jasonli3 sa_jasonli3  2432 Sep 10 11:30 arg.py
-rw-r-----  1 jasonli3 sa_jasonli3  2488 Jul 22  2025 chatsessions.py
drwxr-x---  2 jasonli3 sa_jasonli3  4096 Sep  1 13:30 container
-rw-r-----  1 jasonli3 sa_jasonli3  6801 Oct 29 17:21 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

b) Linux permission system

v. View permissions: `ls -l`

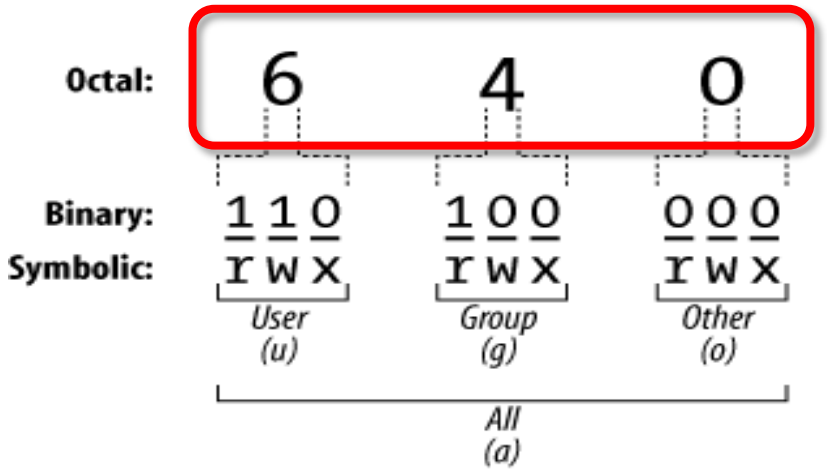
Octal notation:



b) Linux permission system

v. View permissions: `ls -l`

Octal notation:



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**

Outlines

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** ("**ch**ange **mod**e")

a) `chmod [options]` [u g o a] [+ - =] [r w x] `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** ("**ch**ange **mod**e")

a)

```
chmod [options]  [ u g o a ]  [ + - = ]  [ r w x ]  FILE(S)
```

-R (Recursively change in a directory)

2) Change Permissions

- **chmod** ("**ch**ange **mod**e")

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

E.g.,

- `chmod 755 FILE` (set mode to 755)

Summary

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`

Outlines

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

Useful Tools

- **This is HPC training, after all…**

  1) Tools You Can Use on HPC

  2) Tools You Cannot Use on HPC

Outlines

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

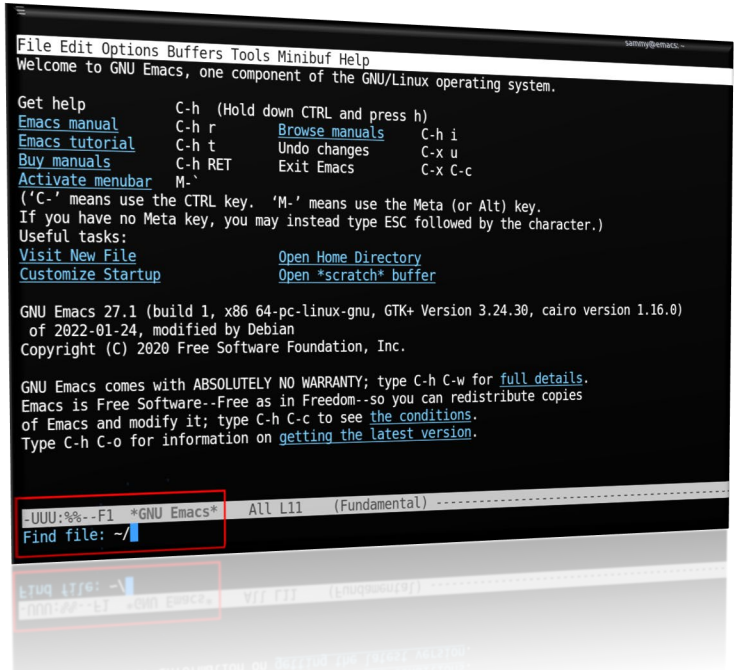
Command	Option	Augument	Description
du (“**d**isk **u**sage”)		**FILE(S)**	Calculate disk usage
	-h	FILE(S)	Show human-readable file size
	-s	FILE(S)	Only summarize total usage
top			Dynamically display system summary and processes
ln (“**l**in**k**”)	**-s**	**TARGET LINK**	Create symbolic link (symlink), where **LINK** is pointing to **TARGET**
wget (“**w**ww **get**”)		**URL**	Download file to current directory

1) Those You Can Use on HPC

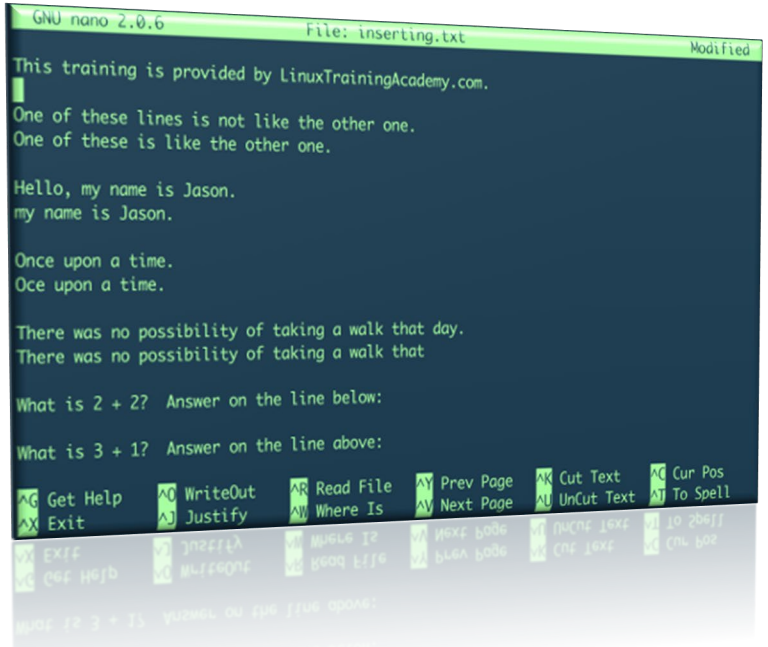
b) File editors

- 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**

Outlines

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** !

2) Those You Cannot Use on HPC

a) sudo

- Run command as **root** user ("**s**uper**u**ser **do**")
- E.g.,

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

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

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

- **Problem**: HPC users do **NOT** have root privilege!

2) Those You Cannot Use on HPC

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 iqtree
```

2) Those You Cannot Use on HPC

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

*Join us on 3/18/2026 for
Magic Tools to Install - Manage Software Part 1: Singularity Container!

Summary

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 us

- **Contact user services**
  - Email Help Ticket: sys-help@loni.org
  - Telephone Help Desk: +1 (225) 578-0900