

Computational Science and Scientific Computing Workshop

Elliot S. MENKAH, Ph.D
Daniella N. APEADU

National Institute for Mathematical Sciences, Ghana.
Kwame Nkrumah University of Science and Technology, Ghana.

October 6, 2025

Linux Command Line



Reading List and Reference Materials?

- 1 The Linux Command Line - A Complete Introduction by *William Shotts*
- 2 Linux Command Line and Shell Scripting Bible - *Richard Blum*

Linux Command Line



What's Scientific Computing and Why Linux?

- Using computers to analyze and solve problems
 - Eg. Automating daunting and repetitive task such as huge-size matrix vector operations.
- It allows the study of mathematical models of physical phenomena.
- It is used to find optimal system parameters.
- Experimentalists use computers to control experiments and to gather relevant data.

Linux Command Line



What's Scientific Computing and Why Linux?

- Using computers to analyze and solve problems
 - Eg. Automating daunting and repetitive task such as huge-size matrix vector operations.
- It allows the study of mathematical models of physical phenomena.
- It is used to find optimal system parameters.
- Experimentalists use computers to control experiments and to gather relevant data.

Linux Command Line



What's Scientific Computing and Why Linux?

- Using computers to analyze and solve problems
 - Eg. Automating daunting and repetitive task such as huge-size matrix vector operations.
- It allows the study of mathematical models of physical phenomena.
- **It is used to find optimal system parameters.**
- Experimentalists use computers to control experiments and to gather relevant data.

Linux Command Line



What's Scientific Computing and Why Linux?

- Using computers to analyze and solve problems
 - Eg. Automating daunting and repetitive task such as huge-size matrix vector operations.
- It allows the study of mathematical models of physical phenomena.
- It is used to find optimal system parameters.
- Experimentalists use computers to control experiments and to gather relevant data.

Linux Command Line - Outline



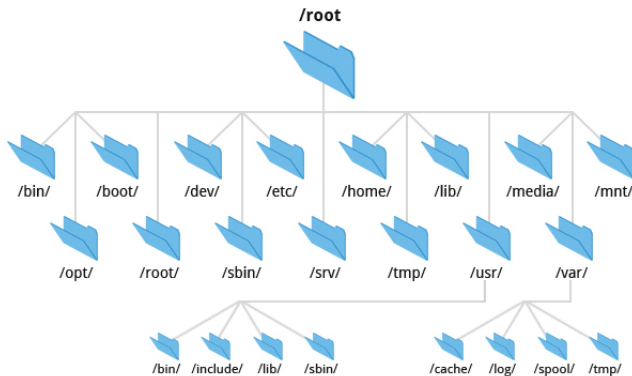
- File system
 - Linux File system
- Basic Operations
 - Basic Commands
 - File operations
 - User Environment
 - Access Control
 - Process Management
 - Network Management
- Text Editor
 - Vim
- Shell Tools & Programs
 - Shell Program
- Shell Programming
 - Bash Scripting
 - Variables
 - Statements
 - Conditionals
 - Control Sequence / Loops
 - Functions
- Regular Expressions
 - Regular Expression

Linux Command Line - File system



- Filesystem Types: ext2, ext3, ext4, reiserfs, vfat, xfs, nfs
- Devices: Block devices, Loop devices
- Block Devices:
- inodes
- FHS: Filesystem Hierarchy Standard
- NFS: Network File System

Linux Command Line - Linux Directory Structure



Linux Command Line - Basic Operations



- Basic Operations
 - Basic Commands
 - File operations
 - User Environment
 - Access Control
 - Process Management
 - Network Management

Linux Command Line - Basic Commands



ls: list files

pwd: present working directory

cd: change directory

cat: list file content

alias: remap a command

date: check or set date/time

uname: OS info. version and architecture

Linux Command Line - Basic Commands



- **pwd**: present working directory
- **ls**: list files/directories

ls: list files

```
1 ~ $ ls
```

pwd: present working directory

```
1 ~ $ pwd
```

cd: change directory
cd directory-name

```
1 ~ $ cd Desktop
```

Navigate one step(directory) back **cd ..**

```
1 ~ $ cd ..
```

- **cd**: change directory

ls

```
1 Desktop Pictures
2 Documents Downloads
```

pwd

```
1 /home/elliott
```

pwd

```
1 /home/elliott/Desktop
```

pwd

```
1 /home/elliott
```

Linux Command Line - Basic Commands



touch: create new files

mkdir: create new directory

cp: cp files & directories

mv: relocate/move file & directories

rm: delete files & directories

Linux Command Line - Basic Commands ...

- **touch**: create a new file
- **cp**: copy files & directories
- **mkdir**: create a new directory

Create a new file using **touch 'filename'**

Eg.

```
1 ~ $ touch file1
2 ~ $ touch file2
3
```

Create a new directory using **mkdir 'directoryname'**

Eg. To create two directories called **dir1** and **dir2**

```
1 ~ $ mkdir dir1
2 ~ $ mkdir dir2 dir4 dir5
3
```

Linux Command Line - Basic Commands ...

Copy a **file** from a particular location to another using

cp 'file-to-be-copied' 'new-destination'

Eg. Copy *file1* to directory called **dir1**

```
1 ~ $ cp file1 dir1
```

Copy a **directory** from a particular location to another. Copying is done **recursively**, [-r]

cp -r 'directory-to-be-copied' 'new-destination'

Eg. Copy *file1* to directory called **dir1**

```
1 ~ $ cp -r dir4 dir5
```

More info on commands: <https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners/>

Linux Command Line - Basic Commands ...

- **mv**: move/relocate/rename a file/directory

Move/Relocate a file/directory to a new location using

mv 'file-to-be-moved' 'new-destination'

Eg. To place *file2* into **dir2**

```
1 ~ $ mv file2 dir2
```

'file-to-be-moved' and 'new-destination' are actually **paths**.

Linux allows relative paths

Technically, the syntax below shows how it absolute paths works with mv

```
1 ~ $ mv /home/elliott/file2 /home/elliott/dir2
```

Rename a **file** using **mv 'file-to-be-renamed' 'new-name'**

Eg. To rename *file1* into **file3**

```
1 ~ $ mv file1 file3
```

Rename a **directory** using **mv 'directory-to-be-renamed' 'new-name'**

Eg. To rename *dir1* into **dir3**

```
1 ~ $ mv dir1 dir3
```


Linux Command Line - Process Management



ps:: list processes

kill: kill processes

top: monitor processes

fuser: find process owner

Linux Command Line - User Environment



env: command to view user environment

export: command to add to user environment

bashrc: file to store user environment settings

profile: file for global user environment settings. /etc/profile

tilde: reference to current user

Linux Command Line - Basic Commands ...



End of Basic Commands, thank you ...