



Introduction to Linux

(Part 1/2)

Misha Ahmadian
High Performance Computing Center

Summer 2023

About this Training Workshop?



TEXAS TECH UNIVERSITY
Information Technology Division™

A hands-on training that covers essential Linux topics for:

- HPC users who need a basic understanding of Linux commands for their daily basis task.
- Non-HPC users who are willing to join HPC and leverage the TTU supercomputing facilities.
- Non-HPC users who just want to learn Linux.

About this Training Workshop?



TEXAS TECH UNIVERSITY
Information Technology Division™

Course Schedule:

- The Introduction to Linux Training is offered twice per semester.
- Each session will be held for 4 hours per day (10 am – 12 pm) and (1 pm – 3 pm) with a 1-hour lunch break.

Requirements:

- Attendees are preferred to have a ready-to-use HPCC account:
 - HPCC Login nodes
 - Linux/Mac Laptops
 - Windows machines with WSL or Linux on a Virtual Machine
 - Online Linux Virtual Consoles (*It may not offer full functionality*) – e.g. JSLinux



How to make the best out of this Training?

- Take your notes during the sessions, but not too many!
- Follow the instructor and try the new Linux commands on your Linux/Mac console.
- A few exercises will be provided during each session, which will help you to practice and learn.



Part 1: Introduction to basic topics in Linux

- ❖ Getting Started
- ❖ About Linux
- ❖ Working with Files and Directories
- ❖ Linux Essential Commands (Part 1)



Logging into HPCC Resources (HPCC Users):

- Linux/Mac Users:
 - SSH (Secure Shell): Freely available on Linux/Unix/macOS and used via the Terminal.
 - ssh **eraider**@login.hpcc.ttu.edu
 - ssh **eraider**@quanah.hpcc.ttu.edu
- Windows Users:
 - MobaXterm (Recommended):
 - <https://mobaxterm.mobatek.net>
 - Putty:
 - <https://www.putty.org>
- User Guides, Other Alternatives, and Details:
 - See the “Connecting to HPCC Systems” link at <http://www.hpcc.ttu.edu/userguides/>



About Linux...



GNU Project:

- Richard Stallman started the GNU project in 1983
 - After AT&T imposed commercial licensing on Unix Operating System
 - Reprogrammed some of the Unix Tools. running on a small kernel (TRIX)
 - Needed more advanced and reliable OS Kernel for his programs

Linux:

- Linus Torvalds created the Linux kernel in the early 1990s
 - Linus released the first kernel under the GPL (General Public License)
 - *Requires that the source code remains freely available to everyone*



Linux has various distributions (Distros):

- Why many distributions?

- Server vs. Desktop
- Commercial user support
- Special hardware support

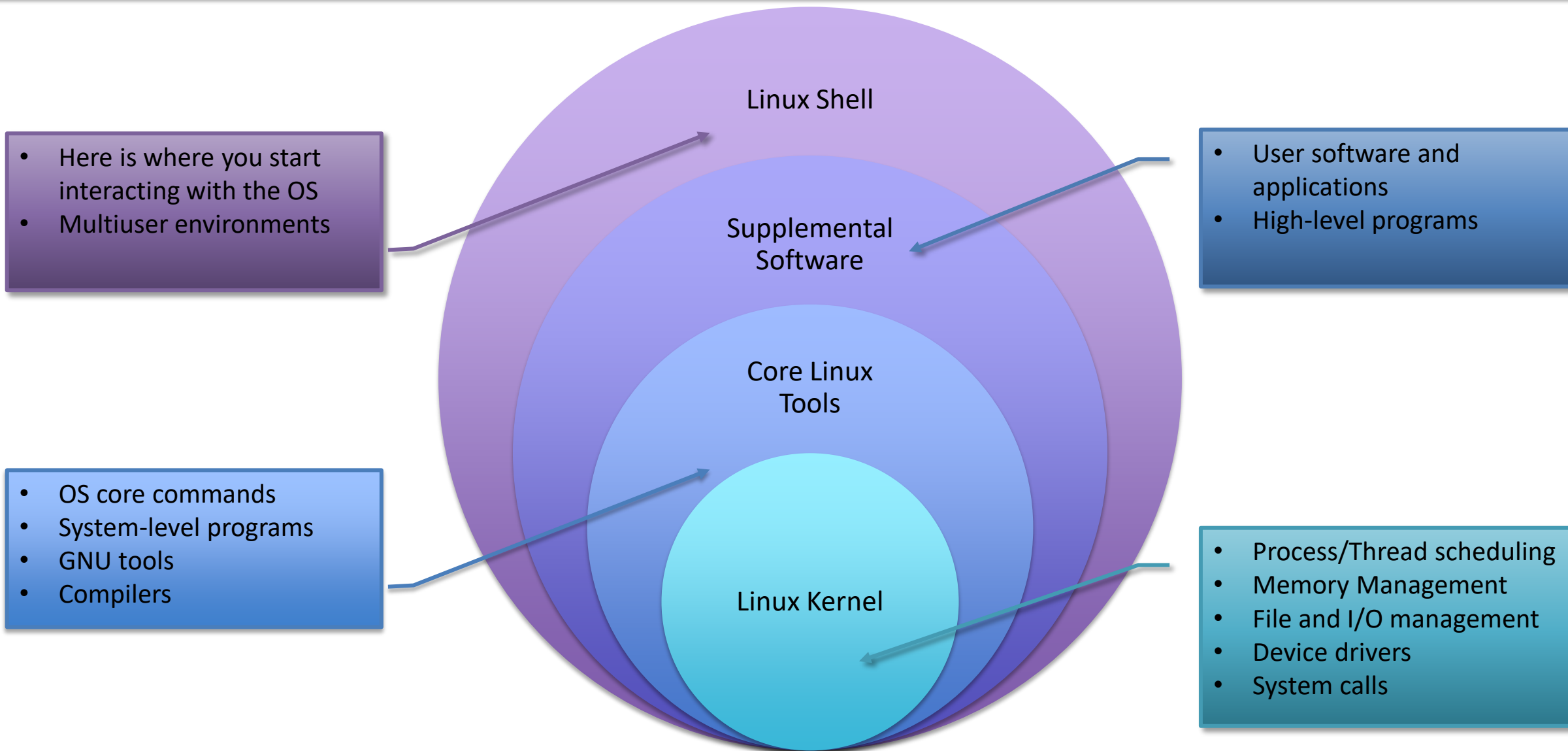


- Debian/RedHat/SUSE/...

- The major difference is the software package management on these distros.



Structure of Linux OS





What is the shell?

- The shell is a program that takes commands from user's keyboard and passes them to the operating system to execute.
 - Many shell programs are available for Linux:
 - bash, sh, csh, tcsh, ksh, zsh, ...
 - For this tutorial, we will use `bash`, but other shells are conceptually similar

```
quanah:$ command
```

What is Terminal?

- Terminal is a program that opens in a window and lets users interact with the shell.



Working with Files and Directories

Understanding Linux Directory Structure



TEXAS TECH UNIVERSITY

Information Technology Division™





Where am I?

- **pwd** command to print working directory

```
quanah:$ pwd  
/home/username
```

- **ls** command to list contents of the current directory

```
quanah:$ ls  
test1.txt  
Mydir1  
mydir2
```



Make/Remove/Go into a directory?

- **mkdir** command to make a new directory

```
quanah:$ mkdir my_new_dir  
quanah:$ ls  
my_new_dir
```

- **cd** command to change into a directory

```
quanah:$ cd my_new_dir
```

- **rmdir** command to remove an empty directory

```
quanah:$ rmdir my_new_dir
```



More about **ls**. (Commands with options)

- **ls -a** (**--all**) list all files including hidden files

```
quanah:$ ls -a
```

- **ls -l** show file details

```
quanah:$ ls -l
```

- Combine multiple options for a command

```
quanah:$ ls -a -l
```

```
quanah:$ ls -al
```




Special Directory Names (Characters)

TEXAS TECH UNIVERSITY
Information Technology Division™

- Return to home directory (~)

```
quanah:$ cd  
quanah:$ cd ~  
quanah:$ cd ~/
```

- Current directory (.)

```
quanah:$ ls .  
quanah:$ ls ./
```

- Parent directory (..)

```
quanah:$ cd ..  
quanah:$ cd ../  
quanah:$ ls ../..  
quanah:$ cd ../../mydir/
```



Absolute VS. Relative path

Absolute Path of a file/directory:

- Always starts with / and goes all the way to the file or directory
 - For instance:

```
quanah:$ cat /home/mahmadia/mydir/file1.txt
```

Relative Path of a file/directory:

- It defines from the current or parent directory
 - For instance:

```
quanah:$ cat ../mydir/file1.txt
```



File and Directory Names

Naming: What to use:

- Letters and numbers [A-Z a-z 0-9]:
 - File and directory names are case sensitive (e.g. 'M' vs 'm')
- (.) can be used anywhere in the file/directory name
 - Using (.) at the beginning of the file/directory name makes it hidden
 - Using (.) alone will be considered as current working directory

- A-Z
- A-z
- 0-9
- dot (.)
- dash (-)
- underscore (_)

What to Avoid:

- Avoid spaces in the file names
 - “my file name.txt” → “my_file_name.txt”
- Avoid these characters:
 - () " ' ? \$ * \ / :

- () parenthesis
- " double quotes
- ' single quote
- ? Question mark
- \$ dollar sign
- * asterisk
- \ back slash
- / slash
- : colon



Let's learn a few file operations before Exercise #1

- **touch** creates an empty file (Not the primary job!)

```
quanah:$ touch my_file
quanah:$ ls
my_file
```

- **echo** prints its arguments to the standard output
- **>** redirects the standard output to a file (Creates/Rewrite the file)
- **>>** redirects the standard output to a file (Append to the file)
 - **More details on output redirects in next few slides**

```
quanah:$ echo "hello world!" > hello.txt
quanah:$ echo "second line" >> hello.txt
```



A few Basic File Operations (Cont.)

Let's learn some file operations before Exercise #1

- **cat** shows the content of a file(s)

```
quanah:$ cat hello.txt
Hello world!
second line
```

- **wc** counts number of lines, words, and characters in a text file

```
quanah:$ wc hello.txt
 2  4 25 hello.txt
quanah:$ wc -l hello.txt
 2 hello.txt
quanah:$ wc -w hello.txt
 4 hello.txt
```

Exercise #1



1. Return to your home directory
2. Create a new directory and name it “exercise1”
3. Go into the “exercise1” directory
4. Create a new text file and name it “test1.txt”. Then add the following lines into the file:

```
Exercise #1:  
This is a test file  
here is the last line
```
5. Now create a new directory under the “exercise1” directory and name it “subdir1”.
6. Go into the “subdir1” and try to print the content of “test1.txt” on your terminal.
7. Return to the “exercise1” directory and count the number of lines in “test1.txt”.



Copy Files and Directories:

- **cp** command to make a copy of a file(s) or directory(s)
 - **-r**: Recursively copy the directories and subdirectories as well

```
quanah:$ ls
file1.txt
quanah:$ cp file1.txt file2.txt
quanah:$ ls
file1.txt  file2.txt
quanah:$ cp -r ../mydir ./
quanah:$ cp ../mydir2/* /home/bob/
quanah:$ cp /*.txt ~/text_dir/
```



Move/Rename Files and Directories:

- **mv** command to make a copy of the source file(s) and directory(s) into the destination and remove the source at the end.
 - Can be used for renaming a file/directory as well

```
quanah:$ ls
file1.txt
quanah:$ mv file1.txt file2.txt
quanah:$ ls
file2.txt
quanah:$ mv mydir/ ../temp/
quanah:$ mv ../docs/*.pdf ./
```




Deleting Files and Directories:

- **rm** command to remove file(s) and directory(s) permanently
 - Removing files and directories by this command may not be recovered easily (This is an exception for /home directory on HPCC clusters)
 - **-r**: Recursively copy the directories and subdirectories as well

```
quanah:$ ls
file1.txt  file2.txt
quanah:$ rm file1.txt
quanah:$ ls
file2.txt
quanah:$ rm ../mydir/*
quanah:$ rm -r ../mydir/
```



- So far, all the commands that we used sent their output to the screen. We can control this in Linux:
 - `<` redirects to the standard input
 - `>` redirects the standard output
 - `>>` appends the standard output
 - `1>` redirects the standard output (equals to `>`)
 - `2>` redirects the standard error
 - `&>` redirects the standard output and standard error
 - `|` (pipe) pass the output of one command to another

Exercise #2



1. Return to your home directory
2. Create a new directory and name it “exercise2”
3. Go to the “exercise2” directory
4. Copy the “test1.txt” file from the “exercise1” directory to the “exercise2” directory
5. Make a new directory under “exercise2” and name it “subdir2”
6. Rename the file “test1.txt” to “test2.txt”
7. List the contents of “exercise2” and redirect the output to “ls.out” file
8. Remove everything under the “exercise2” directory



File/Directory Ownership and Permissions:

- Every file/directory belongs to a specific user or a group of users
- Every user/group may have permissions to `read`, `write`, and/or `execute`

User	Group	Other
<code>r w x</code>	<code>r w x</code>	<code>r w x</code>

- If you set **write** permission for a directory, you can (create/delete) new entries
- If you set **read** permission for a directory, you can list (`ls`) the contents
- If you set **execute** permission for a directory, you can (`cd`) into the directory



File/Directory Ownership and Permission Examples:

- **chmod** command changes the `rwX` mode bits of a file or directory
 - **+/-**: adds or removes the mode bits
 - **u**: Sets the permissions for the owner of the file/directory
 - **g**: Sets the permissions for the group that the owner belongs to
 - **o**: Sets the permissions for the other users
 - **a**: Sets the same permissions for user/group/other

```
quanah:$ chmod +x script.sh
quanah:$ chmod g+rx my_program
quanah:$ chmod a-r my_docs
quanah:$ chmod 755 ./mydir
```



File/Directory Ownership and Permission:

- With **stat** or **ls -l** commands you can check the ownership and permissions of a file or directory
- **whoami** command Displays the username of the current user
- **groups** command prints the groups a user belongs to



Linux Essential Commands

(Part 1)



Check the manual page of a command:

- **man** command to search and open the manual page of a Linux command

```
quanah:$ man ls  
quanah:$ man cp
```

- **/word** search the man page for the text 'word'
- **n** search for the next found match
- **N** search for the previous found match
- **g** Go to the beginning of the manual page
- **G** Go to the end of the manual page
- **q** Exit (quit)



Other useful commands to find out more:

- Several built-in references are available within the operating system.
 - **info** more detailed tutorials and examples.
 - **apropos** Search the manual database for a “string”.

```
quanah:$ info cp  
quanah:$ apropos mkdir
```

- Many reference guides are available.
 - Search the web for “reference” – for example “bash reference”.
 - Specifically for bash, try finding the “bash advanced scripting guide” (but only when you are ready for the advanced stuff!
 - Guides are available for other shells and commands also.



Paging through long text files/outputs:

- **more** command for paging through text one screenful at a time
 - More only allows to move the text forward

```
quanah:$ more ./README  
quanah:$ cat ./README | more
```

- **Less** command is an improved version of **more** command which allows backward movement in the file as well as forward movement

```
quanah:$ less ./README  
quanah:$ cat ./README | less
```



Display a partial content of a file/output:

- **head** command to see first lines of a file (10 lined by default)
 - **-n <NUM>**: First NUM number of a file/output

```
quanah:$ head -n 15 ./README
quanah:$ cat ./README | head -3
```

- **tail** command to see last lines of a file (10 lined by default)
 - **-n <NUM>**: Last NUM number of a file/output

```
quanah:$ tail -n 20 ./README
quanah:$ cat ./README | tail -5
```



Searching for pattern(s) in files/outputs:

- **grep** command to print lines in a file matching a pattern
 - **-i**: Ignore case distinctions in both the PATTERN and the input files
 - **-v**: Invert the sense of matching, to select non-matching lines

```
quanah:$ grep "test" ../exercise1/test1.txt
This is a test file
quanah:$ grep "is" ../exercise1/test1.txt
Exercise:
This is a test file
here is the last line
```



History of your commands:

- **history** command to see the commands you have typed so far.
 - Your command are saved to the **.bash_history** file under your home directory.

```
quanah:$ history | less
quanah:$ history | tail -15
quanah:$ history | grep "cp"
```

Exercise #3



1. Search in your history and try to list your last 10 “ls” commands that you have used
2. Check the manual of the “man” command and see what type of manuals are available on Linux

Coming in Part 2 of this Training ...



TEXAS TECH UNIVERSITY
Information Technology Division™

Part 2: Introduction to more advanced topics in Linux

- ❖ Linux Essential Commands (Part 2)
- ❖ Text Editors in Linux
- ❖ Linux Environment Variables
- ❖ Basic Bash Scripting in Linux



Lunch Break

Let's get back at 1:00 pm



TEXAS TECH UNIVERSITY
Information Technology Division™