Introduction to Linux

(Part 2/2)

Misha Ahmadian
High Performance Computing Center

Summer 2023

Course Agenda

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

Review

Command	Description
`pwd`	Prints Current Working Directory
`ls`	Lists the contents of a directory
`cd`	Change the current path to the destination directory
`mkdir`	Makes a new directory
`rmdir`	Removes an empty directory
`cp`	Copy file or directory
`mv`	Move/Rename file or directory
`rm`	Delete file or directory
`cat`	Concatenates and prints the contents a file

Review

Command	Description
`echo`	Write arguments to the standard output
`wc`	word, line, character, and byte count
`man`	Search and open the manual page of a Linux command
`more`	Paging through text one screenful at a time
`less`	Improved version of more allows backward/forward movement
`head`	Display first lines of a file
`tail`	Display last lines of a file
`grep`	Print lines in a file matching a pattern
`history`	See the commands you have typed so far

Linux Essential Commands

(Part 2)

Define the file type:

- Unlike Windows, files extensions rarely define the type of a file in Linux.
  - For example: "`file.txt`" many not be a TEXT file.
- **`file`** command displays the file type along with a brief summary of the file contents.

```
quanah:$ file myfile
myfile: ASCII text
quanah:$ file /home/mahmadia/program.py
/home/mahmadia/program.py: Python script, ASCII text executable
quanah:$ file test.tar.gz
test.tar.gz : gzip compressed data, from Unix, last modified: Wed
Sep  4 14:04:10 2019
```

CRLF Line Terminator:

- Windows text editors such as notepad add a Carriage Return Line Feed (CRLF) character at the end of each line of the text which causes problems with many Linux applications.
- **`dos2unix`** command fixes the CRLF issue in text files from Windows.

```
quanah:$ file windows.txt
windows.txt: ASCII text, with CRLF line terminators
quanah:$ dos2unix windows.txt
dos2unix: converting file windows.txt to Unix format ...
quanah:$ file windows.txt
windows.txt : ASCII text
```

Compression and File Packaging:

- **`zip`** command packages and compresses files and directories
  - **`zip`** `[OPTIONS] zip_file file_dir1 file_dir2 …`
  - **`-r`** : Add the directories and subdirectories contents into the zip file

```
quanah:$ ls
mydir test1.txt
quanah:$ zip -r archive.zip ./*
adding: ./test1.txt (deflated 62%)
adding: ./mydir/ (deflated 0%)
quanah:$ ls
archive.zip mydir test1.txt
```

Compression and File Packaging:

- **`unzip`** command lists and extracts the contents of a zipped file
  - **`-l`** : Lists the contents of a zipped file

```
quanah:$ ls
archive.zip
quanah:$ unzip archive.zip
quanah:$ ls
archive.zip mydir test1.txt
```

Compression and File Packaging:

- Other compression commands available in Linux

Command	Description	Decompression	File Ext.
zip	Packages and compresses files and directories	`unzip`	`.zip`
gzip	A GNU tool for compressing or expanding files/directories	`gunzip` `gzip -d`	`.gz`
bzip2	Compresses files using the Burrows-Wheeler block sorting text compression algorithm.	`bunzip2` `bzip2 -d`	`.bz`, `.bz2`
xz	Similar to `gzip` and `bzip2`	`unxz`	`.xz`

Linux Essential Commands 2

Archiving:

- **`tar`** command saves many files and directories into a single “archive” file
  - **`tar`** `OPTIONS dest_file src1 src2 …`

  - **`-f`** define the archive file path/name
  - **`-c`** Create a new archive
  - **`-a`** Append to the existing archive file
  - **`-x`** Extract the contents of an archive file
  - **`-z`** Compress archive file with gzip
  - **`-j`** Compress archive file with bzip2
  - **`-v`** verbosely list files processed

Linux Essential Commands 2

Archiving:

- **`tar`** command examples:

```
quanah:$ ls
mydir test1.txt
quanah:$ tar -cf myarchive.tar ./*
quanah:$ ls
myarchive.tar mydir test1.txt
quanah:$ tar -xvf myarchive.tar
test1.txt
mydir/
quanah:$ tar -czf docs.tar.gz /home/mahmadia/docs
quanah:$ ls
Docs.tar.gz myarchive.tar mydir test1.txt
quanah:$ tar -xzf docs.tar.gz
```

Linux Essential Commands 2

Download files from internet:

- **`wget`** command downloads files from internet
  - **`-O`** : (capital O) defined the name of the destination file on your system

```
quanah:$ wget "https://repo.anaconda.com/miniconda/Miniconda3-
latest-Linux-x86_64.sh"  -O miniconda3.sh
```

Exercise #4

1. Go to your home directory
2. Create a new directory and name it “exercise4”
3. Go to the “exercise4” directory
4. Choose a small directory from your home directory
5. Try to archive and compress the directory by tar and save it under the “exercise4” directory.
6. Now try to decompress the tar file that you just created
7. Check the type of the file
8. Now try to untar the file

Access Control List (ACL):

- Second level of discretionary permissions that override the standard **ugo/rwx**
- better granularity in setting access to a file or a directory
- Recommended on HPCC cluster environments
- **`getfacl`** shows the file/directory Access Control List
  - `getfacl <File | Directory>`
- **`setfacl`** modify/remove the ACL permissions
  - `setfacl -m [u|g]:<username>:rwx  <File|Directory>`
  - `setfacl -x [u|g]:<username> <File|Directory>`

Linux Essential Commands 2

TEXAS TECH UNIVERSITY
Information Technology Division™

```
quanah:$ getfacl test.txt
# file: test.txt
# owner: mahmadia
# group: CS
user::rw-
group::r--
other::r--
quanah:$ setfacl -m u:user1:rw test.txt
quanah:$ getfacl test.txt
# file: test.txt
# owner: mahmadia
# group: CS
user::rw-
user:user1:rw-
group::r--
mask::rw-
other::r--
```

Text Editors in Linux

How to edit text files in Linux?

- There are many text editors available on Linux
  - **`nano`** is a small, simple and friendly editor
  - **`vi/vim`** is a powerful text editor which can be used to edit all kinds of text
  - **`emacs`** is part of the GNU project written by Richard Stallman

- In this training course we will cover **`nano`** and **`vi/vim`**
- Let's look into **`nano`** *(Demo)*

How does `vi/vim` work?

- **`vi/vim`** is a very popular text editor among programmers and system administrators
- It supports many programming and scripting languages
- Suitable for more advanced file editing
- **`vi/vim`** has two modes:
  1. **Text mode**: which can be enabled by typing **`i`** (insert) or **`a`** (append)
  2. **Command mode**: which will be enabled by pressing the **Esc** key on keyboard.

Text Editors in Linux

Some useful `vi/vim` commands:

command	description	command	description
`!`	Forces the action	`i`	insert
`:q`	quit	`a`	append
`:q!`	Force quit	`x`	Delete a character
`:w`	write	`y[count]y`	Yank (copy) [count] lines
`:wq`	Write and quit	`d[count]d`	Cut (Delete) [count] lines
`:x`	Write and quit	`p`	Paste after the current line

- Let's look into **`vim`** *(Demo)*

Exercise #5

1. Go to your home directory
2. Create a new directory and name it “exercise5”
3. Go to the “exercise5” directory
4. Open a new text file with the editor of your choice and type the lines below:

```
Exercise #5
This is a Linux text editor exercise
We finally made it!
```

5. Save the file and exit
6. Try to display the contents of the file on your screen
7. Change the permission of the file as below:
   - Set `read/write` permission for the owner
   - Set `read-only` permission to your group
   - Give a `read-only` permission to user `mahmadia` and then remove it.

Environment Variables in Linux

Linux Environment Variables

What is the environment variable?

- Environment Variables stores any user-defined or system-defined information that can be accessed within the shell.
- Environment Variables are useful for passing data to programs or being used in shell scripts.
- Defining a variable is very simple (do not put spaces around = symbol)

```
quanah:$ VAR_NAME=“This is a variable”
```

- When referencing a variable place a ($) before the variable name

```
quanah:$ echo $VAR_NAME
This is a variable
```

Common Linux Environment Variables

HOME	Pathname of the current user's home directory
PATH	Colon separated list of directories where commands can be found
SHELL	Name of the current Shell program
PWD	Print current working directory
USER	Print current username
TERM	The type of the terminal
HOSTNAME	Displays computer's hostname

PATH Environment Variable

- Shell uses the PATH environment variable to locate commands
- The PATH variable is colon (:) separated, and can be displayed with echo

```
quanah:$ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
```

- You can add a directory into the PATH variable of your own environment

```
quanah:$ export PATH="/home/username/bin:$PATH"
/home/username/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin
```

- **export** command promotes a shell variable to an environment variable

Set Environment Variables at login time

- **`~/.bashrc`** : Commands for environment variables that you want to set up at login time should be included in your `~/.bashrc` file.
  - For HPCC users we highly recommend using `modules` instead of modifying the `~/.bashrc` file. (*Modules are covered in HPCC New User Training*)

```
quanah:$ vim ~/.bashrc

export WORK=/lustre/work/mahmadia
export SCRATCH=/lustre/scratch/mahmadia
export PATH="$PATH:$WORK/bin"
```

TEXAS TECH UNIVERSITY
Information Technology Division™

Basic Bash Scripting In Linux

What is Bash Script?

- **Bash** script is an executable file contains Bash shell commands which could be used to automate and simplify things.
  - **Shell** script is a text file starts with (# !) followed by the path to the shell interpreter (i.e. `/bin/bash`)

```
quanah:$ vim myscript.sh
#!/bin/bash
echo "Hello World!"

quanah:$ chmod +x myscript.sh
quanah:$ ./myscript.sh
Hello World!
```

Control flows

- The syntax of the `if-then-else` clause is as following:

```
if [ $var -eq 1 ]; then
  …
elif [ $var -ne 1 ]; then
  …
elif [ $var -gt 1 ]; then
  …
elif [ $var -lt 1 ]; then
  …
fi
```

Loops

- The syntax of the `for…in` loop is as following:

```
for VARIABLE in 1 2 3 … N; do
  …
done

for VARIABLE in file1 file2 file3; do
  …
done

For VARIABLE in `Linux command`; do
  …
done
```

Exercise #6

1. Go to your home directory
2. Create a new directory and name it “exercise6”
3. Go to the “exercise6” directory
4. Create a script file and name it “show_dirs.sh”
5. Program the “show_dirs.sh” to go over all the directories under your home directory and print the following message for each directory:

   ```
   /home/username contains --> the_directory_name
   ```

   then print the following line for “exercise6” directory

   ```
   This is the last directory --> excercise6
   ```
6. Make your script file executable and run it.

Quick Reminder

- HPCC Training Courses
  - Please check the website for upcoming User Training workshops
    - *http://www.depts.ttu.edu/hpcc/about/training.php*
- ShortCourse Survey
  - Looking forward to have your feedback on this Training Workshop
    - *You will receive a survey in your inbox from TTU ShortCourse*
- The PowerPoint slides are available online
    - *http://www.depts.ttu.edu/hpcc/about/training.php*

TEXAS TECH UNIVERSITY

Information Technology Division™