March 7, 2020

Introduction to Linux

Je’aime Powell, Texas Advanced Computing Center
jpowell@tacc.utexas.edu

Jay Alameda, National Center for Supercomputing Applications
alameda@illinois.edu

John Holly, Southeastern Universities Research Association
jholly@sura.org
XSEDE Code of Conduct

- XSEDE has an external code of conduct for XSEDE sponsored events which represents XSEDE's commitment to providing an inclusive and harassment-free environment in all interactions regardless of gender, sexual orientation, disability, physical appearance, race, or religion. The code of conduct extends to all XSEDE-sponsored events, services, and interactions.

- **Code of Conduct:** [https://www.xsede.org/codeofconduct](https://www.xsede.org/codeofconduct)

- **Contact:**
  - Event organizer: Rosalia Gomez (rosie@tacc.utexas.edu)
  - XSEDE ombudspersons:
    - Linda Akli, Southeastern Universities Research Association, ([akli@sura.org](mailto:akli@sura.org))
    - Lizanne Destefano, Georgia Tech ([lizanne.destefano@ceismc.gatech.edu](mailto:lizanne.destefano@ceismc.gatech.edu))
    - Ken Hackworth, Pittsburgh Supercomputing Center, ([hackworth@psc.edu](mailto:hackworth@psc.edu))
Learning Objectives

• After attending this training, the participants would have
  – Developed familiarity with Linux and working in the command-line mode
  – Learnt to use the basic Linux commands
  – Learnt to create and edit files on a Linux OS
  – Learnt the basics of bash scripting
  – Familiarize themselves with the user environment of the Stampede2 supercomputer (which has the Linux OS)
Please make sure your DUO is working, Globus Connect is installed, and if you are on Windows then PuTTY is installed (if needed)

To check DUO:
portal.xsede.org → “Sign In” → “MY XSEDE” → “Profile” → “Manage DUO”

To check Globus Connect:
Globus.org → “Log In” → “Endpoints”
Accessing the content for this class

Section 1: History of Linux, and Connecting to the Stampede2 Supercomputer
Linux is an Operating System (OS)

• What is an OS?
  – Software interface between the user and the computer hardware
  – Controls the execution of other programs
  – Responsible for managing multiple computer resources (CPU, memory, disk, display, keyboard, etc.)
  – Examples of OS: Windows, Unix/Linux, OS X, iOS, Android
In the beginning there was UNIX and it was EXPENSIVE!!
Where did Linux come from??

- Between 1960-1980 UNIX ruled the world!
- In 1984 GNU and in 1991 Linux were both designed as UNIX alternatives
- The two were later combined into a single kernel named GNU/Linux.

29 Million Users or .7% Market Share

Torvald

Stallman
How much does Linux cost?

- Personal Use = Free!
  - Download from the web.
  - Borrow from a friend.
  - Purchase a Disk / Flash drive and install Linux on it.
- Enterprise = Up to $10,000 US (Service)
  - Direct purchase from a Linux distribution company.
Are there choices in Linux?
How can I use Linux when Microsoft owns my soul?
### How can I use Linux when Microsoft owns my soul?

<table>
<thead>
<tr>
<th></th>
<th><strong>WINDOWS</strong></th>
<th><strong>LINUX</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Shell</strong></td>
<td>Microsoft Command Prompt</td>
<td>Gnome, KDE, BASH, Borne, C, Z, T ….</td>
</tr>
<tr>
<td><strong>Desktop Publishing</strong></td>
<td>Microsoft Office 2019/365 $249 or $69 - $99/year</td>
<td>OpenOffice (Free)</td>
</tr>
<tr>
<td><strong>Web Browser</strong></td>
<td>Edge <em>(AKA the thing I use to download Firefox/Chrome)</em></td>
<td>Firefox/Chrome/Chromium</td>
</tr>
<tr>
<td><strong>Email</strong></td>
<td>Windows Mail App/Office</td>
<td>Evolution Mail/Thunderbird</td>
</tr>
<tr>
<td><strong>Software Compiler</strong></td>
<td>Visual Studio <em>(Free as of 2014 prior $1200)</em></td>
<td>GCC, G++, Make, GFortran, OpenMPI ….</td>
</tr>
</tbody>
</table>
How can I use Linux when Microsoft owns my soul?

<table>
<thead>
<tr>
<th></th>
<th>WINDOWS</th>
<th>LINUX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shell</td>
<td>Microsoft Command Prompt</td>
<td>Gnome, KDE, BASH, Borne, C, Z, T ....</td>
</tr>
<tr>
<td>Desktop Publishing</td>
<td>Microsoft Office 2019/365 $249 or $69 - $99/year</td>
<td>OpenOffice (Free)</td>
</tr>
<tr>
<td>Web Browser</td>
<td>Edge <em>(AKA the thing I use to download Firefox/Chrome)</em></td>
<td>Firefox/Chrome/Chromium</td>
</tr>
<tr>
<td>Email</td>
<td>Windows Mail App/Office Evolution Mail/Thunderbird</td>
<td></td>
</tr>
<tr>
<td>Software Compiler</td>
<td>Visual Studio <em>(Free as of 2009, $1200)</em></td>
<td>GCC, G++, Make, GFortran, OpenMPI ....</td>
</tr>
</tbody>
</table>

*Windows Subsystem for Linux: adds Linux functionality to Windows 10*
Oh one more thing ...

Vulnerable applications exploited by cybercriminals

Sample of ~16M Ref: https://securelist.com/it-threat-evolution-in-q2-2016-statistics/75640/
Connecting to remote computers using an SSH client

PATIENCE YOU MUST HAVE

WORTH IT WILL BE
What happens during a SSH connection

```
$ssh
jhpowell@stampede2.tacc.utexas.edu
```

```
OpenSSH_7.9p1,
LibreSSL 2.7.3
debug1: Reading
configuration data
/Users/jpowell/.ssh/config
```

```
login1(597)$
```
Some Sample SSH Clients

<table>
<thead>
<tr>
<th>Windows</th>
<th>Linux</th>
<th>iOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-PuTTY</td>
<td>-Terminal</td>
<td>- SSH Term</td>
</tr>
<tr>
<td>-openSSH</td>
<td>-Secure Shell</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chromium</td>
<td>Android</td>
</tr>
<tr>
<td>MacOS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Terminal</td>
<td>-JuiceSSH</td>
<td></td>
</tr>
<tr>
<td>-iTerm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: We don’t care what SSH client you use as long as you can log into the server.
Example from MacOS Terminal App

MacintoshHD → “Applications” → “Utilities” → “Terminal”
Example from Chrome “Secure Shell App” Extension
Example from Windows PuTTY

If you have an updated copy of Windows 10 you should have ssh already in cmd!!
Where you are ...
Where you are on Stampede 2 ...
Let’s connect!

[Note: The username is your XSEDE User Portal account]

SSH to this server:

```
ssh username@login.xsede.org
```

Then `gsissh stampede2`
What is XSEDE?

“XSEDE is an NSF-funded virtual organization that integrates and coordinates the sharing of advanced digital services - including supercomputers and high-end visualization and data analysis resources - with researchers nationally to support science.”

In short, they connect scientists, researchers, and students (like you!) to large scale computing resources.
What we have done so far...

- Introduced to Linux history and the Opensource movement/indoctrination
- Used some basic Linux Commands
- Connected to login.xsede.org using your XSEDE computing resource allocation

NOTE: MAKE SURE YOU ARE LOGGED INTO stampede2
Section 2: Linux Commands Continued, Editing Text From the Command Line, Bash Scripting
Overview

• Further practice of Linux commands
• Learning to gather Linux system information
• Filesystem navigation
• File manipulation
  – *Text Editors*

**NOTE: MAKE SURE YOU ARE LOGGED INTO stampede2**
General Linux Command Line Syntax

• The command line (and Linux in general) is case sensitive

Example:
The “man” command

```
man ≠ Man ≠ MAN ≠ MaN
```

```
[jhpowell@isp02 ~]$ man
What manual page do you want?
[jhpowell@isp02 ~]$ Man
-bash: Man: command not found
[jhpowell@isp02 ~]$ MAN
-bash: MAN: command not found
[jhpowell@isp02 ~]$ MaN
-bash: MaN: command not found
[jhpowell@isp02 ~]$ _
```
General Linux Command Line Syntax

- A Linux command generally has the following syntax...

  `command -options file1 file2 ...`

Example of how to use the `man` command to see what the `ls` command does:

**Syntax:**

`man ls`
General Linux Command Line Syntax

A Linux command generally has the following syntax...

```
command -options file1 file2 ...
```

Example of how to use the `man` command to see what the `ls` command does:

**Syntax:**

```
man ls
```

Quick Tip: man pages are also on the web, eg google “ls manpage”
General Linux Command Line Syntax

- A Linux command generally has the following syntax...
  \[ \text{command} \ \text{-options} \ \text{file1 file2} \ \ldots \ \]

Example of how to use the \textit{man} command to see what the \textit{ls} command does:

\textbf{Syntax:}
\begin{verbatim}
man ls
\end{verbatim}

Command Prompt (you don’t type this!)

Quick Tip: man pages are also on the web, eg google “ls manpage”
General Linux Command Line Syntax

- A Linux command generally has the following syntax...

  \texttt{command -options file1 file2 ...}

  Example of how to use the \texttt{man} command to see what the \texttt{ls} command does:

  \textbf{Syntax:}
  \begin{verbatim}
  man ls
  \end{verbatim}

  \textbf{Command (you type this!)}

  \textbf{login3(604)$ man ls}

  \textbf{Command Prompt (you don’t type this!)}

  \begin{itemize}
  \item Quick Tip: man pages are also on the web, eg google “ls manpage”
  \end{itemize}
Ok I logged in, now where am I?

```
hostname
show the name of the server

hostname -i
show the IP address of the server

pwd
print the working path of where you are

ls
lists the files and directories
  ls -a
    lists all files
  ls -l
    lists files with a long listing
```
How can I move around?

`cd <dir>`

→ Change to the directory `<dir>`

`cd ~`

→ Change to your $HOME directory
  
  $HOME is a variable associated with your user account

`mkdir <dir>`

→ Make (create) a directory named `<dir>`

`cp </dir/file> </dir/file>`

→ Copy a file from one location to another
Create, display, and edit a file using “cat”

- You can add content to a file as follows
  
  $ cat > test.txt
  
  This is what I am entering from the console
  
  CTRL-D

- You display the contents of the file using the `cat` (short for concatenation) command

  $ cat test.txt
  
  This is what I am entering from the console

- You can append content to a file as follows

  $ cat >> test.txt
  
  Appending more lines
  
  CTRL-D
Let us check the currently available files/directories

- To list the contents of a directory, use the `ls` command
  
  `$ ls$

- To list the contents of a directory along with the time-stamp and in reverse chronological order, use the `ls` command
  
  `$ ls -tr$

- To see all files and directories, including hidden ones use the `-a` flag with the `ls` command. Hidden files have a “.” infront of them
  
  `$ ls -a$

Note: your current working directory can be checked by using the `pwd` command. Also, you can combine multiple flags in a single command.
Copying files and changing directories

- To copy contents of one file to another, use the `cp` command
  
  ```
  $ cp test.txt copytest.txt
  $ cp test.txt test3.txt
  ```
  
  One more example:
  
  ```
  $ mkdir junk
  $ cp test.txt ./junk/test2.txt
  ```
  
  (The command above copies a file to the sub-directory `junk` that exists at the current path)
  
  ```
  $ cd junk
  $ ls
  ```

- To go a level up from the current working directory
  
  ```
  $ cd ..
  ```
Exercise-1 (Part A)

- Run the following commands to make a directory:
  
  ```
  $ mkdir csula29
  $ cd csula29
  ```

- Create a file using `cat` command in `csula29` [Note: Ctrl +D to end]
  
  ```
  $ cat > test.txt
  ```

- Run the following commands in the `csula29` directory
  
  ```
  $ cp test.txt test2.txt
  $ mkdir junk
  $ mkdir junk2
  $ cp test2.txt ./junk/test2.txt
  $ cp test2.txt ./junk2/test2.txt
  $ ls
  ```
Exercise-1 (Part B)

- Run the following commands starting from the `csula29` directory that you created in Part A of Exercise-1

  
  ```
  $ ls
  $ cd junk
  $ ls
  $ cd ..
  $ cd junk2
  $ ls
  $ cd ..
  $ ls
  $ cp test.txt test3.txt
  ```
Deleting files and directories – there is no “undo”

- To remove a file, use the `rm` command
  
  $ \texttt{rm test2.txt}

- To remove a directory, use the “–r” option with the `rm` command
  
  $ \texttt{rm -r junk2}

- You can also use the `rmdir` command to remove an empty directory
  
  $ \texttt{rmdir junk2}

Note: `rmdir` command does not have “–r” option

“–r” stands for “recursive”, be careful!
Renaming files and directories, getting help, checking differences between files

- A file can be renamed by moving it. The same can be achieved by using the `mv` command
  
  ```bash
  $ mv test3.txt newtest3.txt
  ```

- Use the `man` command to get more information about a command
  
  ```bash
  $ man diff
  ```

- Use the `diff` command to see the differences in two files
  
  ```bash
  $ diff test.txt newtest3.txt
  ```
Checking the previously run commands

- Previously executed commands in a shell can be viewed by using the **history** command. For example:

```bash
$ history
1  man ls
2  ls -tr
3  ls -t -r
4  ls -tr
5  history
```
Basic commands Recap

hostname
pwd
ls
  -ltrah
cd
  ~
mkdir
(cp
history
cat
rm
mv
exit
What Applications are there?

Linux needs to know where programs are installed. The “paths” to those locations are saved as “Environmental Variables.”

- XSEDE resources use “Modules” to make this easier.
Modules Example Python3

To see what is loaded use:

```
module list
```

To view available modules:

```
module avail
```

To change use:

```
module swap python2 python3
```

*or if python2 not already loaded*

```
module load python3
```
Command Line Text Editors
Imagine if you, will a time when ...
It is up to you to choose your editor

- **nano**
- **vi**
- **emacs**

Ref: [https://www.reddit.com/r/vim/comments/5trtbj/and_the_text_editor_of_the_year_is/](https://www.reddit.com/r/vim/comments/5trtbj/and_the_text_editor_of_the_year_is/)
To exit: press [Ctrl] [X] and then N and [Enter] to not save
To exit:
press [Esc]
and then
type :q!
and [Enter]
to not save
To exit: press [Ctrl][x] then [Ctrl][c] to not save
Quick vim exercise

**Goal:** Type: 
“The quick brown fox jumps over the lazy dog.” into a text file in your home directory.

**Hints:**
- Do not touch your mouse/trackpad!!!
- `vi filename.txt` to start
- Use `i` to enter insert mode
- Use `[Esc]` to leave insert mode

<table>
<thead>
<tr>
<th>Command</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vim file.txt</code></td>
<td>open &quot;file.txt&quot; and edit with vim</td>
</tr>
<tr>
<td><code>i</code></td>
<td>toggle to insert mode</td>
</tr>
<tr>
<td><code>&lt;Esc&gt;</code></td>
<td>toggle to normal mode</td>
</tr>
<tr>
<td><code>&lt;arrow keys&gt;</code></td>
<td>navigate the file</td>
</tr>
<tr>
<td><code>:q</code></td>
<td>quit editing the file</td>
</tr>
<tr>
<td><code>:q!</code></td>
<td>quit editing the file without saving</td>
</tr>
<tr>
<td><code>:w</code></td>
<td>save the file, continue editing</td>
</tr>
<tr>
<td><code>:wq</code></td>
<td>save and quit</td>
</tr>
</tbody>
</table>
One more time - editing files using the vi editor

• To create a new file use the `vi` command (see cheat-sheet)
  
  ```bash
  $ vi test2.txt
  ```
  
  – Press `i` to start **inserting** text
  – Type some text: Hello Class!
  – To **save and quit**, press “Esc” key, and enter `:wq` (press the enter key after typing :wq)
  – To **quit without saving**, press “Esc” key if in insert mode, and enter “:q!”

• To display the contents of the file, use the `cat` (short for concatenation) command
  
  ```bash
  $ cat test2.txt
  ```
Try the three (or more) editors and choose one!
Scripting Time!

YES, YES

LET THE COMMAND LINE FLOW THROUGH YOU
Linux Scripting

- Instead of typing commands directly in the shell, you can place the commands in a file, grant execute permissions to the file, and then run the file from the command prompt (automation).

- Such a file that contains the Linux commands is known as a shell script.

- Three commonly used types of shells: Bourne shell, C shell, Korn shell.

- We will work with Bash shell script.
Creating and running a simple shell script

• Create a file named `myScript.sh` and put the text below in it
• You may use `vi` or `nano` or any other editor of your choice

```
#!/bin/sh
echo "hello world"
module list
hostname | mail -s "Hi from:" your@email
```

• Change the permissions on the file to 700, or just use, “+x”
  ```
  chmod 700 myScript.sh
  chmod +x myScript.sh
  ```

• Execute the file by typing `./myScript.sh`
What is that `chmod` command?

- File permission can be viewed with `ls -l`
- There are three (3) file permissions for **owner**, **group**, and **everyone**:
  - r = Read files (using more, cat, etc.)
  - w = Write files (using >, editors such as vi, etc.)
  - x = Execute commands in a file (executables, etc.)

- Each file has three sets of permissions – read, write and execute (`rwx`)

- Person creating the file is the owner or user and can modify permissions as desired
  - Owner can modify permissions on files to grant or revoke access to other users
How did I send an email!

- “|” (known as a pipe) symbol is used for input redirection
  $ hostname | mail -s "CSULA class" jpowell@tacc.utexas.edu

or

- “<” symbol is used to send a file as the text
  $ mail -s "CSULA class" jpowell@tacc.utexas.edu < test.txt
Section 3: Putting it all together: From code to Job Submission

WHEN YOU USE THE COMMAND LINE

HACKERMAN
Objectives

Student will…
• Connect to the XSEDE SSO Login Node
• Use gsissh to access Stampede2
• Submit a job to Stampede2
• Use Globus to transfer files

NOTE: MAKE SURE YOU ARE SSH’D INTO stampede2.tacc.utexas.edu
XSEDE Prerequisites

• An active xsede.org account

• An active allocation on a XSEDE resource
  – [https://portal.xsede.org/group/xup/allocations/usage](https://portal.xsede.org/group/xup/allocations/usage)
Let’s put it all together!

1. Download some software
2. Transfer it using Globus.org
3. Extract (`unzip`) the software
4. Edit the submission script
5. Submit (`sbatch`) the job
Let’s download our sample program

Go to the following site and download the zip file to your computer.  
[Note: Remember where you save it, you will need it again]

https://github.com/jeaimehp/xseedexample
Transferring data using Globus.org

Quick Tip: Move xsedeexample-master.zip to default globus location eg, Documents for Windows, home directory for Macintosh

1. Go to globus.org and “Log In”
2. Go to “Your Collections” and choose your “Endpoint”
3. Select the “Panel” with two(2) columns
4. Add the Collection “XSEDE TACC Stampede 2”
5. Select the file(s) you want to transfer and click “Start”
   a. xsedeexample-master.zip
6. Click “refresh list” to see the new files when completed
Submitting a job script (SLURM scheduler)

1. `ssh` to `login.xsede.org` then `gsissh` to `stampede2`
2. `unzip xsedeexample-master.zip`
3. `cd` to `xsedeexample-master`
4. Edit (use either `nano` or `vi`) `example_submission.bash` to modify the email address to yours

```
#SBATCH --mail-user=jpowell@tacc.utexas.edu
```

1. `sbatch example_submission.bash`
2. `squeue -u username`
   
   Tip: Use “whoami” command to find `stampede2` username!
Whew!!! That was a lot ....
DIRECT-STEM Training Survey

- At the end of this training, please complete a short survey about this module. We value your feedback, and will use your feedback to help improve our training offerings.
- Slides from this workshop will be available at http://hpcuniversity.org/trainingMaterials/249/
  - And are currently available at https://bit.ly/2TIShbD