Research Computing at UC

Linux 101

George Turner, Chief Systems Architect
turnerg@iu.edu
Pervasive Technologies Institute, UITS/RT, Indiana University

Contact information concerning any issue related to today’s topic: **arc_info@uc.edu**

First things first…

- This is an introductory tutorial; if you are interested in something more advanced please start the conversation by sending an email to **arc_info@uc.edu**
- This tutorial is focused on using Linux in a research computing environment; i.e. using the command line from a terminal.
- To login to the ARCC you must be on the UC campus or using the UC VPN.
- People are standing by to help.
- While you are logging in, we will cover basic terms, history, etc.

Logging into the ARCC

- ssh *USERNAME*@arcc.uc.edu
  - ssh is a command to connect from your laptop to another system. It is natively available on Macs & Linux boxes. PuTTY is a popular ssh client for Windows (https://www.putty.org/)
  - Your USERNAME will be your UC 6+2 for UC individuals or as otherwise identified in an email you received.
  - If you have problems connecting, instead of arcc.uc.edu try the IP address 10.127.5.254

First and foremost!

- This is a live presentation with the goal of showing how to use the Linux command line interpreter.
- The slides for this tutorial are less important than trying the commands as they are presented.
- At some point I’m all but certain this will become less structured and more of a stream of consciousness. Please ask questions!
- And most importantly, never preface a live demonstration with anything more predictive than “Watch this…”

The man pages are your friend

- The help command for Linux
- man <command>

The Google machine (search engines) are your friends

- Other search engines:
    - bing : Window’s users
    - duckduckgo : privacy concerns
    - altavista : before Google there was AltaVista; old timers like me
    - YouTube : most tutorials are videos

Getting started

- **Linux vs Unix**
  - Linux is open source developed by the Linux community
  - Unix was developed by AT&T Bell Labs, is not open source, and is copy righted
- **ssh** – a command line interface used to connect between servers
  - Native to Linux and Macs
  - putty for Windows (https://www.putty.org)
- **#** - comment in Linux; i.e. whatever follows the hashtag will not be executed
- **stdin, stdout, stderr** : input and out put IO streams; e.g. input from a keyboard, output to a terminal and output to the terminal for error messages
- **directories** – the same as folders on a Windows or Mac. Contains files. Some important directories are the current working directory (.) and the HOME directory (~)

Getting started (continued)

- **man** : the “help” for Linux; e.g. man bash, man ls, man salloc, etc.
  - **man –k <keyword>** : give me all the man pages with <keyword>
- **pwd** : present working directory; what directory am I sitting in
- **ls** : show me the files in this directory
- **cd** : change Directory
  - cd dd2020-hpc - move down into the dd20202-hpc directory
  - cd .. - move up one directory
- **cp** : copy a file
  - cp –rp /opt/ohpc/dd2020-hpc .
    - Recursively copy the directory at /opt/ohpc/training/fdw-L101-210122 to my current directory

Getting started (continued)

- **cat** : stream a whole file to stdout
- **more** : stream a file one page at a time to stdout
- **less** : like more but you can move up and down in the file using the arrow keys; exit with q
- **module** : change your Linux environment; e.g. add applications
- **echo** : repeat the argument to stdout (the terminal)

Getting started (continued)

- **PATH** : a list of directories where applications are; to see the current path do `echo $PATH`
- **ls** – a command for listing files and the contents of directories
- **environment** – usually thought of as the state of the shell variables
- **shell** – the command line interpreter. Examples include sh, bash, zsh, korn, tcsh, etc.
- **process** – a running application. Use the commands ps or top to see running processes
  - Parent, child processes
  - Process ID (PID)
- **Editors** - an application for creating and modifying text in a file

Editors available on Linux

- **nano**
  - Easiest to use but not always available
  - Menu at the bottom of the screen
  - Use control-x (^x) to exit
- **vi** (also sometimes called vim)
  - always available on a Linux system
  - Cheatsheet available at https://devhints.io/vim
  - enter insert mode with `i`
  - Exit insert mode with Esc key
  - Exit editor with Esc-wq to save work
  - Exit editor with Esc-q! to not save work

Editors available on Linux (continued)

- **emacs**
  - very complicated
  - very powerful
  - Written in lisp; very extensible
- GUI editors
  - Works great if you have X11 working via the ssh connection
    - To see if X11 is working, do **echo $DISPLAY**
  - **gedit**
  - **nedit**
- **sed**
  - Meant for pipeline processing; e.g. cat file | sed ‘s/this/that/g’ > out.txt

Once logged in, setting up our environment…

set up our environment

```
module avail
    # what modules are available to load
module spider
    # more detailed listing of modules
module load gnu9
    # loading a module into your environment
module list
    # what modules are loaded
```

Once logged in, let’s prep our environment…

```
# copy over the files needed for the tutorial
cp –rp /opt/ohpc/training/fdw-L101-210114 .
    #               note the space and a period ^
cd fdw-L101-210122
ls -al
```

Compile and run our hello world example

```
# start with the worlds simplest C program
less simple.c
        # remember to type q to exit out of less
gcc -o simple simple.c
./simple
# doesn’t look like much happened; but, it does everything that any program does except work (work as in the physics sense of the word)
```

Compile and run our hello world example

```
less hello.c
        # remember to type q to exit out of less
gcc -o hello hello.c
exit
```

Commands and Concepts

- kernel

- command syntax
  - <command> <parameters>
  - ls -al

- process
  - ps -ef, ps aux
  - top
  - w, who
  - shell (sh, bash, tcsh, zsh)

Commands and Concepts

- environment
    - printenv
    - variables
    - ~ . ? _ PATH MANPATH

- alias
- quotes (",')
- backquote

Commands and Concepts

- file system
  - directories, folders
  - /   (the root directory)
  - /var
  - /tmp
  - /home
  - /opt
  - /usr
    - /usr/bin, /usr/man, /usr/lib, /usr/local

Commands and Concepts

- files
  - executables or applications
  - tab completion

- pipes
- > < |

- sockets

- cat
- wc
- echo
- more or less
- head or tail

Commands and Concepts

- cat
- wc
- echo
- more or less
- head or tail
- which
- ls /usr/bin
- dos2unix unix2dos
- tar, zip
- gzip gunzip

- container
- file
- process

Commands and Concepts

- Containers (Docker, Singularity)
  - file
  - process
  - VM vs Container

Some Other Introductory Tutorials & References

- **Basic Linux introduction**
  - https://training.linuxfoundation.org/training/introduction-to-linux
- **Beginner’s Guide to the Bash Terminal**
  - https://www.youtube.com/watch?v=oxuRxtrO2Ag
- **Cheatsheats**
  - https://devhints.io/
  - https://devhints.io/bash
- **Introduction to High Performance Computing Systems, ADACS**
  - https://www.youtube.com/watch?v=7zJUceJiYxQ
- **tutorialspoint**
  - https://www.tutorialspoint.com/unix_commands/
  - https://www.tutorialspoint.com/unix_commands/bash.htm

Singularity (docker) References & Other Tutorials

XSEDE/Container_Tutorial
https://github.com/XSEDE/Container_Tutorial/tree/master/Gateways2020

Sylabs (developers of singularity) tutorial videos
https://sylabs.io/videos

naked-singularity registry
https://github.com/mkandes/naked-Singularity

ARC Clusters (ARCC)

- **ARCC-1** - a.k.a. the *Pilot* cluster
  - Explore what is involved in establishing research cyberinfrastructure
  - Acquired in three phases
    - Started with Dr Prashant Khare’s startup cluster
    - Addition funded by the Office of Research
    - Expansion of Dr. Khare’s original cluster
  - Dell PowerEdge 36 C6420, dual Intel Gold 6148 2.4 GHz, 20 core CPU (40 total cores), 192 GB RAM
  - Dell PowerEdge R740xd Storage node, 96 TB raw
  - Dell PowerEdge R740 GPU server dual Intel Gold 6148 2.4 GHz, 20 core CPU (40 total cores), 192 GB RAM, Nvidia V100-32G GPU
  - 100 Gb/s OmniPath switch fabric
  - Initial integration Dec-2018 and installed in the UC Data Center Jan-2019

ARC Clusters (continued)

- **ARCC-2** will be UC's first production cluster
  - Funded in part by an NSF MRI grant with Dr Prashant Khare as the project PI.
  - Additional resources provided by UC Office of Research.
  - Architected for longevity and expandability using today's (2020/2021) best technologies. We will be adding to this cluster over the next three to five years.
  - Strategic integration of compute, networking, storage into cohesive environment.
  - Expected production date Q1/2021

ARC Clusters (continued)

- **ARCC-2** (continued)

- *Excuse our Dust : Under construction*
  - Currently 19 compute nodes and one GPU node with two Nvidia V100 GPUs
  - Compute nodes: HPE Apollo 2000, dual AMD EPYC 7452 (Rome), 32 cores (64 total) 2.3GHz, 256 GB RAM, Gen4 PCIe bus
  - GPU nodes: HPE Proliant DL 385 Gen10+ GPU nodes, dual AMD EPYC 7452 (Rome), 32 cores (64 total), 1TB RAM, dual Nvidia A100-40 GPUs , Gen4 PCIe bus
  - One login node, one management node

ARC Clusters (continued)

- **ARCC-2** (continued)
- *Excuse our Dust : Under construction*
  - Funded expansion includes:
    - ~1.5 PB of storage and two data transfer nodes in Q1/2021
    - Additional compute and GPU resources probably in Q2/2021
      - Timing dependent upon completion of upgrades to the data center facilities
      - Component costs at the time acquisition; but, we're looking at adding ~50 compute modes (AMD Milan) and half dozen GPU nodes (AMD Milan and NVIDIA V100 or A100 GPUs)

SLURM – Simple Linux Utility for Resource Management

- Combined resource manager and scheduler
- Resource manager
  - Inventory resources (nodes, sockets, cores, memory, licenses, GPUs, etc.)
- Scheduler
  - Map job requirement onto available resources
  - Predict when resources will become available
  - complex scheduling algorithms e.g. optimize for memory or network topologies, fairshare scheduling, advanced reservations, preemption resource limits; e.g. access restrictions, user & group quotas, etc

Some basic Slurm commands

Slurm Cheat sheet available at https://slurm.schedmd.com/pdfs/summary.pdf

- **salloc** : allocate resources for an interactive job
- **sbatch** : queue a batch script and allocate resources for the job when they become available
- **srun** : utilize resources allocated for a job (or job step) via the salloc or sbatch commands
- **squeue** : check the status of queued jobs
  - **squeue --start -j <jobid>** # when will my job start?

Some basic Slurm commands (continued)

Slurm Cheat sheet available at https://slurm.schedmd.com/pdfs/summary.pdf

- **sprio** : view factors comprising a jobs priority
- **sinfo** : view the state of the system
- **scancel** : cancel queued or running jobs
- **sattach** : attach stdin/out/err to a running job
- **sbcast** : copy a file to local storage (e.g. /tmp) on all the nodes allocated to a job

Questions, Comments, Suggestions?

Project website: https://research.uc.edu/arc/
Project email: arc_info@uc.edu
Direct email: turnerg@iu.edu

License Terms

- Turner, G.. 2021. Linux 101, UC Faculty Development Workshop 22-Jan-2021, University of Cincinnati, Cincinnati, OH.
- Items indicated with a © are under copyright and used here with permission. Such items may not be reused without permission from the holder of copyright except where license terms noted on a slide permit reuse.
- Except where otherwise noted, contents of this presentation are copyright 2020 by the Trustees of Indiana University.
- This document is released under the Creative Commons Attribution 3.0 Unported license (http://creativecommons.org/licenses/by/3.0/). This license includes the following terms: You are free to share – to copy, distribute and transmit the work and to remix – to adapt the work under the following conditions: attribution – you must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). For any reuse or distribution, you must make clear to others the license terms of this work.