CST8177

Winter 2013

Linux Operation Systems II

aka

Linux II

Today's Agenda

- Introductions
- Course Objectives
- Course Outline
- Lectures
- Labs
- Evaluation

About Me

Todd Kelley

Email: kelleyt@algonquincollege.com

Office Hours: send me an email to make an appointment

I started learning Unix in 1987.

Some of what I like about Unix/Linux:

- It is “humankind’s” operating system, as opposed to a product put out by a company with the primary purpose of making that company money as opposed to making sense
- Unix/Linux skills apply to so many computer systems: almost everything except MS Windows (but there’s Cygwin for MS)
- The Unix/Linux skills I devoted effort to in 1987 are still relevant today. For example, I learned vi in 1987, and have used it ever since – no worries about a company changing it on me

More to like about Unix/Linux

- There is a Unix “do one job and do it well” philosophy
- The second part of the philosophy involves putting the smaller pieces together
  - Unix utilities: sort, find, grep, ls, sed … many others
  - Solve bigger problems by connecting these programs together in a pipeline (filters, the ‘|’ character)
  - command sequences or pipelines you use often can become scripts: your own custom programs

What about you?

Introduction Activity

- Form into about 10 groups
- For several minutes, discuss in your group
  - What do you like about Unix/Linux
  - What do you not like about Unix/Linux
  - What's hard about Unix
  - What's easy about Unix
- In 5 min or so, I'll ask each group to report the main or most interesting points brought up

Course Objectives

- To increase your command line skills
- To add to your knowledge of Linux tools
- To learn basic system administration
- To learn how to design, write, and debug a script
- To provide the required background for the successor courses in later semesters

Linux versus Unix

- Linux is one of the “Unix-like” operating systems
- Much of what we learn in this course applies to Linux, Unix, MacOS X, and others
- As a Computer Systems Technician, you’ll want to be able to work on them all
- Different Linux distributions to a CSTech are a bit like different brands of car for a valet-parking attendant, or for you if you rent a car
  - “Sorry, that’s a Toyota, and I only drive Hondas”
  - Why does that seem like a strange thing to say?

Speaking of Linux Distributions

- Fedora: Red Hat's community research and development distribution, End of Life (EOL) every six months
- Red Hat Enterprise: Red Hat's production distribution on which they base their Red Hat Network and IT services, EOL approx ten years
- Oracle Linux: Red Hat Enterprise
- CEntOS: Red Hat Enterprise with all of Red Hat's trademarks removed (we'll use this one)
- Debian: a well respected Linux distribution
- Ubuntu Desktop: a popular Desktop-oriented distro based on Debian, maintained by Canonical (six month cycle)

EOL

- Why does EOL matter to a sysadmin?
- Imagine you are sysadmin of a large server that services (hundreds of) thousands of users around the clock
- Applying a major upgrade to a server like that causes you stress (in real life, you'd stage the new version on similar hardware, if you have it, try to load it and test it as best you can, and then you sweat when it goes live)
- Think of famous incidents (Google, Rim, etc)
- You don't want to have to do that every six months

EOL cont’d

- Surely, running ten-year-old software cannot be good?
- What about new security vulnerabilities?
- Answer: Red Hat back-ports security fixes
- The versions and functionality of all the modules that make up Red Hat Enterprise are kept the same, but Red Hat provides patches that fix security vulnerabilities
- This allows a much more stable path to the future, keeping the system running without having to go through the stress of major upgrades

Meanwhile, back at school…

Prerequisite Course

CST8207, Linux I

Yet to Come

CST8213, Linux III

CST8230, IT Security Fundamentals

CST8231, Network Services

... and perhaps more

Brief Course Outline

1. **Control system processes**

the kernel process table; the boot process; log system services; user processes; runlevel tools; task scheduling

2. **Control user access to system resources**

user and group accounts; a password policy; file permissions

3. **Setup and maintain file systems**

volumes; single and multiple file systems; file system integrity

Brief Course Outline

4. **Automate administrative tasks using scripting**
operating system interface; process automation bash scripts

5. Other automation tools (time permitting)
stream editor (**sed**) and **awk**

Grading scheme

I. Practical 35%

a) Labs: 25%

b) Online Quizzes: 10%

Online quizzes will be designed to be completed as you work at a computer and answer questions about that work.

I. Theory

a) The two mid-term tests are spaced about 5 weeks apart. Each test may include questions from all material covered to date, but each one will focus on the material from the preceding weeks. They contribute 25% to your total mark, 10% from the first and 15% from the second.

b) There will be a final exam at the end of the semester covering the whole course, for 40% of your total mark. Note that the final may be different in style from the mid-terms, since there is more time available (2 hours).

NOTE: You cannot “make up” an exam or test, either for a poor grade or missing the examination entirely.

In summary:

	Practical –					
		Lab assignments	25%			
		In-lab exercises	10%			
		Total-P		35%		
	Theory –					
		Mid-term exam 1	10%			
		Mid-term exam 2	15%			
		Final exam (2 hours)	40%			
		Total-T		65%		
		Grand Total		100%		

Lectures

- In lectures I present the subject matter, trying to help you absorb it – ask me for more examples, ask me to rephrase, repeat, … it’s all about you
- At the very least, it’s a study session with the person who knows all the answers on the tests and exam
- Attendance will be taken.
- Ask questions about topics you don’t understand
- Arrive on time; it's not nice for your classmates to have someone interrupt their lecture.
- Don’t carry on conversations amongst yourselves during lectures (unless I ask you to) – it makes it hard for students around you to hear me
- Turn off your cellphone ringer and other similar noise makers
- Lectures are a good way to get into the habit of punctuality: an important real-life job skill

Labs

- The lab activities are designed to be similar to real-world system administrator activities.
- You will figure things out and check whether you really remember and understand
- They will give you practice at looking things up in the manual
- They are a form of study in preparation for the final exam
- Anyone who doesn’t come to labs is at a drastic disadvantage
- Be honest with yourself:
  - If you don’t do the labs in the scheduled lab period each week, when are you going to do them?
  - When are you actually going to work with the material and really work on your sysadmin skills?
- Every section will have identical assignments and due dates
- If you're having difficulty, see your lab instructor – that’s what he’s there for. Don't delay!
- Lab Attendance is not mandatory, but PLEASE be honest with yourself about preparing for the midterms and final exam.

Attendance

You are expected to attend labs and lectures to get full value for your tuition money. (If you didn't need a professor to explain this material, why aren't you taking a pure online course?) Only lab attendance is recorded. If you have to miss a lab, let your lab instructor know. Unexplained absences will ruin your chance to get a job recommendation from your lab instructor, and will force us to track you down to find out your status in the course. You don't have to attend, but you do have to let us know when you don't.

The 7 Habits of Highly Effective Students

1. Lectures: Attend them all. Arrive on time. Stay alert and attentive. Sit close to the front where you can easily see and hear. Print lecture notes ahead of time and bring them. Read the topic the night or morning before each lecture. Actively participate in the lecture by considering the material being presented, asking relevant questions, and taking notes on important areas and examples.

2. Labs: Attend them all. Arrive on time. Prepare for labs in advance by reading the lab material and doing the lab preparation work. Start working as soon as possible. Prepare for any demo of your work well before the end of the lab. If there is a lecture portion in the lab, see Lectures above.

The 7 Habits of Highly Effective Students

3. Assignments: Avoid delay; start at once. Read the requirements very carefully. Ask questions of your professor so you understand the entire assignment. Meet target dates. Follow all submission requirements. Ask your professor for clarification if you don't understand them. If you should have done better, make sure you understand where and why you were wrong.

4. Study: Establish a routine. Choose good times and locations. Set achievable goals. Start immediately. Review course material regularly by topic. Study from your lecture notes, your lab work and notes, and previous tests and assignments. Understand where you lost marks. Keep a file for each course with all your notes, lab preparation material, completed lab work, and returned assignments, tests, and quizzes. Increase your study hours before a test, especially on recent topics.

The 7 Habits of Highly Effective Students

5. Tests: Read over the test instructions carefully before starting to write. Answer the questions in any order, but first read over the whole test. Budget your time, by marks. When an answer takes too long, switch to another and return to it later. Bring spare pens or pencils, an eraser, and a short ruler.

6. Timing: Work every day to your defined schedule. Plan to spend at least 1 to 1½ times as much time on study and assignments as you spend in class. See your program flowchart for a time estimate (x/y/z: x, lecture hours; y, lab hours; z, study hours; all times are per week). Stick to your schedule but if you fall behind do extra work to catch up.

The 7 Habits of Highly Effective Students

7. Professors: Learn their office hours, whether they take appointments, and their use of email. Ask for help when you need it, with prepared questions. You should meet with your Academic Advisor at least annually about your progress in your program.

CST8177 (Linux II) is a 3/2/4 course:

- lectures are 3 hours weekly;
- labs 2 hours;
- study 4 hours weekly for an average student;
- you may need more, or less;

Textbooks and stuff

"A Practical Guide to Fedora and Red Hat Enterprise Linux"
4th edition (or later), by Mark Sobell, Prentice Hall,
ISBN 0-13-706088-2

Ian Allen's CST8207 Course Notes

http://teaching.idallen.com/cst8207/12f/notes/

You will also VMware installed on your system so that you can create Linux VMs to work on as needed in Labs and for Online Quizzes.

There are also many excellent online resources, but remember Sturgeon's Law: "90% of everything is crud".

•There is a lot of misleading, poor, or downright bad information out there, so check and re-check.

•As an example, you can go and visit the Jargon File at <**http://catb.org/~esr/jargon/html/**>, specifically the entry <**B/bullschildt.html**>, which says:

bullschildt: /bul´shilt/, n.

[comp.lang.c on USENET] A confident, but incorrect, statement about a programming language.

man Pages

- All Linux and UNIX systems have a set of online documentation called the **man** pages – ***man*** for ***man***ual, naturally.
- Some systems will also have other information sources and may have a viewer available from the GUI desktop.
- Man pages are also being joined by alternate sources of information, like **info**. There are often still referred to as **man** pages, though.
- Use the **man** pages, often. In other words, **RTFM**.

  “Read The Fine Manual”
- Use this command to learn about using the **man** pages:

  man man
- Don't ask a question of your neighbour in the lab (or me) until you have checked the relevant **man** pages.

- You can do a keyword search by entering either of the following two commands (they are equivalent):

  man -k <some keyword>

  or

  apropos <some keyword>

- You can search for strings by entering /**string[ENTER]**, where **string** is what you want to find. To search again, simply enter ***/*** alone. To search backwards, use **?**
- See **man less** (really! **less** is now **more**) for details.
- The general statement for **man** is of this form:

  man [-<options>] [section] title

Find out about the **man** section numbers, what the main sections identifiers are and what they mean.

For example, **crontab** is found in both sections 1 and 5, so you must use **man 1 crontab** or **man 5 crontab** to select it (1 is the default here). Reference to one particular form of a command is usually shown in a form like **crontab(1)**, **crontab(5)**, or even **crontab(1p**) for some purposes.

How to read technical material

- **Read the words**
  - What a concept!
  - No, really: ***read*** them (until you understand them)

How to read technical material, 2

- **Take some notes**
  - Did you come across a key item?
  - Is it useful, interesting, or likely of future value?
  - Does it seem puzzling or contradictory?
    - Ask me about it!

How to read technical material, 3

- **Re-read sections**
  - Look at your notes for issues and concerns
  - Can you resolve an apparent contradiction?
  - What does the relevant **man** page have to say?
  - Test it on a computer if appropriate
  - Check another textbook or try googling for more information

How to read technical material, 4

- **Getting good at browsing and reading technical material (manuals) is part of becoming a CSTech**

What is an operating system?

"The operating system manages the resources of the computing environment by providing a hierarchical file system, process management, and other housekeeping functions [so that the user is not burdened with these tasks]."

-- "The UNIX System",
S. R. Bourne, 1983

What is an operating system?

Practically speaking, the operating system manages:

- **Processes**
  user application programs as well as essential services, such as the **cron** service running as **crond** and known as the **cron daemon**; your shell (command line) is a process – when you run ls, that's another process
- **Resources**
  the allocation of processor time, memory, and I/O devices among the various processes which use them;
- **File System**
  including all I/O devices and some things made to look like devices (such as **/dev/kmem** or **/proc** in Linux);
- **Security Services**
  to protect against inadvertent as well as malicious damage to the file system and other resources (**login** is only the beginning).

An Operating System Structure

- Three software layers are typically used to describe the Unix or Linux system:
- Kernel space: The **kernel** layer: This runs the hardware and allocates resources, sharing them where necessary. The **file system** is often a separate process, as are other parts of the kernel and various service **daemons**. The kernel provides some of its functionality to the next layer above through the kernel's system calls.
- User space: The **shell** and other user processes layer: When you type in a command, it's to the shell at the command-line interface (or CLI). Some commands are built-in to the shell (e.g., **cd** for change directory) and others are separate programs (like **grep**, global regular expression print). User space processes use **system calls** to get kernel services

The First Lab

- a review exercise based on the material from the pre-requisite course.
- http://teaching.idallen.ca/cst8177/13w/notes/assignment01.html
- Tasks:
  - read the task
  - understand the task
  - what command do you need?
  - browse/grep the course notes and man pages
  - complete the task