CST8177: Linux Operating System II
Winter 2015

Linux

many flavours, many choices. Freedom

Week 01

Why Unix/Linux?

• Some of the character about Unix/Linux:

➢ Multitasking, multiuser

➢ It is "humankind's" operating system, as opposed to a product with the primary purpose of making money

➢ Architecture independence

  ➢ From embedded systems to IBM mainframes

➢ The Unix/Linux skills are still relevant with improvement

➢ Demand load executable

  ➢ Only the part of a program that is required for execution is loaded into memory.

More 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 ... 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
  - What distributions do you know

- 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 know 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 approximately ten years

- Oracle Linux: Red Hat Enterprise

- CentOS: Red Hat Enterprise with all of Red Hat's trademarks removed (we'll use this one, CentOS 6.5)

- 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)
- 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 stay the same, but Red Hat provides patches that fix security vulnerabilities in those old versions
- This allows a much more stable path to the future, keeping the system running without having to go through the stress of major upgrades
- Still have to go through major upgrades, but not every six months!

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.
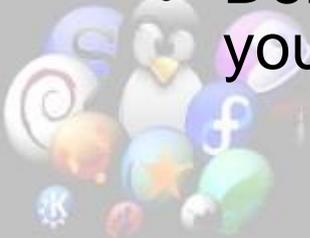
- You can do a keyword search by entering either of the following two commands (they are equivalent):
  man -k <some keyword>
      or
  apropos <some keyword>

- 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, 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

Four Components of a computer system

Adobe Reader 9
AisleRiot Solitaire
Amarok
Brasero Disc Burner
Calculator
Character Map
Cheese Webcam Booth
Desktop recorder
Dictionary
Disk Usage Analyzer
Empathy Internet Messaging
Evolution Mail
Evolution Mail and Calendar
Firefox Web Browser

Operating system

NET2003
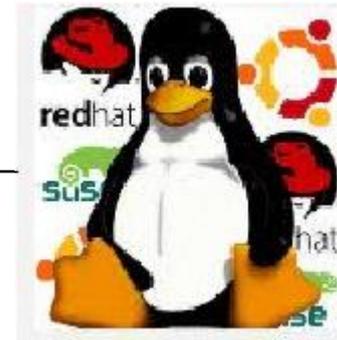
An Operating System Structure

- Two 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