

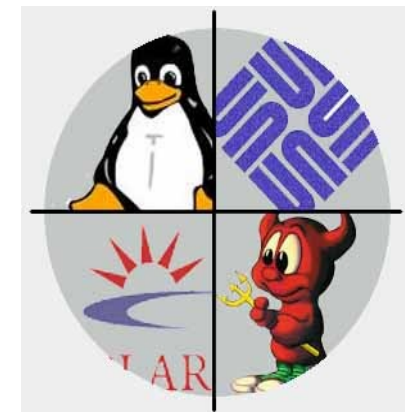
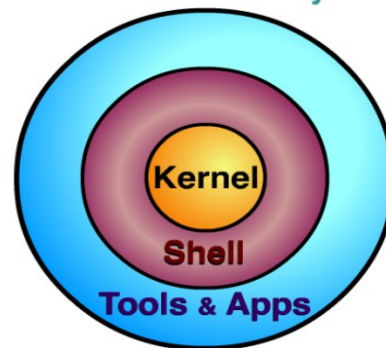
Spring 2009

Modern Computer Science in a Unix-Like Environment

<http://www.seas.upenn.edu/~cis399ux>



Parts of the UNIX System



What is Unix/Linux?

- Linux itself is an Operating System Kernel
- “Linux” is often used to refer to “GNU/Linux” which is an entire software stack
- Unix is also an operating system stack.

What is an “operating system”

- See CIS-380
- A simple explanation:
A software system that is always running which provides an interface between real hardware and everyday “user” programs.

Unix

- Unix is one of the oldest operating systems, originating at AT&T Bell Labs around 1971.
- Written in C, at the time this was a revolutionary idea. All older operating systems written in lower level assembly.
- C is portable! (Can compile to many different machine types.)

GNU is Not Unix

- GNU project is a software system/stack, of completely free software replacements of the original Unix applications.
- Examples
 - ls
 - screen
 - gcc
 - gdb
- Linux is **NOT** GNU software – GNU/Linux is a common combination of software

Linus's Minix (Linux)

- “I called it Linux from the beginning It was Linus with an x... the x is mandatory it just had to be there, its like a law or something” - Linus Torvalds
- Started as a personal project that advanced very quickly and became widely used.

Why Linux?

- You can go online and read articles from Richard Stallman about why “free software”.
- But why Linux? Why is this important for CIS students?

Empowerment

- In a GNU/Linux environment its possible to really understand at a high level everything happening in your computer, and even change things if needed.
- This gives one an extreme geeky sense of power, and that is the goal of this class: to help you achieve a kind of geek nirvana.

Linux Access

- This means you need (convenient!) access to a Linux machine. Good news!
- SEAS can provide you with a free eniac account
 - Easy, but you can't mess with things
- You can setup your own machine
 - Hard but gives you 100% control. Expensive (requires hardware).
- You can virtualize your own machine.
 - All the benefits of your own machine without needing new hardware.

Connecting to Eniac

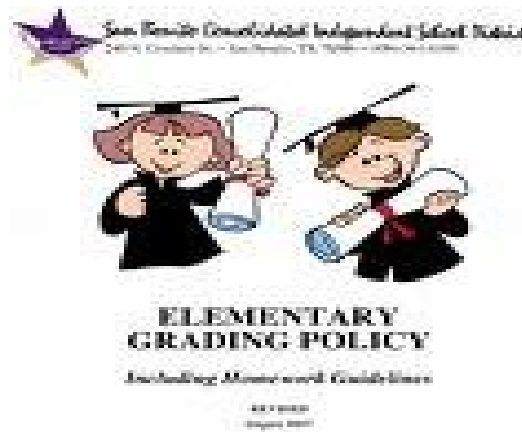
- From Windows
 - Download “putty”
 - Connect to eniac.seas.upenn.edu
 - Enter username / password (same as your seas email)
- From a Linux machine:
 - In a terminal type
 - `ssh USERNAME@eniac.seas.upenn.edu`

Virtual Box

- <http://www.virtualbox.org/>
- Allows you to run another computer from within your current computer!
- Free to download!
- Then download an Ubuntu install CD
 - <http://www.ubuntu.com/getubuntu/download>
- Ubuntu has an easy graphical installer for within virtual box.



Now, the boring stuff. (read: grading)



Course Syllabus

- Course will happen in two parts
 - Going to cover lots of the basics, get everybody comfortable with using Linux.
 - Includes using Linux at the terminal, bash scripting, emacs, other common Linux tools
 - “How Linux Works”.
 - Talk about the software stack that is used on a Linux machine. We’ll talk about how to make your own kernel, how to configure your own X server etc.

Syllabus of the Basics


- The following are some of the high level topics we'll discuss.
 - Linux Distributions
 - File Systems / Unix File System
 - Unix / Linux Shell
 - Emacs
 - Hardware
 - Kernel
 - Booting
 - Basic Networking / SSH
 - Version Control

Homeworks

- There will be homeworks.
- They will be given with plenty of time to do them.
- They will be very easy if you've been keeping up in lecture and practising on your own.
- They will always be “checked.”
 - Will get number grades.
 - Grades will frequently be: 0 or 75 or 100 i.e. you didn't do it, you missed a big part of it or you got it.

Grading

- Grading will be:
 - 20% Class Participation
 - 25% Homeworks
 - 25% Graded in class assignments (a.k.a. quiz)
 - 30% Final Project



Thats all nice and fine but...
How do I get an A?

How to get an A

DO IT YOURSELF!



Do it yourself!

- You will gain much more in a course like this if you take what you learn every class and experiment with it on your own.
- A lot of this course can be learnt simply by sitting down and browsing Internet tutorials.
- My job is to make learning this easy, and to provide motivation and assistance along the way.

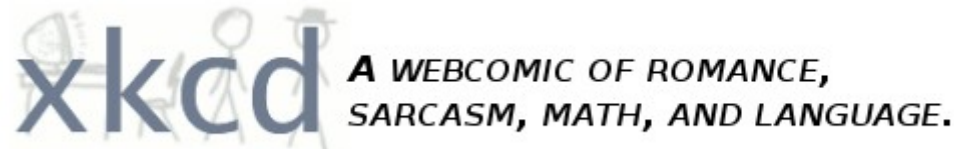
Recommendation

- This class meets just a single hour a week.
- However, you should spend 3-4 hours per week on the course.
- So what should you do for the other 3 hours?
- Get that Linux access setup and play with it!

Pace

- In light of the expectation of a few hours a week of generic usage of topics taught in class the overall pace shall be quick.
- It is expected that things taught in class will be practical enough to be used for legitimate purpose outside of class and reinforced in that manner.

Homework 0 - XKCD



- Read all of the comics and email me or ask in class about any comics which you don't understand.
- If you don't understand the romance part I can't help you, but if its CIS or generic computing knowledge related (which a lot of XKCD is) then it's appropriate to ask.
- This homework is ongoing throughout the entire semester, and ungraded.

Homework 1a

- Get access to a working Linux setup either through eniac (just getting access) or through VirtualBox with a basic Ubuntu install. Installing Linux on new hardware is too advanced for now.
- My recommendation is to go with VirtualBox.
- Due next week and needed for Homework 1b (later).
- Feel free to email class mailing list with setup questions!