CSC161 2011S *Imperative Problem Solving*

Class 03: Linux Files and Permissions

Held: Wednesday, 26 January 2011

Summary: We conclude our initial explorations of Linux.

Related Pages:

- EBoard.
- Lab: GNU/Linux III.
- Reading: Raymond 1.

Notes:

- Please do not use the elevator if you do not need to.
- I have started responding to assignment 0. Expect all responses by Friday. I will also post summaries
- I'm still trying to master the art of dealing with 30+ questions on a reading. You can find many of the answers on the Web page for the reading.
- Due tonight: Assignment 1 (Linux stuff that follows up on the first three days of class).
- Due Thursday night: K&R, Chapter 1. (Yes, reading tapers off after this first week.)
- EC: This Thursday at 4:15: Microsoft on "How to Nail a Technical Interview" (@CDO).
- EC: This Thursday at 7:00 p.m.: Waiting for Superman (@Harris Cinema).
- EC: Next Thursday at 4:30: Summer opportunities in CS (3821).
- Due to unforeseen circumstances, CS Table will not be held this Friday.

Overview:

- Raymond, Chapter 1.
- Lab!

Raymond, Chapter 1

I've answered many of your questions on the page for this reading. Here are some synthesized questions and my answers to those questions. We'll also talk about others in the coming days.

What is an operating system?

- Traditionally, the bridge between most users of the computer (including the programmers) and the underlying resources of the computer.
- OS's manage: Files, disks, memory, processes, input (e.g., from keyboards) and output (e.g., to the screen or to a printer), security, and more.
- OS's provide not only a model of how all these things are organized, but also functions for accessing them.
- These days, we also think about operating systems in terms of their user interface: How does the

normal user interact with all of this.

Can you give an example?

- Sure. Let's consider something that's not quite an OS issue, but is close enough and is also related to things you learned in 151.
- Suppose Scheme had no built-in string type. How would you implement it?
- Three basic options:
  - Immutable lists of characters
  - Vectors of characters
  - Mutable lists of characters
- The representation you choose affects what operations are natural (e.g., can you change a character or insert some) and the efficiency of operations.
- Note that strings are a lot like files.

What's special about Unix?

- It appears that the design decisions in Unix were particularly good and adaptable.
- Certainly, the basic decisions have lasted well.

What did you see as the important points of the chapter?

- ...

Laboratory

- Yay! Even more hands-on explorations with Linux.
- Be prepared to reflect.
