CSC151.02 2014S, Class 02: An Introduction to Linux

Overview

- Preliminaries.
  - Admin.
  - Homework.
  - Questions.
- Lessons from day one.
- Common parts of an algorithm.
- About the course.
- Getting started with Linux. [1:40-2:05]

Preliminaries

Admin

- Sam forgot the attendance sheet. We’ll do attendance again on Wednesday.
  - Handing out accounts is a form of attendance. Not here: BT
- Mr. Stone will be visiting class to bring your account information.
  - Separate from the rest of campus
  - So you get a new password
  - Mr. Stone wrote one for you
  - You should change it
  - If you had one already, you don’t get one
  - If you don’t remember, go to Mr. Stone’s office
  - And you can watch him show me up on correct pronunciation of names
  - Mr. Stone is happy to talk to you about the software we have available, the wonders of open source, and the evils of the NSA
- When doing recitation, I will call on you in physical order in today’s class.
- Welcome to our guest mentors, Toby B and Alex G
- Extra credit:
  - Thursday extra on summer research in CS, Thursday @ 4:30 in Noyce 3821.
  - CS Table Friday at noon in the Day PDR in the Marketplace: 3D Printing and Cyborg Pars

Homework

- Don’t forget that the first assignment is due at 10:30 p.m. tonight. I appreciate that assignments have started arriving.
- There are lots of readings for tomorrow’s class, but they are all short.
Questions

Are there questions on or from the homework?

*Can you correct your homework, as long as you submit it before the due date?*

Certainly. But the odds are about 20% that I’ll grade the wrong assignment.

Are there questions from the readings (other than the ones that I plan to ask you)?

Lessons from day one

Review: Some things we commented about yesterday, and some followup comments from SamR.

Summary of yesterday

- Cs is the study of algorithms, algorithms are instructions.
- Yesterday’s exercise: Instructions to make nut butter and jelly.

Things we learned

- When writing instructions, know what your audience knows.
  - Part of learning to write algorithms is learning the basic set of instructions you can use.
- Be specific. If your audience can misinterpret you, it will.
  - Computers are sentient and malicious.
- When things go wrong, it’s funny, not painful (well, except for me).
  - Be positive!
- You write better algorithms if you can observe them working step by step.
  - We’ll work in a development environment in which you can play with small instructions.
- You write better algorithms when you write them with other people.
  - And so we force you to do so
- Know your data - (Screw-top vs pop top).
  - We’ll learn a new kind of information (e.g., number, string, picture) and the operations that we can use on that information
- Watch for infinite loops.
- Computers seem to intentionally misinterpret instructions.
- Keep instructions simple, if possible.
- Testing should be rigorous and thorough.

Common components of an algorithm

You did a reading about this.
What are some parts of an algorithm?

- Subroutines - A named section of code, that you can use again and again
  - How to open a jar if it’s a screw-top jar: unscrew the top by grasping the bottom with your dominant hand, the top with your non-dominant hand, placing the top toward the ceiling, and rotating your non-dominant hand counter-clockwise otherwise, if it’s a pop-top jar ... otherwise throw it against the wall and cross your fingers
- Variables - , names of things
- Similar: Parameters Inputs to subroutines
- Conditionals - If ... Might or might not do something. (Make a choice.)
  - If you grabbed the knife incorrectly, switch ends
  - If the stay-fresh seal is on top, ...
- Repetition - It’s repeating the same action again and again and again
  - “Turn non-dominant hand counter-clockwise until the lid comes off”
  - Be careful about doing repeating forever

About the course

- You probably learned a bit in the first assignment.
- You’ll learn more over the first week.

Getting started with Linux

- Yay! Learn how to use the bunsen burners and test tubes.
  - No, not really, but the same depth of learning.
  - These are the things that help you get the real work done.
- It’s sad when a CS prof can’t write good instructions, isn’t it?

Samuel A. Rebelsky, rebelsky@grinnell.edu

Copyright (c) 2007-2014 Janet Davis, Samuel A. Rebelsky, and Jerod Weinman. (Selected materials are copyright by John David Stone or Henry Walker and are used with permission.)