

Slide 1

Administrivia

- These presentations will be linked from the schedule page. Usually I'll post a preliminary version before class, a more-final version later (sometimes there are typo fixes, e.g.).
- I've also made a Google Doc with links to Zoom recordings and shared it with all of you. There's a link to it at the bottom of the schedule page too.
- Also, you may notice in your Google Drive "shared with me" a folder with name "CSCI-3215-*your_name*"? This is the shared-only-by-you-and-me folder where I plan to put information about grades and where I want you to put things you're turning in.
- I'm hoping to get the first reading quiz (over chapter 1 and 2) posted over the long weekend. (I had posted a version, but I've removed it to make some changes.)

Slide 2

More Administrivia

- During class, and when doing the reading, you'll probably want access to a UNIX-like system to try things out. Our classroom machines are the easy option and also available for remote use.
- Or if you have a Linux computer you're more or less home free, though you may need to install some optional packages.
- If you have a Mac, its Terminal application seems to be close enough, though details of some commands are different.
- If you have a Windows machine, you could try Cygwin or installing a full-blown virtual machine.

Slide 3

What I Hope You Will Get From This Class (Mostly Review)

- More things in your “bag of tricks” — shell features, shell scripts, makefiles, a text editor, etc., etc.,
(Most of what we talk about will be applicable to all UNIX systems, not just Linux.)
- Details of what you learn may not stick with you (any skill you don’t use regularly ...). What I hope will is a sense of what’s possible! and an awareness that much can be automated — which I hope will carry over into the GUI world. (Some GUIs are more scriptable than they appear, via “APIs”.)

Slide 4

A Bit of Truth in Packaging (?)

- Focus of required work will be on old-style text-mode tools (as opposed to graphical stuff), and on user-mode work (as opposed to installation / administration).
- If what really excites you isn’t something we focus on in class, though — I’m open to project ideas concerning topics we don’t cover in class.

Slide 5

Shameless Evangelism/Ranting

- “UNIX is obsolete — history goes back to 1969!”
You can fix a lot of bugs in 50+ years, and the odds are better that what you learn will still be useful years from now.
- “It’s not user-friendly!”
Old joke(?): Sure it is; it’s just choosy about its friends. Designed by programmers for programmers — “expert-friendly” as opposed to “novice-friendly.” Particularly good if you want to automate something.
- “Everyone knows GUIs are better!”
For some things and some people, maybe so. But which is more expressive, pointing and gesturing or speech? (With apologies to former colleague Dr. Howland, whom I’m paraphrasing.)
- (You don’t have to agree with me; listen and decide for yourself!)

Slide 6

The UNIX Philosophy

- As stated by one of its developers (Doug McIlroy):
“Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.”
(That last point isn’t as true these days as it once was, though maybe, and the general idea of many small tools with well-defined ways of interacting still has merit?)
- There’s more, but the emphasis is on (1) providing a set of lightweight tools that can be put together to do interesting things, and (2) providing choices to users (sometimes almost too many!).

Slide 7

Reading The Fine Manuals

- In times past, testy local experts sometimes responded to questions with “RTFM” (Read The Fine Manual). Well, it’s a useful skill, though it’s losing ground to STFW (Search The Fine Web).
- One of the most useful things you can learn is how to learn more. Documentation on UNIX systems not always perfect, and not particularly novice-friendly, but usually thorough, and traditionally kept in local files.
- Local documentation:
 - `man` pages.
 - `info` pages.
 - Elsewhere on the system. `locate` on Linux may help.

Slide 8

RTFM — `man` pages

- Reference documentation (as opposed to tutorials). Provided for most commands and also for C and some other library functions.
- Organized into “sections” (user commands, sysadmin commands, library functions, etc.). Can have entry with same name in multiple sections. `-a` option or section number gives access to non-default.
- Of particular interest is the section `SEE ALSO` — sometimes lists other related commands.
- `man -k` (or `apropos`) to search for command names.
- Try `man man...`
- Now you might want to know about `more`, or `less`.

Slide 9

RTFM — `info` pages

- Also reference documentation, sometimes more current / complete than `man` pages. (Why two systems? Probably historical reasons!)
- Organized in a way somewhat similar to hypertext.
- Try `info info...`

Slide 10

Other Useful Info-Gathering Commands

- `locate` (fast file-search by name, Linux but maybe not other UNIX).
- `whereis` (finds files associated with a command).
- `file` (tries to identify file contents).
- `which` (looks for command name in search path).

Minute Essay

- What platform(s) do you think you'll use for homeworks and/or experimentation?
- Anything today particularly unclear, or that you want to hear more about?

Slide 11