Slide 1

Administrivia

- Reminder (as if you needed one!): Project presentations December 11. "Deliverables" also due. For anything code-like (scripts, macros, etc.), send via e-mail. For reports, hardcopy preferred.
- Homework 9 due today. (Accepted without penalty through day of project presentations. Just don't forget!)
- Grade summaries sent by e-mail. (Apologies for the long delay in grading Homework 8! Maybe I tried to do too good a job?)
- Final deadline for turning in missed homeworks the day after project presentations at 11:59pm.
- Office hours as announced in e-mail: "Open lab" 12/05; office hours 12/10 and 12/11. Also e-mail is usually a good way to reach me.

Slide 2

Course Wrap-Up — What I Hope You Got From This Class (Details)

- More things in your "bag of tricks" (see later slide).
- Practice in reading man pages and otherwise learning more.
- Exposure to traditional tools you might need, or want, to use sometime. You might not remember details, but I hope you now have a better sense of what's available and can (re)learn if/when you need to. (Also exposure reduces the intimidation factor?)

Slide 3

Course Wrap-Up — What I Hope You Got From This Class (Big Picture)

- Exposure to a different operating system / user interface paradigm — many small programs that work together, information kept in text files, emphasis on being expert-friendly and scriptable, etc.

  "A tour of UNIXworld / TextWorld."

  "More than one way to do things."

- Encouragement to find out how to use all your tools as intelligently as possible.

  I like the old tools because I know how to make them work together. But it's worth noting that many "modern" tools (GUI-based programs, graphical file managers, etc.) have their own way of working together — common set of keybindings, cut-and-paste metaphor, drag-and-drop, multiple selections, etc. Compare and contrast!

Slide 4

Course Wrap-Up — Topics

- Shell features — command history, redirecting input and output, scripting features (if/then/else and loops).
- Pipes.
- Filter programs (`awk`, `sed`, `grep`, etc.).
- Text editors.
- LaTeX.
- `make` and makefiles.
- Regular expressions (for text editing, `grep`, etc.).
- Miscellaneous other text-mode tools — mail, `screen`, etc., etc.
- A little about installing software, Linux filesystem conventions, CGI scripting, and Perl.

Slide 5

From First Lecture — Shameless Evangelism/Ranting

- "UNIX is obsolete — history goes back to 1969!"

  You can fix a lot of bugs in 40+ years, and the odds are better that what you learn will still be useful years from now.

- "It's not user-friendly!"

  Sure it is; it's just choosy about its friends. Designed by programmers for programmers — "expert-friendly" as opposed to "novice-friendly."

- "Everyone knows GUIs are better!"

  For some things and some people, maybe so. But which is more expressive, pointing and gesturing or speech?

- (You don't have to agree with me! but in theory now you have more information on which to base an opinion.)

Slide 6

From First Lecture — 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."

- 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

Minute Essay

- None really — just sign in (unless you have parting remarks?).
- And best wishes for a successful end of semester and a good holiday!