

Administrivia

Slide 1

- For the record: `ncurses` examples updated.
- Reminder (as if you needed one!) Project presentations Friday. “Deliverables” also due. For anything code-like (scripts, macros, etc.), send via e-mail. For reports, hardcopy preferred.
- Reminder: Homework 8 due today. Hardcopy preferred.
- (Status of grading, grade information.)

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

Slide 2

- 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?)

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

Slide 3

- 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!

Course Wrap-Up — Topics

Slide 4

- 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, 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- “UNIX is obsolete — history goes back to 1969!”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 on 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!)

Minute Essay

- None really — just sign in (unless you have parting remarks?).

Slide 7