

Slide 1

Administrivia

- Reminders: Homework 6 (L^AT_EX) due today. Homework 7 (make) due Wednesday.

Slide 2

Minute Essay From Last Lecture

- “What do you use for reading mail?”
Lots of votes for Outlook. Several mentioned calendar. One mentioned multiple folders (most mail programs support this).
Mentions of OS X Mail, Mozilla, Thunderbird, etc. Also one mention each of mutt and pine.
A comment I appreciated — one person dislikes all tools he’s tried — doesn’t understand how they work and that leads to feeling uneasy.

“What Command Do I Use To ...”

Slide 3

- You know about `apropos` as a way to discover new commands. You probably also know that it's not perfect.
- So today (and next time), a tour of some commands I have found useful ...
- (The point of this tour is not to present details of any of the commands, just to make you aware they exist, so you can follow up on those that seem useful.)

Commands for Working With Text and Other Data

Slide 4

- `script` to capture all terminal input/output. (`exit` to stop capturing.)
Not as useful as it might be because you also get stuff to control terminal, make colors, etc., but could be a good approach if you need to capture both input and output.
- `tee` to capture stuff to a file and also show on standard output. Example:

```
gcc gccparameters ... 2>&1 | tee OUTSAVE
```
- `strings` to search a file for printable strings.
Can be useful as a quick-and-dirty (i.e., not necessarily 100% reliable) way of scanning non-text files (e.g., files in MS Office formats) for printable text.

Commands for Working With Text and Other Data, Continued

Slide 5

- `grep` to search a file or files for specific text or regular expression. GNU `grep` has `--recursive` flag for searching subdirectories too.
- `diff` to compare two files. Can also generate output in formats useful for creating “patches” and editing scripts.
Also try `vimdiff` (possibly with `-o` option).
- `ispell` or `aspell` to check/correct spelling. (No man page on our machines. `locate` to find documentation.)
- `od` to show data in various forms (binary, hexadecimal, etc.). Useful for finding out exactly what’s in a non-text file. Examples:
`od -t c textfile` to show characters including line-end and other control characters.
`od -t x1 somefile` to show data a byte at a time in hexadecimal

format.

Slide 6

Commands for Printing

- `lpr` to print PostScript or text. (Also seems to print PDF on our machines.)
`lpq` to check print queue; `lprm` to cancel a print job.
- `enscript` or `pr` to pretty-print text. Options allow printing in landscape mode, with different font sizes, in multi-column format, etc.

Slide 7

Commands to Compress and Archive Data

- `gzip` and `gunzip` to compress/uncompress data. Or use `compress` and `uncompress`.
- `tar` to create Unix-standard-format “archive” file, a.k.a. “tarball”. (Conceptually similar to ZIP archive files — which you can generate, using `zip`.)

A sometimes useful sequence (copies a directory, preserving any symbolic links):

```
(cd sourceDir; tar cf - . ) | ( cd target; tar xf - )
```

Slide 8

Web-Related Commands

- `wget` or `curl` to download a Web page or pages. `wget` has options to allow downloading a page and everything it references.
- `lynx`, `elinks` (a.k.a. `links`), or `w3m` to browse in text mode.
(What's the difference? Slightly different capabilities. Better help (IMO) in `lynx`, but it doesn't do frames. `elinks` does frames but help isn't (IMO) as useful. I haven't tried `w3m`!)

Slide 9

Miscellaneous Fun (?) Stuff

- Some selected text-based games installed in `/users/cs3190u/games/games`. "Caveat user" — installation was quick-and-dirty, so not all games work 100%, but interesting as a sampling of what games were like in the (not-always-so-)Good Old Days.
- And of course there's
`telnet towel.blinkenlights.nl`
(If you try this — just type the above, and be patient. Use control-] and then "quit" to interrupt it if you need to.)

Slide 10

Minute Essay

- None — sign in.

Slide 11