Administrivia

- Reminder: Homework 7 due today.

- Sample solutions for Homework 3 and Homework 4 available online.

- Summary of grades sent by e-mail today.

- Probably no more than two more homeworks, plus project. More about the project coming soon.
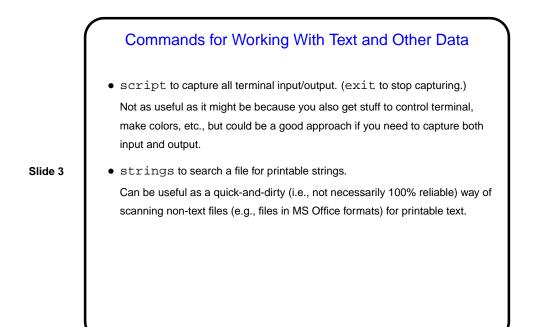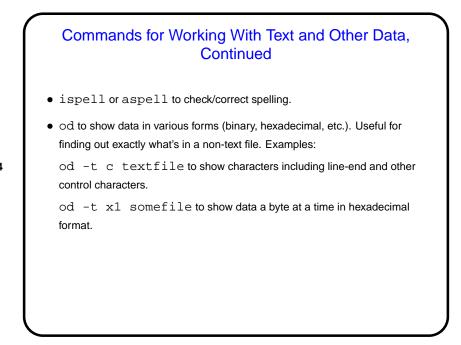
Slide 1

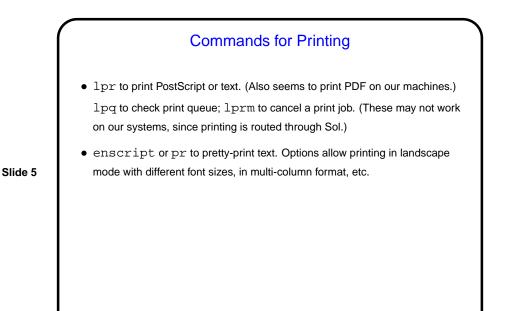## "What Command Do I Use To . . . "

- 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.)

Slide 2

Slide 3

Commands for Working With Text and Other Data

- `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.

- `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.

Slide 4

Commands for Working With Text and Other Data, Continued

- `ispell` or `aspell` to check/correct spelling.

- `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.

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. (These may not work on our systems, since printing is routed through Sol.)

- `enscript` or `pr` to pretty-print text. Options allow printing in landscape mode with different font sizes, in multi-column format, etc.

Slide 5

Commands to Compress and Archive Data

- `gzip` and `gunzip` to compress/uncompress data. Or try `compress` and `uncompress` (not available on our Linux machines, but found on many UNIX systems).

- `tar` to create UNIX-standard-format "archive" file, a.k.a. "tarball". (Conceptually similar to ZIP archive files — which you can generate, using `zip`.)

  Another way to copy a directory, preserving symbolic links:

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

Slide 6

Slide 7

Text-Mode Calculators ($\texttt{bc}$ and $\texttt{dc}$)

- Useful in that both support arbitrary precision. (So, if you want to know *exactly* what 2^{100} is ...)

- I sometimes use from within $\texttt{vim}$, for quick calculations.

Slide 8

Commands for Accessing Other Machines

- $\texttt{ssh}$ to remotely log in / run commands.

  $\texttt{-X}$ flag allows running X-based (GUI) programs.

  $\texttt{ssh user@machine}$ logs in as a (possibly) different user.

  $\texttt{ssh user@machine "command"}$ to execute single command (or commands). *Note* that this may bypass some of normal shell setup (e.g., reading $\texttt{.bash\_profile}$.

  Can set up so it doesn't prompt for a password. Link to instructions on "Useful links" page.

- $\texttt{rsh}$ and $\texttt{telnet}$ to provide similar functionality, but less securely. Often turned off by sysadmins for that reason.

Slide 9

Commands for Copying Files Between Machines

- `scp` to copy file(s) between machines. If you set up `ssh` to not prompt for password, applies to this too.

- `rsync` to "synchronize" a target file/directory with a source. Useful in maintaining backup. Example:

```
rsync -avz --delete /users/yourName/ \
      /directory-for-backup
```

  Precede directory/file with `user@machine:` to copy to/from remote machine. `-e ssh` may be needed in order to use SSH rather than RSH.

Slide 10

Commands for Working with Programs

- `-E` (show preprocessor output) and `-S` (generate assembly-language output) flags on most compilers.

- `gdb` source-level debugger. Semi-graphical version available from within `emacs`.

Commands for "Batch" Programs

Slide 11

- `batch` and `at` to run something "in batch mode" / at specified time. If output is not redirected, it's sent to you by e-mail.

- `crontab` to set up "cron job" to execute periodically. Not necessarily restricted to root user.

- `nohup` to run a command in a mode in which it keeps running even when the launching terminal session goes away. Caveat: Command can't make use of terminal input/output (so all input/output must be routed to a file, a pipe, etc.). Or run using `screen` (later).

Web-Related Commands

Slide 12

- `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`!)

Minute Essay

- Any favorite tools that seem to fit with this lecture, but that I forgot to mention?

- Any "how do I do this?" questions that also fit but that I didn't talk about?
  Other topics I plan to talk a little about are e-mail and X/GUI stuff.

Slide 13