Slide 1

Administrivia

- Reminder: Homework 7 due Wednesday.

Slide 2

“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 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.

Slide 5

Commands for Printing

- `lpr` to print PostScript, PDF, or text. Add `-P` and a printer name to specify the printer (e.g., `lpr -Pstylus foo.pdf`).

  `lpq` to check print queue; `lprm` to cancel a print job (`-P` to specify a printer here too. If these don't seem to work on our systems, try them on Sol, which does the actual printing.)
- `enscript` or `pr` to pretty-print text. Options allow printing in landscape mode with different font sizes, in multi-column format, etc.

Slide 6

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 7

Text-Mode Calculators (bc and dc)

- Useful in that both support arbitrary precision. (So, if you want to know *exactly* what 2^{100} is ...)
- I sometimes use from within vim, for quick calculations.

Slide 8

Commands for Accessing Other Machines

- ssh to remotely log in / run commands.

  -Y flag allows running X-based (GUI) programs. (Also -X, but that may be less secure.)

  ssh user@machine logs in as a (possibly) different user.

  ssh user@machine "command" to execute single command (or commands). *Note* that this may bypass some of normal shell setup (e.g., reading .bash_profile.

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

Minute Essay

- What do you currently use to produce formatted documents? What do you like/dislike about it?