

Slide 1

Administrivia

- Reminder: Homework 8 due today. Turn in by e-mail.
- I plan one more regular homework, a sort of "reflect on the course" near the end of the semester.
- And then there are the projects mentioned in the syllabus . . . I've posted a description of requirements on the course Web site. (Briefly review.)

Slide 2

Minute Essay From Last Lecture

- Many people thought they might find `screen` useful. (A few already do!)
- A few people mentioned possible uses for `at` and `cron` jobs. For the latter, one said the script I use to find `*swp` files sounded useful, and also that something to periodically do a `git commit` might help his workflow.

Makefile Correction

- I discovered recently, to my chagrin, that I was misusing one of the variables used by GNU `make`'s implicit rules:
- `LDFLAGS` is meant not for a list of extra libraries to list — that's `LDLIBS` — but other link-related options such as `-L`.

Slide 3

“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, 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 4

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 can be useful if you need to capture both input and output and are willing to do some editing after the fact.

Slide 5

- `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, PDF, or text. Add `-P` and a printer name to specify the printer (e.g., `lpr -Pportia 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

Slide 7

- `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 - )
```

Text-Mode Calculators (`bc` and `dc`)

Slide 8

- 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.

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.
- `sftp` to initiate FTP session to remote machine. Can move around in directory hierarchy and copy files. I also find it useful for copying not-globally-readable from one user to another. Recent versions appear to (finally!) support tabbed filename completion.

Slide 10

Command for “Synchronizing” Directories — `rsync`

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

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

Slide 11

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. Optional filtering allows fairly precise control of what is copied — *but* details, and getting them right, are tricky.

- **CAUTION:** I recommend always running first with “dry run” flag (`-n`) to be sure what `rsync` will do is what you want!

Commands for Working with Programs

- `-E` (show preprocessor output) and `-S` (generate assembly-language output) flags on most compilers (`gcc`, `g++`, etc.).
- `gdb` source-level debugger. Semi-graphical version available from within `emacs`.

Slide 12

Web-Related Commands

Slide 13

- `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. Also doesn't seem to work well on this year's build. `w3m` looks promising as an alternative.)
(Advantage of these commands is that they're lightweight and ignore some things that make graphical Web browsing annoying. Disadvantage is also that they ignore some things ...)

Miscellaneous Other Command(s)

Slide 14

- `time` to run a command and say how long it took. (Actually there's often more than one thing by this name, e.g., a shell built-in and also a command. Access the latter with a full path name or by preceding the name with a backslash (in `bash` anyway).)
- `top` monitors performance in realtime.
- `stdbuf` overrides usual buffering of `stdin`, `stdout`, `stderr`. Appears to be useful when capturing output of a program that crashes, causing some of what goes to `stdout` to not be written out.

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?
(There was a request to talk more about `awk` and/or `sed`. I'll try to do that soon.)

Slide 15