

Slide 1

Administrivia

- Late policy changed, as per e-mail.
- Slides from last time updated to include a few more examples and comments. Slides from class on regular expressions updated to mention version differences (Linux versus OS X). (As a general rule / in my experience — versions of UNIX are all pretty similar in functionality, but not identical in all details.)
- Information about projects coming next week. (Briefly — almost anything that (1) uses something we've talked about and (2) I can reasonably supervise/grade.) There will be a few more short homeworks as well.

Slide 2

“What Command Do I Use To ...”, Continued

- Continuing from last 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.)
(In fact, that's really the point of this course in general!)

Commands for Working with Numbers

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

(I sometimes find this useful in combination with `vim`'s ability to execute commands from within the editor.)

Slide 3

- `gnuplot` to generate plots. Nice for \LaTeX users because it can produce output in various \LaTeX -friendly formats (including ones that allow final typesetting to use same fonts as document). Help available from within program by typing `help`.

Commands for "Batch" Programs

- `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.

Slide 4

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 `xemacs`.

Slide 5

Commands for Accessing Other Machines

- `ssh` to remotely log in / run commands.
 - `-X` flag allows running X-based (GUI) programs.
 - `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 with less security. Often turned off by sysadmins for that reason.

Slide 6

Commands for Copying Files

- `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 7

A Text-Based Window Manager

- `screen` is ... a “virtual virtual terminal”, a “text-based window manager”, something that multiplexes a physical terminal between several processes, usually interactive shells.
- Supports one or more “windows” (programs, usually shells), plus one or more “regions” (areas on screen).
- Functionality includes
 - Ability to leave programs running even if “real” terminal isn’t there — i.e., disconnect/reconnect.
 - Ability to copy and paste text among windows, log stuff, etc..

Slide 8

Slide 9

A Text-Based Window Manager, Continued

- You know what to do to find out more ... (Or try <http://www.gnu.org/software/screen.>)
- Some useful info for getting started:
 - `screen` starts things up.
 - `control-A c` creates a new window. `exit` to end. `control-A "` gives menu for selecting window.
 - `control-A S` creates a new region ("splits the screen").
`control-A X` to end. `control-A Tab` switches to next region.
 - `control-A ?` shows key bindings.
 - `control-A D` detaches session. `screen -r` to resume.

Slide 10

Miscellaneous Other Command(s)

- `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 output must be routed to a file, a pipe, etc.).
- `time` to run a command and say how long it took.
- `urlview` to (try to) find URLs in text and display text-based menu for accessing them.

More Miscellaneous Fun (?) Stuff

- One more text-based game — RobotFindsKitten
(<http://www.robotfindskitten.org/>, now installed in /users/cs3190u/games).

Slide 11

Minute Essay

- Any favorite tools that seem to fit with this week's lectures, but that I forgot to mention?
- Any "how do I do this?" questions that also fit but that I didn't talk about? (I'll talk just a little about X/GUI stuff next time. I might talk just a little about sysadmin stuff in a future lecture.)

Slide 12