Slide 1

Administrivia

- Sample solutions to Homework 5 on Web.
- Sample solutions to Homework 6 available, but — only two people have turned something in?! and only one person has turned in Homework 7?!
- One more short homework, due the last day of class. Probably best not to start now (!).
- Information about projects on the Web (at last!). Proposals due a week from Monday; everything else the day of the final.
- (E-mail about class before holiday sent . . .)

Slide 2

Minute Essay From Last Lecture

- “Any downsides to running multiple X sessions on the same machine?”

Slide 3

Installing and Updating Software — Packages

- “Modern” way to package software for installation is as a “package” — `.rpm` files (originated with RedHat), `.deb` files (originated with Debian).
- Key idea is that packages bundle up files, installation scripts, and dependency information, and interact with a database representing what else is installed.
- Low-level tools for installing and querying individual packages exist — e.g., `rpm` command. (Can also use to create your own packages.)

Slide 4

Installing and Updating Software — Package Managers

- Still more convenient/recent — “package manager” that uses the lower-level tool(s) and also provides a way to download needed packages from one or more “repositories”.
- If installing in “normal” system directories, and as root, probably best to take this approach.
- If you want to install in other directories (e.g., your home directory), or you don’t have root access, some packages allow that, or you can (probably?) unpackage it. Or there’s the traditional UNIX approach . . .

Slide 5

Installing and Updating Software — “Tarballs”

- Traditionally, UNIX software distributed in the form of a “tarball” (archive created by `tar`, possibly compressed, usually containing source). Still often available and useful — e.g., to install in your home directory.
- What do you do with a tarball? Typical installation goes like this . . .

Slide 6

Installing and Updating Software — Installation from “Tarball”

- “Untar” the file (`tar xf`). Usually creates a directory, often containing `README` and/or `INSTALL` files — which you should review.
- Run `configure` script to set system-specific options. Usually figures most things out for itself, but may need/allow user input, either via command-line options or standard input.
- Run `make` to compile, etc. Normally puts created files in the same directory.
- Run `make install` to move/copy executables, etc., to system directories. Notice that this is the only step that requires root privileges — and only if installing in system directories.

Slide 7

Minute Essay

- Do you have experience with any of the things discussed today? (What?)