

Slide 1

### Administrivia

- Project proposals were due Wednesday. Due date moved to next Monday.

Slide 2

### Minute Essay From Last Lecture

- Most people use TMail most of the time. Pluses mentioned are the interface (which I find much too busy!) and how it integrates with other services (admittedly nice).
- A few had tried `mutt`. One had used `alpine` (for Dr. Zhang). And there were mentions of graphical mail clients (Thunderbird, mailspring, Outlook).

### Installing and Updating Software — Packages

Slide 3

- “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.  
(Not a new idea — existed in the mainframe world decades ago.)
- Low-level tools for installing and querying individual packages exist — e.g., `rpm` command. (Can also use to create your own packages.)

### Installing and Updating Software — Package Managers

Slide 4

- 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 . . .

### 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 5

### Installing and Updating Software — Installation from “Tarball”

- “Untar” the file (`tar xf`). (May need to add a flag to uncompress. Several choices.) 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. (This is where you typically say where you want to install, via `--prefix`.)
- Run `make` to compile, etc. Normally puts created files in the same directory.
- Run `make install` to move/copy executables, etc., to system directories. Note that this is the only step that requires root privileges — and only if installing in system directories.

Slide 6

### Minute Essay

- Do you have experience with any of the things discussed today, or other ways of installing software on UNIX-like systems? (What?)

Slide 7