Slide 1

Administrivia

- (None?)

Slide 2

Installing and Updating Software — Packages

- "Modern" way to package software for installation is as a "package". Two major forms: `.rpm` files (originated with RedHat), `.deb` files (originated with Debian). Some distributions may use other forms.
- 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 — something similar 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.)

Slide 3

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" (standard sources — a distribution will often have its own, possibly more than one, and system administrator can add others).
- 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 4

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` — for files ending in `.tar`). (May need to add a flag to uncompress. Several choices. E.g., Filename ending in `.tgz` uses old-style compression; untar with `tar xzf`.) Usually creates a directory, often containing `README` and/or `INSTALL` files — which you should review.
- Run `configure` script to set system-specific options. `configure --help` will (usually?) list them. 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`.)

  Builds makefile(s).

Slide 6

Installing and Updating Software — Installation from "Tarball", Continued

- Run `make` to compile, etc. Normally puts created files in the same directory. Optionally, run `make check` (if available) to do some testing. Some errors considered normal.
- Run `make install` to move/copy executables, etc., to system directories. Note — only step that requires root privileges, and only if installing in system directories.

Slide 7

Digression: Text Editors Revisited

- Some text editors (`vim` among them) allow you to "filter" text through an external program.
- One thing this allows is building on-the-fly scripts — construct in `vim` the lines to execute, then execute them with, e.g., `:%!sh`. (No need to save unless you want to reuse another time.)

Slide 8

On-the-fly Scripts, Continued

- I like this "on-the-fly scripting" for various kinds of file moving/renaming operations — use `r!ls` to get a list of files, "massage" with various editing operations, then execute as above. I find this works well as a way of dealing with filenames containing spaces — relatively easy to add double quotes around names. A useful idiom employs a simple regex and `&` to reference the matched text, e.g.,

  ```
  :%s/.*/mv -v "&" targetdir/
  ```

- (Of course I could also use a `bash` loop, and sometimes I do, but — whatever seems easiest for the particular use case?)

Slide 9

Text Editors Revisited, Continued

- I also use `vim`'s ability to record and play back "macros" fairly regularly. To do this: Start recording with `q` plus a single letter. End with another `q`. Play back with `@` and the single letter.

  (Somewhere sometime I think I remember a comment to the effect that with regard to certain repetitive tasks there were two kinds of people — the ones who write macros and the ones who write a regular expression. I do both, depending on the situation.)
- It can be tricky to record in a way that will "play back" effectively, but when this works, it works well.
- I use this sometimes when I need to make the same edit to several files.

Slide 10

Minute Essay

- Do you have experience with any of the installation tools just discussed? Or have you used other ways of installing software on UNIX-like systems? (What?)