Slide 1

Administrivia

- Reminder: Homework 6 due today.
- Homework 7 on the Web; due next Wednesday.

Slide 2

What are TEX and LATEX?

- TEX — program for typesetting mathematics, developed by Knuth (1978) for his book *The Art of Computer Programming* and made freely available.
- LATEX — extensive set of macros for TEX written by Lamport (1985), that provide functionality needed for scholarly papers. Extended over the years by many people.
- These are “text formatters” not “word processors”, and as such don’t include a built-in editor. (But in this modern world, there are IDE-like programs for working with them, as mentioned later.)
- Basic idea — you write “source code” for your document (text and markup) with a text editor, then use TEX or LATEX to turn it into a formatted document.
- Both available in zero-cost form for many platforms. Included in complete Linux distributions (as far as I know).

Slide 3

Basics (Under UNIX)

- You write "source" (`foo.tex`) with a text editor of your choice. It includes your text plus "logical markup" — e.g., `\section{A Section Heading}`.

  (What about checking spelling? Use a separate tool — "each program should do one thing, and do it well." `ispell` and `aspell` are common ones.)

- You use the command `latex` to generate a `.dvi` file, then `dvips` to generate PostScript, then (if desired) convert to PDF with `ps2pdf`.

  (You can also go directly to PDF with `pdflatex`.)

Slide 4

Isn't That a Lot of Trouble?

- In some ways, yes — there is a learning curve, and there are many "gotchas".
- For some jobs (where visual layout matters more than logical structure), LaTeX is probably the wrong tool.
- But if you persevere . . .

Slide 5

Why It Might Be Worth the Trouble

- Output looks good — math in particular.
- Logical structure of document is clearly spelled out. (You can sort of do this with, e.g., MS Word, but it's less transparent.)
- Cross-referencing, bibliographic references, footnotes, tables of contents, indexing, etc., "just works".
- Documents are stable — only way to "corrupt" a document is to mess up with your text editor. Very old documents usually still compile, and if they don't the content is still accessible.
- Once you figure out how to do a particular trick, it's there in the `.tex` source for future reference.

Slide 6

Basics, Continued

- LaTeX provides a small set of "document classes" — article, report, book, etc. These classes group definitions for section headers, lists, etc., in a way that everything looks good together. Also can have "packages" that group together related customizations, provide extra features.
- Basic document structure (look at example):
  - `\documentclass[options]{foo}`
  - Additional global definitions, packages, etc.
  - `\begin{document}`
  - Your text. "Paragraphs" continue until first blank line.
  - `\end{document}`

Slide 7

Some Features

- "Sectioning commands" provide consistent layout and automatic numbering. Also allows collecting info to make table of contents.
- "Environments" provide support for lists (bulleted and numbered), tables, centered text, "verbatim" (equivalent of HTML preformatted text), etc.
- Macros provide simple markup, e.g., `\textit{foo}`.
- Math — a bit cryptic, but IMO not worse than point-and-click equation editor. Support for (automatically) numbered equations.
- Graphics in EPS (Encapsulated PostScript) form can be included, and scaled nicely. I use `xfig` to draw pictures — old, but nice integration with LaTeX. There are other tools.

  (Notice — EPS is the traditional format and works with the traditional source-to-DVI-to-PostScript toolchain. `pdflatex`, however, allows most currently-popular image formats, but *not* EPS.)

Slide 8

More Features

- Figures and tables can "float" (LaTeX will put them where they fit). Also footnotes.
- Lots of cross-referencing features — declare symbolic label (for section, figure, etc.) with `\label{foo}`, reference with `\ref{foo}`.
- Support for bibliography / list of references — usually use companion package BibTeX.
- Support for indexes. (Also glossaries, through add-on packages.)
- Facilities to define your own "commands" and "environments". Makes it easy to get consistent formatting; also can provide convenient shorthand ways of doing things.

More Features / Add-Ons / Tools

- Tools to convert LaTeX source to HTML. (I use `latex2html`; there are others.)
- Document classes for producing “slides”. (I use `seminar`; there are others.)
- Tools for editing LaTeX source. Support in both `emacs` and `vim` (`auctex` and `vimlatex` respectively). Also GUI frontends. See “useful links” page.

  (*Note:* Our current Fedora systems have installed a `vim` plugin for editing LaTeX source. Looks interesting but can be annoying. Disable by putting `filetype plugin off` in `.vimrc` file.)

Gotchas

- Some characters have special meaning and must be “escaped”: backslash, brackets, #, %, $<$, $>$, $|$, caret (ˆ), underscore (_), tilde (˜).
- Quotation marks should be entered as ` ``foo'' `. Dashes should be entered as `--` (“en dash”, suitable for connecting numbers, e.g., 1–100) or `---` (“em dash” — between words).

Advice For Getting Started

- Get hold of an example that looks somewhat similar to what you want to produce, plus some sort of documentation — a guide from online or a book.
- Tinker with the example, putting in your prose and other stuff.
- When something doesn't work, ask a local expert.

Slide 11

Minute Essay

- None — sign in.

Slide 12