A Brief Introduction to LaTeX

Marie desJardins
CMSC 691B, Spring 2004

January 23, 2004

LaTeX is a text processing system that uses an embedded command language in plain text to generate a formatted document. LaTeX has excellent support for mathematical formulas and references, and is usually easy to use for people who are comfortable with programming languages. LaTeX is *not* a WYSIWYG ("What You See Is What You Get") word processor. However, it is (in my opinion) much more powerful and flexible than most WYSIWYG systems, including Microsoft Word. It is also the standard word processing system for publishing in most computer science conferences and journals. Many conferences provide only latex templates for producing formatted papers. LaTeX also works in conjunction with BibTeX, which automatically selects and formats bibliographic references, and produces in-line citations from your own database file of references. (We'll get to BibTeX a bit later in the semester.)

I will not require that you use LaTeX for your assignments in 691B; however, I **highly** recommend that you do so. You will be glad you learned LaTeX later.

1 Running LaTeX

LaTeX runs (at least) on the cs and gl linux machines. The easiest way to create a latex file is to use emacs (or any other text editor). There is a very simple sample file on the class website, in the file `sample.tex`. I've also put this file in my home directory on the gl machines, `~mariedj/sample.tex`. Here's how you produce a typeset document from this file (after copying it into your own directory).

```
% latex sample
% xdvi sample
% dvips -P pdf -G0 -t letter -o sample.ps sample.dvi
% ghostview sample.ps
% lpr sample.ps
```

The command "latex" processes the input `.tex` file, producing an output `.dvi` file. You can view this file on your screen using `xdvi`. If it looks the way you want it to, you can convert it to a PostScript file using the `dvips` command, with the switches shown in the script above. Now take a look at the PostScript file using `ghostview`. If it looks the way you want it, print it out using `lpr`.

You can also use `pdflatex` to directly produce a PDF file from the latex source:

```
% pdflatex sample
% acroread sample.pdf
```

The most difficult part of LaTeX is getting it to include figures properly. There are several packages: `psfig`, which includes postscript files, `epsfig`, which includes encapsulated postscript, and `graphics` and `graphicx`, which includes several file formats, including encapsulated postscript and jpg. I used to always use psfig; I've now mostly switched to using graphicx, but that only seems to work with pdflatex and not with regular latex.

2 Example Files

Here's what the `sample.tex` file looks like:

```
\documentclass{article}

\title{Title of My Document}
\author{My Name Goes Here}

\begin{document}
\maketitle

Hello, world!

{\em Hello, world!}

{\bf Hello, world!}

{\Large \bf Hello, world!!!}
\end{document}
```

The `\documentclass{article}` command on the first line of the file tells LaTeX that this is in fact a LaTeX document, of class "article." (There are other document classes, such as `report` and `book`, but typically you'll use the article class.)

The lines after the `\documentclass` command and before the beginning of the document are called the *preamble.* The preamble includes any initialization commands and general specifications for the document style. In this file, the preamble just contains the title and author commands, on the next two lines. You can also specify, in the preamble, the date you want to appear on the document, using the command `\date{Your Preferred Date}`, or leave the date blank using `\date{}`; if no date is specified, LaTeX will use today's date.

Now the body of the document starts; this is signaled to LaTeX by the `\begin{document}` command. The first command within the document body is `\maketitle`, which uses the title and author defined in the preamble to create a title section in the output file.

After the title is the rest of the document: in this case, four paragraphs (which are delineated by blank lines), each greeting the world in a different style: normal, emphasized (i.e., italic), boldface, and large boldface. Note that the font-changing commands and text to be changed are enclosed in curly braces `{}`; these delineate the scope of the font-changing commands.

The last line of the file, `\end{document}`, tells LaTeX that the body of the document is complete.

That's it!

This file (that is, the one you're reading now) is also on the website, in `latex.tex`.

3 Resources

The standard LaTeX reference book is *LaTeX: A Document Preparation System, 2/e*, Leslie Lamport, Addison-Wesley, 1994, ISBN 0-201-52983-1.

Here are several useful websites. I will post these on the course wiki.

- LaTeX Project home page:
  `http://www.latex-project.org/`
  - LaTeX Project FAQ:
    `http://www.tex.ac.uk/cgi-bin/texfaq2html?introduction=yes`
- CTAN: the Comprehensive TeX Archive Network:
  `http://www.ctan.org/`
- Peter Flynn's "Beginner's LaTeX" guide to basic LaTeX:
  `http://www.silmaril.ie/downloads/documents/beginlatex.pdf`
- The AMS "Short Math Guide for LaTeX," which includes every mathematical symbol you ever wanted to produce, and more:
  `http://www.ams.org/tex/short-math-guide.html`
- "Simplified LaTeX," a beginner's guide with a nice tutorial section at the beginning:
  `http://www.ctan.org/tex-archive/info/simplified-latex/`
- In order to run LaTeX on your own computer, you will need to install TeX and LaTeX. I use MikTeX on my laptop (under Windows XP): `http://www.miktex.org/` . You'll also need a TeX shell for actually running the latex program in Windows; I use ¡a href="http://www.projectory.de/texshell/index.html"¿WinTeXShell32¡/a¿. Mostly, though, I find it infinitely easier to just use Linux or UNIX to run LaTeX (just as with most things...).