Basic LaTeXTutorial

Terry Vaskor

September 8, 2003

1 Introduction

This tutorial is designed to teach the basics of LaTeX, in particular for use in preparing online submissions for courses such as CS 251. There is a wealth of information for those wishing to learn LaTeX more in-depth; a nice instructional introduction can be found at http://www.maths.tcd.ie/ dwilkins/LaTeXPrimer/ and a handy quick reference is available at http://www.giss.nasa.gov/latex/ltx-2.html.

Commands in LaTeX begin with a backslash (\). The first command in any LaTeX document is the `\documentclass` command. This specifies what style of document is being used, which will then be used by LaTeX to determine formatting styles for the document. For the purposes of assignments, this will generally be specified as `\documentclass{article}`.

The rest of the *preamble* generally consists of other commands that will globally affect the document. One of the more important ones is `\usepackage`. This specifies groups of commands that are to be made available to the rest of the file, in a fashion similar to Java's *import* command. For example, the statement `\usepackage{graphicx}` will allow, among other things, the use of commands that include images in the document. This section can also use redefinitions of commands; for further details, see section **??**.

The main body of the document is enlosed within the `\begin{document}` and `\end{document}` commands. It is here that all of the information to be displayed in the final produced document is included.

To make any points in your source document more clear, the percent sign (%) will cause LaTeX to treat any information on the rest of the line as a comment. Comments always specify the rest of the line following %, and do not span multiple lines.

As well, LaTeX ignores most of the white space in your source file, using its own parameters to determine the amount of white space in the output document. To specify a new paragraph, two consecutive newline characters are used; ie, leave a blank line between two segments of text to force them into two separate paragraphs. Any extra line breaks can be specified by two consecutive backslashes (\\).

2 Primary document formatting considerations

Firstly, you will want your name on your submission. LaTeX already includes a command called `\maketitle` to accomplish this. It is specified by `\title`, `\author`, and `\date`. If no date is specified, the date the document is generated will be used. When convenient, the meanings of the three fields can be abused. For example,

```
\title{CS 251, Winter 2003, Assignment 1}
\author{Due January 31, 12 Noon}
\date{30 points}
\maketitle
```

will produce the heading style commonly seen on assignments. You will want the title to specify which assignment is being submitted, and author to have your name and UNIX user ID.

When answering actual questions, it would be prudent to take advantage of LaTeX's built in numbering system. You can set up automatic numbering by placing `\begin{enumerate}` after `\maketitle` and placing `\end{enumerate}` immediately before `\end{document}`. The text to be included for any given question would then be placed inside the proper question number. For example, this code:

```
Here's some answers:
\begin{enumerate}
\item
Here's my answer to question 1.
\item
Here's my answer to question 2.
\end{enumerate}
```

would produce the following output:

`Here's some answers:`

1. `Here's my answer to question 1.`

2. `Here's my answer to question 2.`

3 Math mode

One of the strengths of LaTeX is the extended ability to output mathematical information. There are many ways to enter "Math mode," which allows the use of mathematical formulae, greek letters, and a host of other editing options. The math environment can be denoted using `\begin{math}` *`Math mode text`* `\end{math}` . Because math is meant to be very easy to use in LaTeX, it can also be entered by using \(... \) or \$ ... \$. For example, the text `$ \sqrt{\lambda + \epsilon} $` would produce $\sqrt{\lambda + \epsilon}$. This feature alone is enough to make LaTeX extremely useful for mathematics students who wish to prepare electronic versions of assignment solutions.

4 Advanced commands

4.1 Inserting Graphics

The insertion of graphics may also be of use in some assignments. To do this, make sure that the statement `\usepackage{graphicx}` is in the preamble of your file. Then, graphics can be included in the document by using the command `\includegraphics[height=##in]{`*filename*`}`. The ## can be replaced by a real numerical value to specify how high the image should appear in the output document. The filename specifies the name of the file the image is contained in. It is recommended that no file extension be used. Different translation tools require different image file formats; see section **??** for more details.

4.2 Tables

Tables will also be used frequently in various assignments. The table is enclosed by `\begin{tabular}` and `\end{tabular}`. The `begin` statement is followed immediately by a specification defining how many columns the table will have, the alignment of each column, and the dividers between these columns. For example, `\begin{tabular}{c||lr|c}` will create a table where the first column is centre-aligned followed by two vertical bars, then a left-aligned column and a right-aligned column with no divider between them, and finally another vertical bar before another centre-aligned column.

Each cell of the table is separated from others by an ampersand. The end of a row of the table is specified by a double backslash (\\). Dividers between rows can be denoted with an `\hline`. For example,

```
\begin{tabular}{c||lr|c}
Student & Course & Grade & Status \\ \hline \hline
imaslacker & EASY 100 & 35 & FAIL \\
imastudier & PHYS 334 & 97 & PASS \\ \hline
\end{tabular}
```

would produce:

Student	Course	Grade	Status
imaslacker	EASY 100	35	FAIL
imastudier	PHYS 334	97	PASS

4.3 Defining new commands

New commands can be defined for use throughout your document. This would be useful if there is a long string to do something that you commonly use and you want it instead it a concise format, or you wish to create a name for something that makes its purpose clear. This can be done using the `\newcommand` command. For example, in logic functions you often have to work with the negation of a variable A, and would commonly refer to this as “A bar” because it is drawn with a line over the letter. This can be reflected in your LaTeX document by creating a `\bar` command to do this for you. This would be accomplished by declaring in the preamble: `\newcommand{\Bar}[1]{\overline{#1}\,}` , and then in the main body of the document using $\Bar{A}$ to produce $\overline{A}\,$.

5 Creating a finished document

Once a LaTeX file has been created, there are a few different options to create a finished product. The undergraduate environment has two main programs: *latex* and *pdflatex*. The program latex will take the .tex input file and convert it into a .dvi (device independent) file. The program dvips can then be used to convert this file to a postscript file. A useful script, called something intuitive like "tex2ps" to convert LaTeX source files directly to postscript, would be:

```
latex $1
dvips -Pcmz $1.dvi -o $1.ps
```

The other available command, *pdflatex*, converts the source file directly from .tex to .pdf format.

Unfortunately, these two programs will not work exactly the same when image files are being imported. The program *latex* expects files to be in eps format. *Pdflatex*, on the other hand, expects imported images to be in pdf format. Making both versions of any files available and not specifying a file extension in the source LaTeX document will enable you to use the same file with both programs.

It is recommended that you create documents from your source on a regular basis. LaTeX is a very finicky system, and a small error could be quite difficult to track down later.