

1. Semester and Year : 2012 FALL
2. Course Number : CS-227
3. Course Title : Linux & System Software
4. Work Number : PR-02
5. Work Name : Presentation FINAL
6. Work Version : Version1
7. Long Date : Monday, 3, December, 2012
8. Author(s) Name(s) : Jake Waffle

# **An Introduction to L<sup>A</sup>T<sub>E</sub>X**

## **1 Introduction**

### **1.1 History**

Typesetting has been around for a good number of years throughout history. For all those who don't know, typesetting is ,". . . [T]he presentation of textual material in type format ready for printing."([1]) And I understand it as the actual act of displaying the formatted text.

Typesetting has been a problem throughout the history of humans. Some of our early solutions include rock engravings with rocks and chisels, painted caves or rocks with some sort of paint brush, letters written with ink pens, a paint brush or typewriter. But since the invention of the computers, we've de-

veloped word processors (like word or openoffice) and typesetting languages (like  $\text{\LaTeX}$ .)

## 1.2 Explaining $\text{\LaTeX}$

We now know about the problem of typesetting that has plagued humanity for some time. Now let's go into one of humanities great solutions to the problem. This solution I speak of is called  $\text{\LaTeX}$ (which you may have already heard.)  $\text{\LaTeX}$ allows an author to focus on the content they are writing instead of having to worry about the aesthetics. What I mean by that is that the actual content doesn't need to be in any specific format at all for  $\text{\LaTeX}$ to use it. So after inputting all of the content that is needed along with the appropriate  $\text{\LaTeX}$ commands,  $\text{\LaTeX}$ reformats the content in the .tex document and displays it in a consistent and neat format (multiple kinds of formats can be used.)

## 1.3 Thesis

Our goal is to learn about some practical uses for  $\text{\LaTeX}$ by a College Student. So a primer on the  $\text{\LaTeX}$ compiler and Kile, along with some similarities I see with other programming languages and finally the practical uses will be discussed. And by the end we all will be able to display the contents of a paper and display floating bodies (equations, tables.)

## 2 Body

### 2.1 The $\text{\LaTeX}$ Compiler and Kile

Unlike word processors (like Word or openoffice,)  $\text{\LaTeX}$  isn't built into a text editor. It is just a language that can be used to define how the actual formatting of a body of text will be displayed. So the  $\text{\LaTeX}$  programs can be written in a number of different text editors. In this tutorial we are going to be talking about Kile, an editor specifically for  $\text{\LaTeX}$ .

Now that we've gone over text editors and the creation of the  $\text{\LaTeX}$  programs. We now need to know how the  $\text{\LaTeX}$  programs create the actual documents. This is all done with the  $\text{\LaTeX}$  compiler. And it can be installed on a Ubuntu system with command:

```
sudo apt-get install textlive-base
```

After getting the compiler, we can now use this along with the "pdflatex" command in order to convert our .tex  $\text{\LaTeX}$  files into a .pdf format. Then a .pdf file will show up in the directory of the original .tex file.

### 2.2 Similarities Between Other Programming Languages

When listening to Daniel explain  $\text{\LaTeX}$  to me for the first time, I immediately saw similarities with other programming languages. There are three distinct areas for the code in a single-filed program for  $\text{\LaTeX}$  programs and Python programs.

The first area is the place where modules are imported in Python for use in the program and in the case of  $\text{\LaTeX}$  that's where extensions and packages are included (along with the definition for the type of document.)

The second area is where global variables are defined for use later in the program. In Python the variables are assigned values with the '=' operator. But in  $\text{\LaTeX}$ , special variables such as the title are assigned a string like so:

```
\title{This is a Title for a Paper}
```

The third area is denoted by the `\begin{document}` and `\end{document}` commands and represents a sort of main functions for the program. This can be compared to almost any programming language's structure and still holds true for  $\text{\LaTeX}$ . The "main function" in  $\text{\LaTeX}$  is where the content for the document is placed in accordance with its position in the paper (the beginning of the main corresponds with the beginning of the paper.)

$\text{\LaTeX}$  while its programs have similar structures to other programming language's programs, it also has other similarities with how things operate. Such as the use of functions.  $\text{\LaTeX}$  has a number of built-in functions that it gives the user to be able to do many things. For instance, the author can force the program to move to the next page after a certain section has been displayed with the `\pagebreak` command. These  $\text{\LaTeX}$  functions all seem to have a

"\" as their prefix.

L<sup>A</sup>T<sub>E</sub>X also like Python and C are able to include files that are in the same directory as the program file. So this makes it possible to use files written by other people as tools for a program. And this means that the program that is written can exist over multiple files (which is good for modular programming.)

## 2.3 Writing an Essay for School

When writing a school essay with L<sup>A</sup>T<sub>E</sub>X, one can use commands and extensions to alter the default formatting of L<sup>A</sup>T<sub>E</sub>X to suit a certain format type. But there are also existing extensions that will save the author the trouble of dealing with the formatting (apa.cls can be used for apa documents, journals and manuscripts ([2]).) The author is also able to define his/her own format with the L<sup>A</sup>T<sub>E</sub>X built-in commands. So with these commands one is able to recreate the mla format, or even the format asked of us by our Professor in CS-227. But a L<sup>A</sup>T<sub>E</sub>X program for an article only requires three things to yield a basic hello world.

Firstly, a L<sup>A</sup>T<sub>E</sub>X program for an article requires three things to yield a basic hello world. The first requirement is the definition of the documentclass, which can be done like so.

```
\documentclass[letterpaper,12pt]{article}
```

The arguments with the square brackets define certain aspects for the paper to make it consistent the whole way through (margin width and font size.)

While the argument within the curly brackets defines the type of document (it can alternatively be letter, book or slides.)

Secondly, a  $\text{\LaTeX}$  program requires a clear definition for the start and end for the main function. This can be done like so.

```
\begin{document}
```

```
\end{document}
```

Inside this main function, that is where we actually give the content of the document in the order that it will be presented in the final version after being compiled.

Lastly, there has to be the actual content within the previously mentioned main function. And to make sure that the content makes it to the screen in the correct format, there are a number of different commands to do specific tasks. These tasks can include justification (right-, left- or center-justification,) change in text properties (bold, italic, font size, font,) document structure (paragraphs, subparagraphs, sections, subsections,)

When creating this document in  $\text{\LaTeX}$  I used a variety of commands that gave me something close to what Daniel asks for in his assignments. But the main part of it really is just the use of document structures (sections and paragraphs.) If you'll look around the document, the parts with a number and a bold title are the section headers for the document. And the paragraph

structure is being used after the sections to store the paragraphs belonging to that section. Other than this, the header/title and the bibliography were the other two pieces that had to be put together.

The information at the beginning of the document that Daniel likes along with the title were another problem needed to be solved to create the proper formatting for this paper. But there wasn't much to it really, it just needed the proper lines of information along with a simple justification command.

The the bibliography was constructed using the thebibliography environment ([3]). With this special built-in environment for L<sup>A</sup>T<sub>E</sub>X, we can create a Bibliography page and actually cross-reference the citations from the text with a special command. This can be done like so.

```
\cite{keyitem}
```

But to actually create a citation for the bibliography, one must define the item's key and follow it with the respective citation content. And the command for adding the citation and defining its key may look like the following.

```
\bibitem{Author's surname} Citation content . . .
```

## 2.4 Displaying Floating Bodies

When displaying floating bodies (equations or tables) in a L<sup>A</sup>T<sub>E</sub>Xproject, one can do so by simply stating when they are beginning and ending the

floating body. This can be done with the `begin` and `end` keywords followed by the type of floating body in curly brackets. But there are a lot of specific commands that exist for the equations and the tables. So while they may be similar, their formatting is different and hence their commands are different.

The first thing about the equations is to know that they usually make use of extensions called `amssyb` and `amsmatt`. They can be included into a  $\text{\LaTeX}$  program like so.

```
\usepackage{amssyb,amsmath}
```

And from there, math symbols and a variety of other things needed when writing Math equations are able to be used in the program. After that, we can touch up on the different things that we can do with respect to Math in  $\text{\LaTeX}$ . Equations inside of a body of text can be used, but only if dollar signs are placed before and after the equation. Other than that, equations will have to be signaled with the `begin` and `end` keywords. Equations can also be set up in a multitude of ways. For instance, if one was wanting to make a multiline equations for showing the steps for solving an equation, one could do something like the following.

```
\begin{equation}
\begin{align*}
x &= \frac{10 + 4}{12} \\\end{align*}
\end{equation}
```



```
x &= \frac{14}{12}\\
x &= \frac{7}{6}
\end{align*}
\end{equation}
```

And this bit of code will yield something like the following.

$$\begin{aligned}
 x &= \frac{10+4}{12} \\
 x &= \frac{14}{12} \\
 x &= \frac{7}{6}
 \end{aligned}
 \tag{1}$$

The align command is one of a few different justification commands for equations. There is also a split and multiline command.

Tables are also a good thing to know about with L<sup>A</sup>T<sub>E</sub>X. They seem a little bit more involved than the equations though. The main commands that are needed for a table are the begin/end table and tabular commands. And for the content itself one would just add in the lines with an ampersand to separate the columns (the ampersand is like a tab of sort.) So ones table may be written like the following.

```
\begin{table}
\caption{This is a Title for the Table}
\centering %This centers the table
```

Table 1: This is a Title for the Table

Col1	Col2	Col3	Col4
Row1	data	data	data
Row2	data	data	data
Row3	data	data	data

```

\begin{tabular}{c c c c}
\hline\hline
Col1 & Col2 & Col3 & Col4\\
\hline
Row1 & data & data & data\\
Row2 & data & data & data\\
Row3 & data & data & data\\
\hline
\end{tabular}
\end{table}

```

And this bit of code should yield a table at the beginning of the current page.

### 3 Conclusion

So now that we've talked about how to use  $\text{\LaTeX}$  and some of its uses. It is now possible for one to use  $\text{\LaTeX}$  to write their school assignments. Since we've gone over Mathematic equations,  $\text{\LaTeX}$  could be used for a math assignment (math professors Brian and Dan both are familiar with  $\text{\LaTeX}$ .) And I'm sure that the tables would come in handy for a research paper (tables are nice for presenting data.) So I hope that whomever reads this has gained enough knowledge and confidence to start on some  $\text{\LaTeX}$  documents of their own.

## References

- [1] Gerald Jackson. (January 22, 2010). What is typesetting?. In Getting Published. Retrieved December 3, 2012, from <http://gettingpublished.wordpress.com/2010/01/22/what-is-typesetting/>.
- [2] Athanassios Protopapas. (December 9th, 2008). apa.cls. In Institute for Language and Speech Processing. Retrieved December 3, 2012, from <http://www.ilsp.gr/homepages/protopapas/apaccls.html>.
- [3] Daniel Eisenstein. (June 19, 2006). The thebibliography environment. In Scientific Linux. Retrieved December 3, 2012, from <http://james.as.arizona.edu/~psmith/proposals/SOinstructions/node29.html>.