Why T_EX and L^AT_EX?

Kofler Gerald

`gerald.kofler@student.uibk.ac.at`

June 10, 2015

1 Introduction

This document will answer the question "Why you should use T_EX and L^AT_EX?" Section 2 and Section 3 will give you an overview of what T_EX and L^AT_EX is. In Section 4 we will have a look at typesetting and typography and in Section 5 we compare L^AT_EX to other approaches for typesetting. In the last Section there will be some concluding remarks.

2 T_EX

After seeing the trend of deteriorating typographical quality affecting his own books and articles, Donald E. Knuth started writing the T_EX typesetting engine in 1977. T_EX is a low-level markup and programming language to typeset documents attractively and consistently. While being a programming language and supporting if-else constructs or making calculations, which are performed during compile time, you only want to use T_EX for typesetting. The power of T_EX lies in structuring and formatting your documents. T_EX development is frozen since 1989 with only bug fixes released periodically. The current version of T_EX is 3.1415926 and the version numbers are converging toward π.

Some trivia about the name T_EX. It is an uppercase form of $\tau\epsilon\chi$ steming from the Greek words beginning with $\tau\epsilon\chi\ldots$, like the English word 'technology'. The meaning is art as well as technology and punctuate the purpose of T_EX to produce technical documents at the finest quality. Another thing about T_EX's name is the displaced 'E' which is a reminder that T_EX is about typesetting. [1]

The learning curve for programming in T_EX is quite steep and building custom macros for text formatting is pretty time consuming. At the cost of complete design flexibility there exist some document preparation systems based on T_EX, consisting of collections of pre-built macros. These pre-built macros help the user to reduce errors, automate certain repetitive tasks and are time saving. One of these macro packages is called L^AT_EX. [3]

3 LaTeX

LaTeX is a system for typesetting documents created by Leslie Lamport. It is a special version of TeX that understands LaTeX commands. Or in the words of Leslie Lamport: "Think of LaTeX as a house built with the lumber and nails provided by TeX. You don't need lumber and nails to live in a house, but they are handy for adding an extra room." Most users of LaTeX will never need the lower-level TeX commands, although they are helpful if you want to create new packages for LaTeX.

The first widely available version of LaTeX appeared in 1985 with the version number 2.09. Its goal is to simplify TeX typesetting, especially for documents containing mathematical formulae. There exists a heap of extensions from various authors, called *packages* or *styles*, which are bundled with most TeX/LaTeX software distributions. More can be found in the Comprehensive TeX Archive Network (CTAN[1]). The current version of LaTeX was released in 1994 with the postfix 2_ε. This version contain for example improved methods for handling different styles of type, commands for including graphics and producing colors.

LaTeX's input is a file including the document's text together with commands that describe the document's structure. The output is a file of typesetting instructions, which will be converted into a printed output, generating book-quality typesetting. [2]

4 Turning Typing into Typography

When publishing a book, traditionally the author gives a typed manuscript to the publisher. The formatting of the manuscript is provided by the publisher's typographic designer. He decides the length of the printed line, what style of type to use, how much space to leave above and below section headings, and many other things that determine the look of the printed document. The designer also gives instructions to the typesetter to decide where on the page the author's words and symbols are put.

Back in the days the typesetter produced a matrix of metal type, nowadays they produce computer files for the document. In both cases the output controls the machine doing the actual typesetting.

In our case LaTeX is the typographic designer, and TeX is the typesetter. Our typed LaTeX commands are translated into lower-level TeX typesetting commands. Then TeX produces a computer file called the device-independent or dvi file consisting of binary data describing the visual layout.

---

[1]http://www.ctan.org

Comparing a human typographic designer with LaTeX, the human knows what the manuscript is about and uses this knowledge to decide how to format it. LaTeX on the other hand can't understand English or the visual structure the author indented and is unable to set the type according to this. Using LaTeX you must explicitly indicate the logical structure since LaTeX can't understand your prose.

Almost all LaTeX commands you'll use describe the logical structure of your document. Meaning you should be more concerned about the logical structure, and not its visual appearance, when writing a document. The approach LaTeX follows to typesetting can therefore be characterized as *logical design*. [2]

5 Why TeX and LaTeX?

In 1985, the year LaTeX was introduced, only few authors had the facilities for typesetting their own documents. Nowadays you can buy a WYSIWYG[2] (What You See Is What You Get) program such as Openoffice.org Writer[3] or Libre Office[4]. WYSIWYG programs are appealing, because you see exactly what your document will look like as you type it. With them it is easy to put text wherever you want, in whatever size and style of type you want. They replace LaTeX's logical design with visual design. For simple and short documents, like letters and memos, visual design is sufficient, but for scientific papers it is not useful. [2]

The approach LaTeX is using can be called WYSIWYM[5] (What You See Is What You Mean). You'll see the logical structure of the document, but can't see the final version of it while typing. LaTeX makes the formatting for you.

Some advantages to the LaTeX approach:

- Compared to the complex binary and XML formats WYSIWYG programs are using, your document sources can be read with any text editor and understood.

- You can concentrate purely on the structure and contents of the document and not get caught up with superficial layout issues.

- LaTeX takes care for you adjusting fonts, text sizes, line heights or text flow for readability.

- Since LaTeX uses logical design the document structure is visible for the user, and can be easily copied to another document. In comparision to WYSIWYG applications, which use visual design, it is often not obvious how a certain formatting

---

[2]http://en.wikipedia.org/wiki/WYSIWYG
[3]http://www.openoffice.org/
[4]http://www.libreoffice.org/
[5]http://en.wikipedia.org/wiki/WYSIWYM

was produced, and it might be impossible to copy it directly for use in another document.

- Consistency of the layout, fonts, tables and so on are provided throughout the document.

- It it easy to typeset mathematical formulae.

- Indexes, footnotes, citations and references are generated easily and will spare you a lot of time compared to WYSIWYG programs.

- LaTeX forces you to structure your documents correctly.

But there are also some disadvantages compared to WYSIWYG programs:

- When editing your document you can't (usually) see the final version of the document.

- You have to learn the necessary commands for LaTeX markup.

- If you want to obtain a special look or layout for your document, it can sometimes be difficult in LaTeX.[3]

6 Conclusion

Why and when should you use TeX/ LaTeX? It really depends on the length of the documents your writing. For short documents a WYSIWYG program is sufficient, but for articles or books you should use LaTeX, because you can concentrate on writing and let LaTeX do the formatting for you. If you want to design your document in a very specific way and want flexibility in doing so use TeX.

References

[1] Donald E. Knuth. *The TeXbook*. Addison-Wesley, 1986.

[2] Leslie Lamport. *LaTeX - A Document Preparation System: User's Guide and Reference Manual, Second Edition*. Pearson / Prentice Hall, 1994.

[3] Wikibooks. Latex — wikibooks, the free textbook project, 2015. [Online; accessed 26-May-2015].