

# COMP516 Practical 5

## (non-assessed)

### 4 November 2008

Research papers in Computer Science are typically not written using a word processor like Microsoft Word or OpenOffice, but with a typesetting system called  $\LaTeX$  (pronounced LAY-tech).

Word processors commonly have a graphical user interface, text appears as you type, what you see resembles more or less what a printout will look like (WYSIWYG), the quality of typesetting is mediocre, and text is saved in a proprietary binary format which is not future proof (see ‘OpenDocument’ for an attempt to change that).

In contrast,  $\LaTeX$  is more like a markup language (e.g. HTML). Text and formatting commands are entered and stored into plain text files using your favourite text editor. Output, in the form of DVI, PostScript or PDF files, is produced by processing those text files akin to compiling program files to produce executable code. The algorithms producing the output are quite sophisticated and the typesetting is of high quality. Like for program development, special editors exist (e.g. TeXnicCenter for Microsoft Windows or Kile for Linux) which ease the burden of this write-process-look cycle, but their use is not strictly necessary.

$\LaTeX$  was originally developed by Leslie Lamport, and its first widely available version  $\LaTeX$  2.09 appeared in 1985. The current version of  $\LaTeX$ ,  $\LaTeX$  2 $\epsilon$ , was released in 1994. However, development of  $\LaTeX$  2 $\epsilon$  macro packages is ongoing and there are new releases every few months. Both  $\LaTeX$  and  $\LaTeX$  2 $\epsilon$  are based on the typesetting system  $\TeX$ , developed by Donald E. Knuth, from 1977 onwards.  $\TeX$  is essentially stable since the release of version 3.0 in 1990. Since then only bug fixes have been added. The latest version is 3.141592 (version numbers converge to  $\pi$ ) and was released in December 2002, i.e. since then no further bugs have been discovered.

This practical and several following practicals will introduce you to  $\LaTeX$ . This document can be found at

<http://www.csc.liv.ac.uk/~ullrich/COMP516/notes/practical5.pdf>

Some of the tasks below might be performed quicker if you cut-and-paste from the PDF document.

1. We will use the Department’s Linux systems to experiment with  $\LaTeX$ . Use Exceed to connect to one of the Linux systems, log in using your departmental account name and password, and open a web browser (preferably firefox).

2. Download the file

<http://www.csc.liv.ac.uk/~ullrich/COMP516/notes/small.tex>

and store it at an appropriate place. By convention, files containing  $\TeX$  or  $\LaTeX$  typesetting code end in ‘.tex’.

3. Open the file `small.tex` in a text editor on the Linux system and have a look at it.

4. The command sequence that needs to be executed to generate PostScript from `small.tex` is depicted in Figure 1 below. To give it a try, open a terminal, change to the directory where you have stored the file `small.tex` and execute the following commands in sequence:

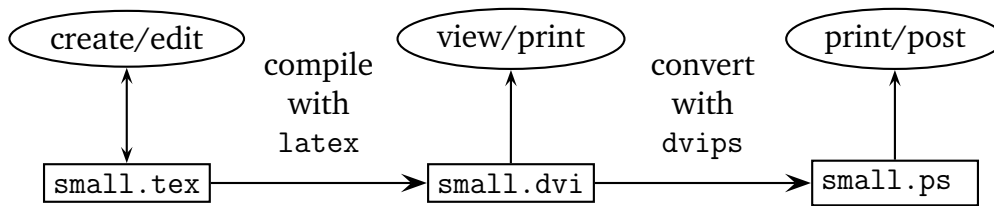


Figure 1:  $\text{\LaTeX}$  compile cycle [1]

```
latex small
dvips -o small.ps small
```

The command `latex small` creates a DVI-file `small.dvi` from `small.tex`. During the creating process it will display some diagnostic output in your terminal:

```
latex small
This is pdfTeX, Version 3.141592-1.21a-2.2 (Web2C 7.5.4)
entering extended mode
(./small.tex
LaTeX2e <2003/12/01>
Babel <v3.8d> and hyphenation patterns for american, french, german, ngerman, b
ahasa, basque, bulgarian, catalan, croatian, czech, danish, dutch, esperanto, e
stonian, finnish, greek, icelandic, irish, italian, latin, magyar, norsk, polis
h, portuges, romanian, russian, serbian, slovak, slovene, spanish, swedish, tur
kish, ukrainian, nohyphenation, loaded.
(/usr/share/texmf/tex/latex/base/article.cls
Document Class: article 2004/02/16 v1.4f Standard LaTeX document class
(/usr/share/texmf/tex/latex/base/size10.clo) (./small.aux)
(/usr/share/texmf/tex/latex/base/omscmr.fd) [1] (./small.aux) )
Output written on small.dvi (1 page, 2216 bytes).
Transcript written on small.log.
```

You can take a look at the DVI-file using a DVI-previewer like `kdvi` or `xdvi`. Execute

```
kdvi small.dvi &
```

in the terminal window to do so.

The second command that you have executed, namely `dvips -o small.ps small`, has created a PostScript-file `small.ps` from `small.dvi`. You can take a look at the PostScript-file using a PostScript-previewer like `kghostview`, `evince`, or `gv`. Execute

```
kghostview small.ps &
```

in the terminal window to do so, and compare the DVI-file with the PostScript-file.

5. We can also generate a PDF-file from `small.tex`. We can either use `pdflatex` instead of `latex` to directly process a  $\text{\LaTeX}$ -file to generate PDF or we can use `ps2pdf` to convert the PostScript-file that we already have into PDF. Execute

```
pdflatex small
ps2pdf small.ps small2.pdf
```

in the terminal window to try both alternatives. The first command will produce a file `small.pdf`, while the second command will produce a file `small2.pdf` (without the optional second argument, `ps2pdf` would also produce a file `small.pdf`, overwriting the file generated by `pdflatex` in the previous command).

PDF-files can be previewed using `kpdf`, `evince`, or `acroread`. Compare `small.pdf` and `small2.pdf` using

```
kpdf small.pdf &
kpdf small2.pdf &
```

6. PDF is the preferred format for the distribution of documents prepared with  $\text{\LaTeX}$ . It gives you a high-level of certainty that any receiver of a PDF-file will see your document in exactly the format you intended it to be and that a printout will also conform to exactly that format (the same cannot be said for Microsoft Word). PDF-viewers are available for all major platforms, even including mobile devices.

However, if you prefer to make your documents easily accessible via the web, you can also convert  $\text{\LaTeX}$ -files to HTML. There are a number of  $\text{\LaTeX}$ -to-HTML converters available. One of them is `latex2html`. Execute

```
latex2html -split 0 small.tex
```

in the terminal window (You can use `latex2html -h` to find out what the `-split 0` option does). You will see that a subdirectory `small` has been created in the directory where you have stored `small.tex`. In the subdirectory are two files, `small.css` and `small.html`. Execute

```
firefox small/small.html
```

to have a look at the result of the conversion.

7.  $\text{\LaTeX}$  has its own rules for deciding the lengths of blank spaces. In particular,  $\text{\LaTeX}$  will put an extra amount of space after a period ‘.’ if it considers that the period marks the end of a sentence. It regards a period as the end of a sentence if it is preceded by a lowercase letter. If a period is preceded by an uppercase letter then  $\text{\LaTeX}$  assumes that this a period that follows the initials of somebody’s name.

This heuristic does not always make the right decision, e.g., in ‘Mr. Smith’ or in ‘Proc. IJCAI’, the period does not end a sentence but indicates an abbreviation. This can be corrected by putting a backslash ‘\’ after the period, e.g., ‘Mr.\ Smith’ and ‘Proc.\ IJCAI’.

To see the difference insert the following text before the line `\end{document}`.

```
\begin{flushleft}
Mr. Smith (without backslash)\
Mr.\ Smith (with backslash)\
Proc. IJCAI (without backslash)\
Proc.\ IJCAI (with backslash)
\end{flushleft}
```

Save the file, execute `latex small` **once** in a terminal window, and see how the (minute) difference in the amount of space after the period depending on the presence or absence of the ‘\’. You will probably have to increase the magnification in your previewer to be able to see it.

8. LaTeX also determines by itself how to break up a paragraph into lines, and will occasionally hyphenate long words where this is desirable. However it is sometimes necessary to tell LaTeX not to break at a particular blank space, e.g., there should never be linebreak between ‘Example’ and ‘7’ in ‘Example 7’. The special character used for this purpose is `~`. It represents a blank space at which LaTeX is not allowed to break between lines, e.g., ‘Example~7’. It is often desirable to use `~` in names where the forenames are represented by initials, e.g., ‘U.~Hustadt’.

9. You can find a PostScript-file of Greenberg's introduction to  $\text{\LaTeX}$  [1] at

```
http://www.ctan.org/tex-archive/info/simplified-latex/simplified-intro.ps
```

Download the file, read chapter 2, and experiment with fonts (described in sections 2.1 and 2.4), lists (described in section 2.2), and tables (described in section 2.3).

In the following practicals we will consider labels and references (chapter 4) and citing and bibliographies (chapter 3).

10. As mentioned earlier, there are special editors for  $\text{\LaTeX}$ . One of them, Kile, is available on the Linux systems. Execute

```
kile small.tex
```

to start Kile. Use the Kile Handbook in the on-line help of Kile to learn how to use it.

An interesting feature of editors like Kile is that they allow a closer integration of editor and previewer (called 'inverse search' in Kile). To experiment with this feature, add the line

```
\usepackage[active]{srcltx}
```

to `small.tex` after the line

```
\documentclass{article}           % YOUR INPUT FILE MUST CONTAIN THESE TWO LINES
```

Process the file ('QuickBuild' in Kile terminology). Now, when you click the middle-mouse button in the DVI-viewer opened by Kile, the cursor in the editor window should jump to the corresponding point in `small.tex`.

## References

- [1] H. J. Greenberg. *A Simplified Introduction to  $\text{\LaTeX}$* . World Wide Web, <http://www-math.cudenver.edu/~hgreenbe/booksEtc/SimplifiedIntro.html>, 1999–2006. Also available at [www.ctan.org/tex-archive/info/simplified-latex/](http://www.ctan.org/tex-archive/info/simplified-latex/) and at [samizdat.mines.edu](http://samizdat.mines.edu).