Unix Tutorial 5: LaTeX

Today, we are going to introduce the LaTeX system for document processing. If you are running your own Linux distro, you might want to install the software for yourself. On Ubuntu, try

```
$ sudo apt install texlive-full
```

If your Ubuntu distro is not fully up-to-date, you may need to run this command first ...

```
$ sudo apt update
```

to refresh the package index.

Or on a Mac, download MacTeX from `https://www.tug.org/mactex/` and install the dmg file. Note these installations are multi-GBs, so will take a *long* time to download.

The alternative approach (and probably easier!) is to use `https://overleaf.com`. This is a cloud-based LaTeX service. You will need to set up an account. Because the University has a subscription to this service, you should be able to access Pro features for free by registering with your Uni email address. Check out `https://www.overleaf.com/edu/glasgow` for details.

Your first document

With LaTeX, you write a script that specifies the markup for the final document (like markdown or HTML). Then you compile this script into a pdf to view the rendered document. We recommend that you use LaTeX for reports in PSD(H) and for your Honours dissertation in your final year project. So it's worth investing some time to learn the tool now ...

Below is the simple scaffolding—the minimal number of macros that you need for a simple document.

```
\documentclass{article}
\begin{document}
Hello world.
\end{document}
```

Create this file in vim, save it as firstdoc.tex. Now you can turn it into a pdf and view it:

```
$ pdflatex firstdoc.tex
$ evince firstdoc.pdf &
```

Note that the evince tool requires a Window manager—you need to be running with a GUI rather than a terminal. Other pdf viewing apps are available, such as Acrobat Reader, pdf.js (Mozilla) and Preview.app (Mac). In fact, on a Mac you just have to type `open firstdoc.pdf` in the terminal and it loads the Preview app directly.

If you are using Overleaf, then you simply enter the LaTeX source code into the middle pane, and click to recompile the PDF in the right hand pane.

Now let's add some more complex markup to firstdoc.tex, for instance ...

```
In \LaTeX, you may use \textit{italic} and \textbf{bold} face fonts.
It is also possible to have \texttt{typewriter} and
\textsf{sans–serif} fonts.
You may change the font size from {\tiny very small}
to {\Huge very large}.
```

You will also want to include `\section` directives to structure your document. You can use `\label` and `\ref` commands to add references to different sections as follows:

```
\section{Introduction}
\label{sec:intro}

Some intro text.

\section{Another Section}
\label{sec:nother}

Please refer back to Section \ref{sec:intro}.
```

Mathematical Typesetting

In case you hadn't guessed, all the Algorithmic Foundations tutorial sheets were produced using LaTeX. Indeed, formatting math. equations is one of LaTeX's strengths. For instance, try adding the following snippets to the body of your firstdoc.tex:

```
Below is an instance of a De Morgan rewrite rule:
\begin{equation}
\lnot(A \land B) \equiv \lnot A \lor \lnot B
\end{equation}
```

which should produce this equation:

$$\neg(A \wedge B) \equiv \neg A \vee \neg B \tag{1}$$

Check out the webpage at `https://en.wikibooks.org/wiki/LaTeX/Mathematics#List_of_Mathematical_Symbols` for more details of mathematical symbols in LaTeX. Note that you can use the $ notation for inline mathematical expressions, like $\exists a \in S.(a > 0)$ and so on.

More Exercises

0.1 Make a Makefile

Remember your Makefiles from yesterday? Write a Makefile to automate the build of the `firstdoc.pdf` file from LaTeX. It will be useful to add a `clean` target that removes all the intermediate files (.aux, .log, .out) generated by the pdflatex execution.

0.2 Recall your Maths

Remember all the set equivalences you learnt in Algorithmic Foundations? Typeset them in a LaTeX document. I'll give you one to get started: it's a distributivity rule ...

$$A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C) \tag{2}$$

```
\begin{equation}
A \cup (B \cap C) \equiv (A \cup B) \cap (A \cup C)
\end{equation}
```

0.3 Timetabling

Check out the info at `https://webmaster.iit.edu/files/graduate-academic-affairs/latex-table-help.pdf` to find out how to draw tables in LaTeX. Armed with this information, construct a lecture timetable for yourself using LaTeX. Each column can represent a day, each cell an hour for a lecture. An alternative online tool that helps to generate LaTeX tables is `https://www.tablesgenerator.com/`.

Further Reading

There are plenty of LaTeX resources online.

- tutorials e.g. at `https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes`
- textbook at `https://en.m.wikibooks.org/wiki/LaTeX`