LaTeX Report Template in IEEE Style

Mike Murphy
Department of Computer Science
UiT The Arctic University of Norway
Tromsø, Norway
michael.j.murphy@uit.no

Øyvind Nohr
Department of Computer Science
UiT The Arctic University of Norway
Tromsø, Norway
oyvind.a.nohr@uit.no

***Abstract*—This document is both an example of and a tutorial for using LaTeX and the IEEE LaTeX template to write a technical report. It follows a structure that can work well for systems research in computer science and its content is an introduction to LaTeX. Students can use this as a template for their reports by replacing the content and adapting the structure. This template is tailored to the Operating Systems course (INF-2201) at UiT The Arctic University of Norway, but it should be useful for other courses as well.**

***Index Terms*—Assignment submission, LaTeX, paper, template, typesetting**

I. Introduction

The introduction section should give a brief overview of your system and the fundamental problem it is trying to solve. For an assignment report, briefly restate the assignment requirements in your own words.

This example document will explain some TeX and LaTeX concepts along the way as an example of a report.

A. Outline

An introduction section typically ends with a compact outline of the rest of the paper. The `label` and `ref` commands can be used to make cross references. Use `label` set a point in the text to refer to, then `ref` will give the section/subsection number where the label appears. For example, this is Subsection I-A. The `autoref` command gives not just the number, but also identifies the section/subsection type For example, this is Subsection I-A.

The rest of this report is organized as follows. Section II outlines concepts and background information relevant to the project. Section III describes related work. Section IV describes our solution. Section V describes our methodology for evaluating our solution. Section VI gives the results of our evaluation. Section VII discusses the results. Section VIII describes known bugs and future plans. And finally, Section IX is the conclusion.

II. Background

The background section should give a brief review of concepts necessary to understand your solution. In the case of Project 2 in this course, this would include concepts like processes, threads, context switching, scheduling, user space vs kernel space, and system calls. In the case of this tutorial, we will provide background on typesetting and on writing styles.

A. Typesetting, TeX, and LaTeX

LaTeX is a great way to program about writing while you're procrastinating on writing about programming.

Mike Murphy

Typesetting is the art of arranging text on a page for printing. Typesetting is often judged by how well a column of text lines up at its left and right edges, and by how uniform the greyness of the page appears. This involves carefully placing line breaks and then adjusting the spacing between words to achieve the most eye-pleasing result. TeX is a typesetting engine with a sophisticated algorithm to do just that [1]. For each paragraph, it tries several different arrangements, scores them, and chooses the one with the least overall "badness" score.

LaTeX is a comprehensive set of TeX macros that make TeX easier to use [2]. LaTeX provides infrastructure to separate a document's text and structure from its presentation, much like Cascading Style Sheets (CSS) later did for HTML. It also provides infrastructure for gathering macros into modular *packages*. These days it is relatively rare to use plain TeX without LaTeX. Plain TeX is probably best left to the hard-core *TeXnicians* [1].[1]

B. First Person vs. Objective Writing Styles

Some teachers are dogmatic about avoiding first-person pronouns ("I" and "we") in reports. The result is an *objective* tone that can sound more professional.

First-person	"We considered several strategies for implementing the scheduler."
Objective	"For implementing the operating system scheduler, several strategies were considered."

We are not dogmatic about this. Maintaining an objective style sometimes requires awkward phrasing or over-reliance on passive voice. A first-person style can often feel more natural both to write and to read.

III. Related Work

The related work section is a compact review of other work in your project's specific subfield. This is where you cite most of the things that you read while researching your project.

[1] A summary of relevant passages on *TeXnician* terminology is available at https://tex.stackexchange.com/a/474039/229926

What other projects inspired yours? What are you building on? What are you competing with? A well-written Background section and Related Works section will combine to form a crash course in the state of the art of the topic at the time the paper was written.

In this course you will not be doing a great deal of research, but if you, say, read up on how Linux does scheduling before you implemented your own scheduler, this the place to write about that.

For typesetting, Thành's, *Micro-typographic extensions to the TeX typesetting system* [3] gives a concise history of typesetting and explores the use of subtle techniques to enhance the appearance of text on the page.

Lamport's original LaTeX book [2] is the definitive guide to LaTeX, but there are also many resources available online. Overleaf, an online LaTeX collaboration platform, has a wealth of documentation, including a quick tutorial: "Learn LaTeX in 30 minutes" [4]. *The Not So Short Introduction to LaTeX2e* [5] is a longer tutorial (139 minutes, according to the book's subtitle). The LaTeX Wikibook [6] is a comprehensive but accessible crowdsourced guide. And *LaTeX2e: An Unofficial Reference Manual* [7] is a detailed reference for LaTeX's many commands.

The IEEE LaTeX template comes packaged with two helpful example documents.[2,3] "How to Use the IEEEtran LaTeX Class" [8] explains the many options and commands that the template provides. And the example document, named `conference_101719.pdf` or similar, is filled in with advice on typographic and grammatical details, such as avoiding math and citations in the abstract, keeping units of measurement consistent, and reminders about English words that are easy to mix up.

Those who wish to learn the inner workings of TeX and become TeXnicians must read Knuth's original *TeXbook* [1]. There are also several books available online, including *TeX for the Impatient* [9], *TeX by Topic* [10], and a separate TeX Wikibook [11].

For advice on actually *writing* a report, the short paper "The ABC of academic writing: non-native speakers' perspective" [12] offers some concise writing advice that is especially geared towards those who are not native English speakers. To go deeper on the subject of *technical* writing, useful books include Alley's *The Craft of Scientific Writing* [13] and Young's *The Technical Writer's Handbook* [14].

IV. System Description

Actually describe the system you built. We like to describe a system from the top down at four different layers of abstraction: *idea, architecture, design,* and *implementation.*

A. Idea

Start with an extremely high-level "elevator pitch" for your system. What is the main idea that motivates it and sets it apart from similar existing systems?

[3]The IEEE LaTeX template should be bundled with the source of this document. It is also available for download at https://www.ieee.org/conferences/publishing/templates.html

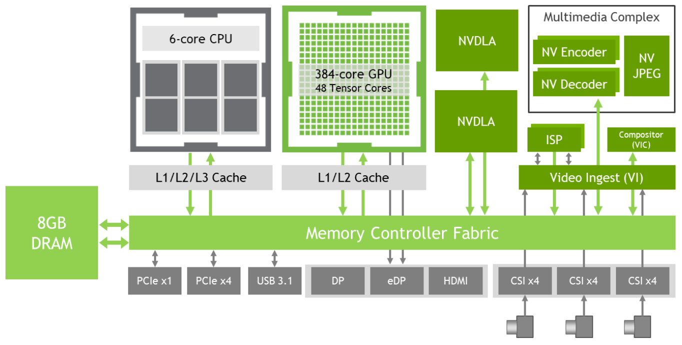

Fig. 1. Block diagram of the Nvidia Jetson Xavier NX module

```
procedure OnFloatDeclaration
    if specifier contains '!' then
        placement restrictions are ignored
    else
        placement restrictions are enforced
    if specifier contains 'h' then
        try to place float here in text
    if not already placed and specifier contains 't' then
        try to place float in top area
    if not already placed and specifier contains 'b' then
        try to place float in bottom area
    if not already placed then
        add float to the holding queue
procedure OnFinishColumn
    if holding queue not empty then
        if any floats in queue have 'p' specifier then
            use them to create a float page
    if holding queue still not empty then
        attempt to place 't' and 'b' floats in next column
```

Fig. 2. Abridged pseudocode for the LaTeX float placement algorithm [15]

Since the assignments have relatively fixed requirements, you will probably not have a lot to write here. You can omit this subsection or you can restate the requirements, the problem you are trying to solve.

B. Architecture

Then begin describing the system itself, still at a very high level. What are its major components and how do they connect? What are the most essential factors that shaped the highest-level structure of your system?

This is a good place for a high-level block diagram. Figure 1 shows a block diagram of the Nvidia Jetson Xavier edge computing AI module. Figures are part of a class of LaTeX document elements called *floats* because they float to other positions on the page. In a scientific paper, floats tend to work best at the top of a column.

C. Design

Now go a little deeper, but not too detailed. What are the smaller components inside the major ones? What factors

```c
/* Simple copy from src to dest */
char *strcpy(char *restrict dest,
             const char *restrict src)
{
    char *destorig = dest;
    for (;; dest++, src++) {
        char copiedchar = *dest = *src;
        if (!copiedchar) break;
    }
    return destorig;
}
```

Fig. 3. An implementation of `strcpy` in C

shaped your design decisions as you fleshed out the architecture? Give pseudocode for important algorithms or protocols that make your system unique. At this level, someone reading your report should be able to write a similar system in a different programming language or for a different architecture.

LaTeX has a sophisticated algorithm for placing floats [15]. Simplified pseudocode for this algorithm is shown in Figure 2. When LaTeX encounters a float declaration in the source, it tries to place it at one of the locations specified by the declaration. The top priority is "here" placement ('h'), followed by top of the current column ('t'), and finally bottom of the current column ('b'). If it cannot place the current float in the current column, it places the float in a holding queue. Then, when it finishes a column, it attempts to empty the holding queue. First it attempts to create a *float page* containing all floats in the queue that had the "page" placement specifier ('p'). Then it attempts to place the next top and bottom floats from the queue into the next column.

Note that the TeX engine works one column at a time, and it cannot backtrack. Once a column has been set, TeX flushes its output, reclaims that memory, and moves on to the next column. Therefore, depending on how your columns flow, you may have to declare a figure far ahead of the section that references it in order to have it appear in the desired column. Even then, editing your text may cause columns to reflow, which may cause floats to end up in wildly different places. That is why the common advice is to not worry about figure placement until the very end of the editing process, after you have stopped changing the actual text.

D. Implementation

Finally, describe implementation details such as programming language used and target hardware.

This LaTeX document can be rendered with a standard TeX Live distribution[4] plus the IEEE LaTeX template.[5] The build is controlled by GNU Make,[6] with help from the latexmk script.[7]

Actual code listings will typically be too low-level to include in your report. But if you do need to list code, you can use the `listings` package. An example code listing is shown in Figure 3.

[4]https://www.tug.org/texlive/
[5]https://www.ieee.org/conferences/publishing/templates.html
[6]https://www.gnu.org/software/make/
[7]https://www.ctan.org/pkg/latexmk

TABLE I
PRECISION RESULTS OF CLASSIFIERS FOR DIFFERENT FEATURE SETS

	Precision					
Classifier	(1)	(2)	(3)	(1&2)	(1&3)	All
Perceptron	0.78	0.82	0.24	0.81	0.77	0.83
Decision Tree	0.65	0.79	0.56	0.75	0.65	0.73
One-Class SVM	0.74	0.72	0.50	0.80	0.73	0.85
Isolation Forest	0.54	0.51	0.52	0.53	0.54	0.53

E. Image Formats

Images like diagrams, plots, and logos look best when the source image is in a *vector-based* image format, where shapes and curves are described geometrically, rather than a *rasterized* format, where the image is stored as a grid of pixels. Raster formats like JPEG and PNG will look pixelated if one zooms in on them, but vector formats can be re-rendered crisp and smooth at any resolution. When drawing diagrams and generating plots from data, be sure to export them to a vector-based format like PDF or SVG. The default LaTeX program, `pdflatex` can include other PDFs natively. SVG support can be enabled by using the `svg` package,[8] but be aware that the `svg` package enables SVG support by quietly converting them to PDFs during processing. This requires that you have Inkscape[9] installed to perform the conversion.

V. EVALUATION

Describe the methods used to evaluate the system. Define the metrics used to measure performance, and the procedures used to gather metrics. Include details like the specs of the hardware that ran the experiments. Do not include results yet. Those go in the next section.

VI. RESULTS

Give the results of the evaluation, including data tables and plots. Describe your results in a very concrete, quantitative way. Just the facts. Qualitative discussion can go in the next section.

An example data table is show in Table I. In this experiment, the One-Class SVM classifier had the highest over-all precision score at 0.85, followed by Perceptron at 0.83, Decision Tree at 0.73. Isolation Forest had the lowest precision, with a score of 0.53.

If you plot your data, be sure to export the plot in a vector-based format as noted in Subsection IV-E.

[8]https://www.ctan.org/pkg/svg
[9]https://inkscape.org/

VII. Discussion

Discuss your system, your experiments, and your results in a more qualitative way. What are the positives and negatives? Discuss unexpected results and their implications.

Discuss alternative solutions. What other designs did you consider? Why did you reject them? What are the trade-offs involved? Show that you understand the problem and the different ways it might be solved. Show the thought that you put into your system.

This is also a section where you can tell more of a story about your project. Did you first try other designs that did not work? Were there difficult bugs that you had to work out or other difficulties that you had to overcome?

VIII. Future Work

Discuss possible ways to improve or expand the system. If there are known bugs that you were not able to fix, describe them here and offer suggestions for how you would fix them if you had more time to do so. Similarly, if there are parts of your design that you are not satisfied with, describe how you would change them if you had more time.

IX. Conclusion

Briefly restate the key points of your work and wrap up the paper.

Note that it is ok if your report does not follow this structure exactly. Since the parameters of your assignments in this course are relatively narrow, you may not have a lot of content for every section listed. In that case you may shorten or combine sections, but try to follow the spirit of the template.

References

[1] D. E. Knuth, *The TEXbook*. Addison-Wesley.

[2] L. Lamport, *LATEX: A Document Preparation System*, 2nd ed. Addison-Wesley Professional.

[3] H. T. Thành, "Micro-typographic extensions to the TEX typesetting system," Ph.D. dissertation, reprinted in TUGboat 21:4. [Online]. Available: http://www.tug.org/TUGboat/Contents/contents21-4.html

[4] Overleaf. Learn LATEX in 30 minutes. [Online]. Available: https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes

[5] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl, *The Not So Short Introduction to LATEX2e*. [Online]. Available: https://ctan.org/tex-archive/info/lshort/english/lshort.pdf

[6] Wikibooks. LATEX. [Online]. Available: https://en.wikibooks.org/wiki/LaTeX

[7] K. Berry *et al.*, *LATEX2e: An Unofficial Reference Manual*. [Online]. Available: https://latexref.xyz/

[8] M. Shell, "How to use the IEEEtran LATEX class," 2015. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/IEEEtran/IEEEtran_HOWTO.pdf

[9] P. W. Abrahams, K. A. Hargreaves, and K. Berry, *TEX for the Impatient*. [Online]. Available: http://mirrors.ctan.org/info/impatient/book.pdf

[10] V. Eijkhout, *TEX by Topic, A TEXnician's Reference*. Addison-Wesley. [Online]. Available: https://www.eijkhout.net/tex/tex-by-topic.html

[11] Wikibooks. TEX. [Online]. Available: https://en.wikibooks.org/wiki/TeX

[12] S. Nakagawa and M. Lagisz, "The ABC of academic writing: non-native speakers' perspective," *Trends in Ecology & Evolution*, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169534724000338

[13] M. Alley, *The Craft of Scientific Writing*, fourth edition. ed. New York, NY: Springer New York, 2018.

[14] M. Young, *The Technical Writer's Handbook: Writing with Style and Clarity*. University Science Books.

[15] F. Mittelbach, "How to influence the position of float environments like figure and table in LATEX?" vol. 35, no. 3. [Online]. Available: https://www.tug.org/TUGboat/Contents/contents35-3.html

Appendix A
Use of Generative A.I.

Generative artificial intelligence tools were not used in writing this report.

In this course, we do not forbid you from using generative A.I. tools like ChatGPT, but we do discourage their use. We encourage you to write for yourself because it is a good exercise in collecting your thoughts and expressing them clearly. Good writing takes practice, and these reports are a chance to practice.

If you do use ChatGPT or other generative AI tools, you must add a section to your report explaining what tools you used and how you used them. For example, if you used a ChatGPT to generate a draft that you then edited, say that.

Appendix B
Pro Touch: Even Columns on Last Page

IEEE likes to have the two columns on the last page be about even length. To accomplish this, you can add a manual column break in your text. If the target place for the break is in normal text, you can use the following commands:

```
\vfill\pagebreak
```

The `\vfill` fills the rest of the vertical space in the column that you are ending, so that LATEX does not try to stretch the text to fit. And then the command to end a column is actually `\pagebreak` because internally columns are treated as miniature pages.

If the target place for the break is in the bibliography, you can use a special command that IEEE provides as part of the template:

```
\IEEEtriggeratref{10}
```

This will cause a column break before reference number ten.