The PracTEX Journal, 2007, No. 4
Article revision 2007/11/25

Teaching LaTeX for a staff development course

Nicola L C Talbot

Email nlct@cmp.uea.ac.uk
Website http://theoval.cmp.uea.ac.uk/~nlct/
Address School of Computing Sciences
University of East Anglia
Norwich. Norfolk. NR15 1AJ.
United Kingdom

Abstract I taught LaTeX at the University of East Anglia from 1997 to 2004 as part of the staff development course. In this article I will describe the headaches and lessons I learnt which helped me improve the course. This article is intended to assist those who are planning to teach LaTeX in a practical environment.

1 Introduction

I first started teaching LaTeX at the University of East Anglia around 1997 for the Centre for Staff Educational Development (CSED). It was the first time I had to do a presentation where the participants sat at a computer and my talk was interspersed with the participants doing a practical exercise based on what I had just described. This method does have the advantage in that the participants can get a feel for what is involved, and they have someone on hand to help them if they have problems, but it does however have disadvantages. In this article, I shall relate some of the pitfalls I fell into, and the way in which I adapted and learnt from my experiences. Some of the tips may seem obvious, but things that seem obvious to one person, may not necessarily be so to another.

2 Target Audience

The participants on my LaTeX courses were composed of university staff (academic, support and administrative), PhD students and also members of the various research institutes which, together with the university, comprise the Norwich

Research Park. So a typical class might contain secretaries, computer scientists, linguists, chemists, biologists, mathematicians, environmental scientists and, on one occasion, a librarian. Some of whom had experimented a little with LaTeX, while the others had only ever used a word processor. Such a wide ranging mix of abilities is, quite frankly, a recipe for disaster!

The computer scientists who were used to the idea of writing source code and then compiling it caught on quickly, whereas some of the others who were used to only using a word processor, by and large struggled to get to grips with the concept of a typesetting language. As a result, during the practical parts of the class, some of the participants would be finished in no time at all, whilst some of the others struggled to understand the very basics. In the end the faster ones would be flicking ahead through the notes, surfing the web or playing computer games, whilst the slower ones had to abandon the exercise because I simply could not give them any more time.

Tip 1: Try to limit your audience to those with roughly similar skill sets. Having said that, this was something over which I had no control, so instead I added optional harder parts to each exercise to occupy the faster ones. Also, bear in mind the fact that some people are very slow typers, so don't expect people to type more than a paragraph for each exercise if you are only allocating, say, ten minutes per exercise.

3 Operating Systems and Editors

In the earlier years in which I taught this course, I think that the most troublesome aspect, which at times drove me frantic, was the way in which the participants had to access LaTeX. The computer labs in which I taught all had computers with Windows installed, but there was no TeX installation. I can't remember the reason why, but it took about three or four years (or possibly more) before I finally taught in a lab with MikTeX installed. In the beginning, it was therefore necessary to remote login to the Unix computers (which had teTeX installed) via Exceed running on Windows. I realised the first time the course ran that in future I would have to give a crash course on Unix before even attempting to explain LaTeX.

The major difficulties I encountered with this setup included network failures (which thankfully didn't happen very often, but even one instance is catastrophic)

and the lack of an easy to use editor (or, at least, one which the participants found easy to use.) Another problem arose from the fact that some of the participants couldn't understand the difference between UNIX and LaTeX commands, which resulted in LaTeX source code which contained lines like `latex mydoc` and `xdvi mydoc.dvi` (followed by puzzled expressions when nothing happened.)

Most of these problems went away when we finally moved over to using local MikTeX installations. I tried out both WinEdt and TeXnicCenter, and we went with TeXnicCenter. I tested all my exercises on those computers a few days before the course started, to ensure that everything was correctly installed, and I made sure that there was a folder available for the participants to use in the event that they were unable to access their own account over the network. This made a considerable improvement to the efficient running of the course, and it gave me more time to cover more detailed LaTeX topics than previously.

Tip 2: Use MikTeX and a front end, unless you are talking to users who you know are all completely familiar with a specific operating system. This recommendation may surprise people who know me, as I am very much a Linux user and command line devotee, however most computer users these days have been brought up in a point and click environment, and they are the ones who are most likely to flounder when you talk about typing the `latex` command at the prompt and typing a LaTeX command in your document.

Tip 3: Always check the software runs without a problem a few days before the course. I was once caught out when I turned up to find that Exceed had been removed from the computers since the previous course and no one had mentioned it to me! Make sure you know in advance who to contact in the event of equipment failure.

4 Presentation

These days I always use the beamer class whenever I create a presentation, but when I started teaching I was using the seminar class, so that's what I used for the course. Initially I used `xdvi` to display the slides on a data projector, which was fine until I used specials that `xdvi` couldn't handle. Initially I just distributed handouts, but I had so many requests to keep flipping back to earlier slides whilst people were doing the exercises, that in the end I decided to put a copy of the slides on the web, so that they could go through them at their leisure. Rather than

relying on the computers to have software that could view PDF or PostScript files, I wrote a script which converted each slide to a bitmap along with HTML files which linked them together, including a table of contents and an index.

In the latter years that the course ran, I moved over to pdfLaTeX, and used Adobe Reader which was becoming more widely available and much more practical. I was then also able to supply the PDF version of the slides on the web, although I still retained the HTML and bitmap version. In fact, even though the course has finished, they are still available at http://theoval.cmp.uea.ac.uk/~nlct/latex/csed/ although they are now dated.

Tip 4: If your participants have Internet access, put your slides on the web where they can access them if they want to go back to material you have already covered. Remember however, to have a local copy of your slides (for example on a USB stick) in case of network failures. You can of course supply printed versions of the slides in 2 or 4 to a page format, but if you have a lot of slides, this might be construed as being environmentally unfriendly.

5 Dealing with Queries

My own knowledge of LaTeX has increased through my teaching simply by being asked how to do something I had never needed to do. I'm not usually worried about being asked how to do something, since if I don't know the answer, it's a good way of teaching people how to look for help. The questions that cause the most problems for me are those that I would never expect to be asked, and the hardest part is not to stare in dumbfounded amazement! If you are planning on teaching LaTeX, you will inevitably be asked what seems a stupid question, but remember that although the question may seem stupid to you, it's usually not stupid from the point of view of the one asking the question.

Most people new to LaTeX are people who have previously used word processors, so don't be surprised when they assume that their word processor's default settings are the correct way of doing things. Instead of rolling your eyes in despair and launching a tirade against word processors and the vendors thereof, be polite and direct them to books such as "The Chicago Manual of Style" [5]. In fact, it's a good idea to bring along some sample documents of good and bad typesetting to use as an illustration.

Tip 5: Bring some text books with you (such as the Companion series [1, 3, 2],

and Kopka and Daly's "A guide to LaTeX" [4]) and view awkward questions as a way of illustrating how to look for information in text books or by searching the web, such as the UKTUG FAQ and the comp.text.tex and texhax archives.

6 Course Contents

Initially I taught two separate LaTeX courses: a beginners and a follow-on or further course. Both courses were two part courses that covered two mornings or afternoons each. The problem with the follow-on course was that I had no way of knowing the actual level of expertise of the participants, as there was no requirement to attend the beginners course. As a result, the participants on the further course ranged from people who already knew everything I was describing to those who had to ask how to create a document. In the end I gave up and combined the two courses into a single four part course which ran two or three times a year depending on demand.

It took me a while to work out which subjects would interest people. Some topics that I covered, such as creating an index and using the picture environment, I dropped because they were unpopular, only to later be requested to include them. This was of course due to the wide ranging background of the participants. Some weren't in the least bit interested in typesetting mathematics, whereas some of the others had specifically enrolled (or been told to enroll) for that very purpose. This is why I decided to create supplementary material on the web. I covered the basics in the class, and was then able to direct the participants to my website for further details. However once I had finally achieved this balance, the course was cancelled due to lack of demand. The supplemental material is still available on the web at `http://theoval.cmp.uea.ac.uk/~nlct/latex/`.

Tip 6: If you are teaching a set of people with wide ranging backgrounds, it is better to briefly introduce topics and give pointers to further information. If you are teaching a set of people with the same background (for example, mathematicians) then you have the luxury to cover the most relevant topics in much more detail.

7 Acknowledgments

Thanks to Paddy Ansty in CSED for all his advice to me while I was finding my feet, and thanks to Dave Knock also in CSED for championing my cause and ensuring that MikTeX and TeXnicCenter were finally installed on the teaching computers.

References

[1] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX companion*. Addison-Wesley, 1994.

[2] Michel Goossens, Sebastian Rahtz, Eitan Gurari, Ross Moore, and Robert Sutor. *The LaTeX web companion*. Addison-Wesley, 1999.

[3] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX graphics companion*. Addison-Wesley, 1997.

[4] Helmut Kopka and Patrick W. Daly. *A guide to LaTeX2ε: a document preparation for beginners and advanced users*. Addison-Wesley, 1995.

[5] William Strunk, Jr. *The Chicago Manual of Style & The Elements of Style*. www.bnpublishing.com, March 2007.