We have the TEXnology ...

Russell K. Standish,
ANU Supercomputer Facility,
Computer Services Centre,
ANU.

TEXis a document processing system designed by Donald Knuth that takes input in the form of a simple text file with formatting commands that are preceded by a special escape character (\). In this regard it is very similar to systems such as Scribe (a DEC Vax proprietary product), and troff, (a standard Unix command). It is quite dissimilar to WYSIWYG[1] wordprocessors.

1 Why use TEX?

Overheard one day in the Forum
Sagredus:

> The first question one might ask is why use a system having a complex input language which doesn't look anything like what you finally print out, when systems are available that do show you what you final output looks like as you type it?

Salvatius

> This is very much like asking a statistician why they are analysing their data using a programming language rather than a spreadsheet. The answer to this question is usually one of a) flexibility, and b) speed. For example, the statistician may require the use of mathematical functions that aren't available in a standard package, or they may have so much data that the speed at which a spreadsheet processes the data is unacceptable.
>
> TEX is similar to a programming language in that it allows programs (called macros) to be written to do various repetitive tasks, keep track of page numbers, etc. This gives great power to the user to design document styles and page layouts. Secondly, considerable computer power is required to update a WYSIWYG display,

[1] What you see is what you get!

and this can seriously degrade the wordprocessor's performance at such things as scrolling and editing, particularly when complicated mathematics is being typeset.

Sagredus

I grant that TEX is more powerful and sophisticated than your average wordprocessor, but would your average Jo/e Blow need this sophistication, and be prepared to learn a complex computer language in order to type their papers?

Salvatius

Fortunately for the novice user, a number of sophisticated macro packages have been written to perform the usual tasks of writing papers, books, theses etc. The most popular of these is LATEX, which keeps track of page, section and equation numbering, automatically produces the bibliography, and so non. The language is fairly obvious to use, for example, `\underline{hello!}` produces $\underline{\text{hello!}}$, `\footnote{This is a footnote}` produces [2] and

```
\bar{x}=\sum_{i=0}^{n}x_i
```

produces $\bar{x} = \sum_{i=0}^{n} x_i$, so the novice user can get to work right away. Even WYSIWYG wordprocessors become nonintuitive once you start doing more complex things like tables and mathematical equations.

Sagredus

O.K., so TEX is easy to use, at least for the simple applications that wordprocessors are easy to use for. However, you haven't answered the question as to why you might need the sophistication of TEX.

Salvatius

Have you ever written a 200 equation thesis, and wanted to insert an extra equation near the beginning? Just renumbering all the succeeding equations and making sure that all references to equations within the text still refer to the correct equation is a Herculean task in itself. Added to that, many document processing systems require an add on program (such as MacΣqn) which makes the task more difficult. Another problem with WYSIWYG systems is changing mathematical notation. In most systems, it is well nigh impossible to alter every instance of α to β without having to do every one by

[2] This is a footnote

hand. A global search and replace will also replace every instance of a with b, as the Greek letter are just represented by Roman letters in a different font, and not as separate letters in their own right.

Finally, LaTeX offers automatic generation of table of contents, indices and bibliographies. So you can see that LaTeX can save you a lot of time and effort.

Sagredus

But don't some of the more sophisticated wordprocessors offer these features anyway?

Salvetius

That is true. For example, Microsoft Word offers automatic generation of tables of contents, indices and bibliographies. However, the use of these features can be quite arcane, and offers no significant advantages over the same features in LaTeX. A friend of mine once said of Word (in comparison to LaTeX) that "it is about half way there".

One area where WYSIWYG systems do shine is the ability to easily include graphics into the document at a simple mouse click. If your task was to produce a glossy brochure with lots of pictures, then you would probably be better off using a desktop publishing program. This is not to say that it is impossible to include graphics into a LaTeX document, only that it requires a little more work. However, the savings in effort in other areas more than makes up for this extra work, and the results are often superior.

Sagredus

You keep on mentioning mathematics. Does this mean that the program is only useful to mathematicians, or would this be of use to a colleague of mine in the Department of Ancient History?

Salvatius

Most physical scientists use mathematics as a language for communicating amongst themselves, and mathematics is a very demanding language to typeset. In fact any language based on the Roman alphabet can be typed in LaTeX. However, if you are not using the mathematical features, then it is probably on a par with the top of the range wordprocessors as far as user functionality is concerned. There is a lot of typesetting intelligence built into TeX, and so the results do look more professional at the end of the day. The one main advantage of TeX under these circumstances is that it is absolutely FREE! You will need to shell out about $50 for a manual, but it is still an absolute bargain.

Sagredus

I can see now that it is very much what is the appropriate horse for the course. On large documents like books, LaTeX looks like a clear winner, but for small documents, such as a two page letter, wouldn't a wordprocessor be easier?

Salvetius

I can see the merit in using wordprocessors for typing letters. There are a lot of people though who continue to use LaTeX for letters and making overhead transparencies. I personally prefer the "hand-written" look, and use the traditional method of quill and standish.

Sagredus

I can see now the merits of using a text oriented document processor such as TeX. However, aren't there several such systems to choose from?

Salvatius

Yes. For example, there is the troff system available on Unix systems, and the Scribe system available for VMS, Unix and a few others. The troff system does provide a document description language similar to TeX, but is not as sophisticated. Scribe is, on the other hand, quite sophisticated. In fact, LaTeX took all the good features of Scribe into the TeX environment. However, Scribe is a commercial package, with a hefty license fee, and for this reason will never become a standard. TeX, on the other hand, is being used by people all around the world, and some journals are accepting articles submitted by email in TeX.

2 Using TeX at ANU

Good public domain versions of TeX exist for most popular computer systems, in particular PCs, Macintoshes, Amigas, generic Unix and VMS systems. A good TeX environment requires a lot of disk space (around 5 Megabytes min), core memory (around 1 Megabyte min), reasonably swift processors and I/O, a graphical display for previewing and a high quality printer.

Macintoshes from about the Mac+ up can run a program called OzTeX. If you don't have a hard disk, then you can access a fileserver volume on the campus mainframe CSC2. Even if you do have a hard disk, then it is useful to copy software from a central location, particularly fonts and style files you don't use all the time.

To access the fileserver, open Chooser from the Apple menu, click on the Appleshare icon, set the zone to ANU and open the CSC Unix fileserver. You will be asked for a userid and password. If you don't have and account on CSC2, then just log on as a guest. You will then be presented with a number of volumes, one of which is OzTeX.

The procedure to set up OzTEXfor use is explained in OzTeX:Help-files:OzTeX at ANU. Whilst you can run OzTeX by double clicking on the OzTeX icon, this is not recommended as it causes a lot of network traffic, and will slow your and other people's work down. Certain files which are accessed frequently should be copied to a local floppy or hard disk, as explained in "OzTeX at ANU".

OzTEX 1.3 has just been installed, the main change being an upgrade to TEX 3.0. This change is largely transparent to users, read the Changes file if you are interested in the details. Also on the volume is a copy of BibTEX, a system for preparing bibliographies; dvim, an imagewriter driver; MEdit, a macro editor with a special LATEX macro environment and ΣEdit, a simple desk accessory editor (useful for small memory Macs that can't run multifinder). Also distributed with OzTEX is metafont and the Web/Tangle software. These latter programs are MPW tools. Contact me directly if you are interested in obtaining copies.

An even better TEX environment is provided by an X-windows terminal. An example is the Sun 4/20 SPARCstation. Here, you would have a window running the GNUemacs editor, which understands quite a bit about TEX, a second window in which you run TEX, and a third window running xtex, an X-windows TEX previewer. The whole process of saving a source file, running TEX on it, and seeing the results in the previewer takes a couple of seconds, and about four button presses. This is practically WYSIWYG, without the disadvantages. It is hoped that TEX will be installed on the CSC2 system so that users of the Workstation Laboratory can use the workstations for previewing TEX.

If anyone is interested in running TEX on PCs, a couple of good public domain implementations do exist, and a fileserved volume like that of OzTEX's could be set up.

3 Mixing Graphics with TEX

An often asked question is how one includes graphics (e.g. diagrams and figures) into a TEX document. There are many ways of doing this, and it will often depend on your needs, and what packages are available as to how you do it.

To appreciate how this can be done, you need to understand that TEX translates its input into the DVI page description language. This consists of three types of statements: the first is to specify that a particular character from a particular font is to be placed at such and such position, the second is to place a rectangular blob of ink (called a rule) at such and such position, and the third is the \special command that may be interpreted by the printer driver however it wishes.

I hope to try all the different ways of including graphics, as a form of "benchmarking" them. At the moment, I will just list all the different ways I know, and their general features.

1. LaTeX picture environment. This is built into LaTeX, and so is portable and easy to use. Unfortunately, it is extremely limited. It works by using a special font which contains circles and line segments at various angles.

2. PicTeX. This is a macro package that sits on top of LaTeX, that allows much greater flexibility than the inbuilt environment. It works by creating curves out of little line segments that can be represented in the LaTeX picture environment. As a result it is extremely memory and cpu intensive, and still limited in its capability. However, it is easily portable to any TeX environment (that has sufficient memory capacity) in that you can easily supply the PicTeX sources along with your document.

3. \special. This allows an arbitrary file containing printer commands to be included at that point. Obviously this is printer dependent, and the previewer will not be able to interpret its contents. However, more and more printers are now accepting their commands in a language called postscript, and so it is quite reasonable to include a postscript file containing the graphics. The structure of this command is **\special**{ filename x y **translate** } where x and y are integers used to position the image on the page. For example, **-72 36 translate** means move the image one inch to the left and half an inch up. Often it requires a little experimentation to get the graphics position correctly – sometimes the graphics is completely off the page to start with. If you are using an X-windows computer, then a program called Ghostscript is available which will preview postscript files.

4. Macintosh graphics. Any Macintosh graphical output can be included with the \special command by producing a postscript file. This is done by printing to a laserwriter, but holding down command-F while clicking on the OK button. Instead of the output being sent to the printer, it is saved in a file called postscript$n$ in the System folder, where n and integer. This doesn't appear to work with background printing, so turn it off in the chooser menu. You will also need to include a header file which contains postscript macros that the Macintosh expects. This file is basically a copy of LaserPrep. Most dvi to postscript translaters will have an option that allows this inclusion.

5. Brendan McKay's picture macros. These work by means of \special special commands that are interpreted by a special version of the dvi to postscript translater. It allows access to many nifty postscript features (such as rotated text, Bezier splines and greyscales) at a LaTeX level. This means

that TEX fonts can be easily integrated with graphics, and the results in general look far cleaner and more "integrated". The drawback is that it is only available on some systems (VMS and Unix) and not readily portable to other systems. Hopefully, a Macintosh version will be available soon.

6. Other packages which produce postscript, for example Mathematica and Arnstein Pritz's FPLoT program via a HP Plot to postscript translater. Two packages worth special mention here are xfig, which is a MacDraw like package for X-windows; and GNUPlot, which produces graphs of data and of functions. Both of the latter are public domain, and produce not only postscript files, but also TEX source files compatible with the LATEX picture environment and PicTEX.