GNU Emacs as a Front End to LaTeX

Kresten Krab Thorup
Dept. of Mathematics and Computer Science
Institute of Electronic Systems
Aalborg University
DK-9220 Aalborg Ø
Denmark
krab@iesd.auc.dk

Abstract

As LaTeX and TeX are more widely used, the need for high-level front ends becomes urgent. This talk describes AUC TeX, one such high-level front end for LaTeX, written for GNU Emacs.

AUC TeX does not at all attempt to be WYSIWYG; rather, it offers the author of a LaTeX document a plain ASCII file, together with a number of features to simplify and help the author perform certain tasks such as running (LA)TeX, finding errors, and simply typing in the document.

Background — The GNU Emacs Paradigm

GNU Emacs is a powerful text editor which, very much like TeX, leaves the sophisticated Emacs user with the choice of affecting its behavior on demand. The extension language of GNU Emacs is a derivation of lisp called Emacs-lisp. Just as TeX allows you to associate functionality to a token, Emacs lets you bind functionality to the keys of your keyboard. When TeX implicitly inserts a `\vbox` whenever a plain letter is encountered, Emacs implicitly calls the function `self-insert-command` (which simply inserts the letter in the current buffer) whenever a key is pressed. Emacs-lisp is itself a fully featured general-purpose language; this makes it possible to make it behave anyway you want.

At my site, Emacs is being used for many different applications, from various kinds of text editing (such as source code) to Mail and Usenet news reading. The low-level editing features of Emacs will stay the same regardless of the current application. As a user becomes familiar with the basics of the editor, he will have only a few problems converting to a new application.

As many users are used to (and pleased with) Emacs, it is desirable to use Emacs for (LA)TeX editing too, and this is where AUC TeX enters at the scene. AUC TeX is a general customizable environment for editing LaTeX documents. It is written entirely in Emacs-lisp, which makes it (relatively) easy to modify to suit personal taste and needs.

Advantages of Structure-Oriented Editing

AUC TeX is an application for editing (LA)TeX documents, especially LaTeX. The most general advantage is that by knowing the general structure of a LaTeX document, which is quite simple, AUC TeX can help a user perform certain tasks. The following is an outline of the major features of AUC TeX.

- Insertion of templates for logical-structural compositions such as environments and sections.
- Hot-keys for easy access to certain often used constructs, e.g., font changes, accented letters, and mathematical symbols.
- Running application programs (such as TeX), and then parsing the output so that errors in the document may be located easily.
- Support for multi-file documents.
- Online help for (LA)TeX error messages.
- Outlining — i.e., manipulating the document as a composition of nested/sequential logical constructs.
- Instant formatting and indentation of the ASCII-document in order to make it easier to read.
- 'Completion' (and thereby spell-checking) of partially written control sequences.

AUC TeX incorporates a large number of well-known facilities for user interfacing. At first glance, it may seem that it's just too much, but it has been put together in such a way that you can easily use just parts of it, without even knowing about the rest. Though featuring a lot of fancy functionality, AUC TeX still conforms to the standard Emacs environment — basic operations such as cursor movement and file handling are the same.

Many of the features of AUC TeX which are not basic Emacs functionality are implemented conforming to certain unwritten conventions, so that if you have already tried some other Emacs mode, such as C-mode for instance, you will simply know what do do.

Writing a LaTeX Document

I guess it is about time to let you know how it really works. We will now go on a little journey through the world of AUC TeXand explore some of its features.

Getting started. To start AUC TeX you simply run Emacs, with a LaTeX file as argument:

```
prompt$ emacs myfile.tex
```

Emacs will start and enter LaTeX mode. If you have already started Emacs, you may enter LaTeX mode, by typing `M-x latex-mode`.[1]

The first thing you are likely to do is to insert a template for the overall document structure. To do this, press `C-c C-c`. You will be prompted to insert an environment. Since the document is empty so far, AUC TeX will choose `document` as the default environment. Now type `RET` and you will be prompted for a document style, which defaults to `article`. Typing `RET` once more will prompt you for a list of style options. Write something and type `RET` again. Now AUC TeX will display a template something like the following on your screen:

```
\documentstyle[a4,12pt,dk]{article}

\begin{document}
_
\end{document}
```

and the cursor will be placed at the `_`.

[1] Emacs keying sequences are usually a combination of `Control` + another key, or `Meta` + another key. Thus, the notation `C-x` means "while holding down the `Control` key, type the `x` key"; `M-x` means "press and release the `Meta` key (which can be system-dependent) and then type the `x`." Combinations of sequences (such as `C-c C-x`), or combinations of sequences + explicit words (such as `Meta-x latex-mode`) are also possible.

To insert some sectioning command, press `C-c C-x`, and you will be prompted for a command (section, chapter, etc.), a title, and a label for it. Again, AUC TeX will look at the document so far, and choose some appropriate default for the command, in this case `section`.

In general, environments are inserted with the `C-c C-c` sequence. Some of the environments have special handlers attached to them: if you are inserting a figure environment, it will ask for placement modifiers, label, caption and whether the figure should be centered or not.

Completion. Since you have now specified the document style and options, AUC TeX is now (in principle) aware of all the commands you may use in this particular document. One of the AUC TeX advantages is to allow *completion* in various situations. To try this, type `C-c C-c` again, and press `TAB`. AUC TeX will now display a list of all the environments you can possibly use in the current document. If, for example, you want to insert a verbatim environment, just type `ver TAB`, and AUC TeX will complete the word `verbatim` for you. In case more environments start with the sequence `ver`, it will complete as much as it unambiguously can, and display a new list of possible completions.

Another, more general application of completion is the completion of control sequences. Type a part of some command, and press `ESC TAB`. If you want to insert the command `\thispagestyle`, which is very long and tedious to type in, especially since you are likely to introduce an error, you can simply type

```
\this ESC TAB
```

and AUC TeX will complete the entire command `\thispagestyle` for you. As before, a list of possible completions will be displayed in case of ambiguity.

Invoking LaTeX

Now suppose you'd like to process the contents of the buffer, i.e., run the file through LaTeX. This is handled very easily from within AUC TeX. Press `C-c C-a` (Mnemonic: do it *all*), and the current view will be split in two, of which the lower half is used for TeX output, while you can still edit the document in the upper half.

AUC TeX also allows you to process only part of a document. This is done by marking the region you'd like to have processed, and pressing `C-c C-d` (Mnemonic: *don't* try to remember it). A temporary file to be processed by (LA)TeX will then be created in the current directory, in which AUC TeX will put

the preamble there (i.e., from `\documentstyle` to `\begin{document}`), after the marked region, and then insert an `\end{document}` in the bottom.

Multi-file documents. In case your document is spread over several files, AUC TEX can handle that too. If you insert the sequence:

```
%% Master: somefile.tex
```

then the file `somefile.tex` will be the file actually to be formatted if you invoke `C-c C-a`. Also, if you invoke `C-c C-d` to format just a part of the document, then the preamble will be sought in that file.

Another mechanism is also provided. If you have neither specified a `%% Master` line, nor does your document contain a proper preamble, then AUC TEX will insert a command to load the file `texheader.tex` in the beginning of the file, which is then supposed to contain some standard preamble.

Debugging facilities. In case errors occur, the message 'errors!' is shown in the echo area, and you are asked to press `C-c C-n` (Mnemonic: *next* error) to locate the first error. Doing this will place the cursor as close as possible to the first reported error, and a description of possible causes of the error is displayed in the lower half of the view.[2] This may be repeated as often as there are more reported errors. Please note that one error is likely to produce more, so if you don't understand what some error message means, it may be a good idea to reprocess it all, to see if your changes have perhaps eliminated some errors.

Locating the error To find some error, AUC TEX parses the log file. This is perhaps one of the most interesting parts of AUC TEX. The parsing is based on the fact that whenever TEX encounters an error, it will print something like the following sequence to the log file:

```
! Something's wrong--perhaps a missing \item.
```

⟨context lines⟩

```
l.234 \section
                {Formatting}
```

This means that the error "Something's wrong–perhaps a missing `\item`" occurred at the control sequence `\section` of line 234 of the current buffer. This spot is quite easily located — and this is where the interesting part begins. Whenever TEX reads a file, it will print some sequence like the following to the log file:

```
(somefile.tex [3] [4]
```

⟨other log messages⟩

```
)
```

The parsing of this construct is complicated somewhat by the fact that ⟨other log messages⟩ may actually be any arbitrary text, and especially parentheses, which may be unbalanced, and perhaps even followed by things that may look like file names.

The Big Picture

Several features of AUC TEX are aimed at making it easy to overview your document. This can be a great help, especially if you must edit documents not written by yourself. In this talk, I will describe the features for formatting, and outlining.

Formatting When you write a document using AUC TEX, you will notice that the text is automatically formatted and indented as you write it. Lines are automatically wrapped at a particular column, and the left margin is also adjusted to reflect the structure of the document.

To get an idea of what this formatting stuff means, here is a sample of the 'item list' from the beginning of this article as it appears in the document.[3]

```
\begin{itemize}
\item Insertion of templates, for
  logical-structural compositions such as
  environments and sections.
\item Hot-keys for easy access to
  certain often used constructs, e.g.,
  font changes, accented letters, and
  mathematical symbols.
\item Running application programs
  (such as \TeX), and then parsing the
  output so that errors in
  the document may be located easily.
  ...
\item 'Completion' (and thereby
  spell-checking) of partially written
  control sequences.
\end{itemize}
```

There are several advantages in such a formatting scheme. Most important, it is easy to locate a given point in the document, as the ASCII-text reflects the actual printed document. Moreover, the indentation is a great help in localizing errors in the document — if the indentation doesn't look right, you've probably missed some closing construct, such as an `\end` tag.

There are many aspects of formatting in AUC TEX. First of all, instant processing is automatically

[2] Leslie Lamport kindly supplied me with the "help" texts for (LA)TEX error messages, as descibed in Lamport (1986)

[3] The verbatim sample shown here is formatted with a narrower margin than in the actual document in order to fit the column.

taking place, while you write a document. Next, reformatting of paragraphs is very useful to clean up a some messy construct, and this even works for things like an item of a list. Last, general reformatting features are available, which let you reformat sections, environments or the entire document. Refer to the function listing in the appendix of this article for further information.

Outlining. A special minor mode is available along with AUC TeX to allow outlining of a LaTeX document. The outlining feature allows body text or subheadings to be made temporarily invisible or visible again. Such invisible text is attached to the end of the heading to which it belongs, and moves along with it. A heading under which some body text is hidden is marked with an ellipsis (`...`). For example, the current section looks like this, when totally collapsed:

```
\section{The Big Picture} ...
\subsection{Formatting} ...
\subsection{Outlining} ...
```

This outline mode is enabled via the command `M-x outline-minor-mode`, after which certain key sequences can be issued to manipulate structural elements of the document. See Thorup (1992) for further documentation.

Other Subtle Features

Mathematical symbols. A special minor mode is available for easy access to mathematical symbols, which is often convenient when writing an application full of them. The general idea is that once you've entered this mode, pressing the sequence '⟨left quote⟩-⟨letter⟩' causes some symbol to be inserted, e.g., `‘-a` inserts `\alpha`, `‘-b` inserts `\beta`, etc. The translation is controlled by a table, which may be easily redefined if needed.

Accented letters. As with mathematical symbols, there is another a minor mode for entering accented letters with the key sequence '⟨accent⟩-⟨letter⟩'. The mapping is easily redefined by the user.

Availability

AUC TeX is available by anonymous ftp to the address `iesd.auc.dk`, but should also be available at major TeX archives around the world.

If you do not have ftp access, you may send mail to `auc-tex_mgr@iesd.auc.dk` who will be happy to mail you a copy of the latest release.

A version of AUC TeX for Freemacs (Emacs for the IBM PC), written by Richard Flamsholt Sørensen (`richard@iesd.auc.dk`) is also available as part of the Freemacs distribution.

Acknowledgements

I should like to thank Per Abrahamsen, Lars P. Fischer and a lot of unnamed people on the net, for contributing to the discussion/development of AUC TeX; ICL Data Denmark for sponsoring my travel to the Annual Meeting; and finally Leslie Lamport, who supplied me with the help text for LaTeX error messages.

Bibliography

Lamport, Leslie. *LaTeX, A Document Preparation System*. Reading, Mass: Addison-Wesley, 1986.

Stallman, Richard M. *The GNU Emacs Lisp Reference Manual*. The Free Software Foundation, 1992.

Thorup, Kresten Krab. *The AUC TeX Reference Manual*. To appear, 1992.

Functional summary for AUC TEX version 5.6

Run TEX/LATEX on buffer	C-c C-a
Run TEX/LATEX on region	C-c C-d
Print the DVI file	C-c !
Preview dvi file	C-c C-p
Next error in TEX/LATEX session	C-c C-n
Run BibTEX on buffer	C-c @
Run makeindex on buffer	C-c #
Kill job	C-c C-k
Re center output buffer	C-c C-l
Toggle Debug Boxes	C-c C-w
Home Buffer	C-c C-h
Insert **bold** syntax	C-c C-b
Insert *italics* syntax	C-c C-i
Insert roman syntax	C-c C-r
Insert *emphasized* syntax	C-c C-e
Insert `typewriter` syntax	C-c C-t
Insert *slanted* syntax	C-c C-s
Insert SMALL CAPS syntax	C-c C-y
Insert Sectioning command	C-c C-x
Insert `\begin{}...\end{}` environment	C-c C-c
Comment out a region	C-c ;
Comment out a paragraph	C-c ’
Un-comment a region	C-c :
Un-comment	C-c "
Insert item	M-RET
Format a paragraph	M-q
Format a region	M-g
Mark a section	M-C-x
Format a section	M-s
Mark an Environment	M-C-e
Close off an Environment	C-c C-f
Format an Environment	M-C-q
Complete Symbol	M-TAB
Up-list	M-}
Terminate Paragraph	RET
Smart “Quote” Insert	"
LATEX-indent-line	TAB
Re-indent, then newline and indent	LFD
Terminate paragraph	C-c LFD