AUC-TeX を用いた LaTeX 文書の作成

守岡 知彦

1 AUC-TeX とは

AUC-TeX は GNU Emacs/XEmacs で動作する TeX・LaTeX 文書作成支援ツールである。LaTeX で文書を作成する際には、さまざまな

- 命令（*foo*{...} のような形式）
- 環境（`\begin`{*foo*} ... `\end`{*foo*} のような形式）

を適切に指定する必要がある。こうしたものを字面通りに手で一字一字打つのは繁雑であるし誤りの元にもなりかねない。そこで、エディタがこうしたものの入力支援を行うことは大変有益であるといえる。

AUC-TeX はこうした入力支援機能に加え、LaTeX 文書のコンパイル（DVI 形式への変換）やプレビュー・印刷などの処理をエディタ上から呼び出したり、エラーを判りやすく表示するなど、LaTeX 文書を作成する上で有用な機能を豊富に備えている。

ここでは、XEmacs CHISE（以下では、単に XEmacs と呼ぶことにする）上で AUC-TeX を動かした場合を例に説明する。

2 ファイルを開く

XEmacs ではファイルを開くには

1. toolbar 上の  **Open** ボタンをクリックする
2. **File** メニューをクリックした時に出てくる項目 **Open** をクリックする
3. Control を押しながら x を押した後、Control を押しながら f を押す

   [Note] 以下では「Control を押しながら x を押すこと」を **C-x** と表記する。即ち、ここでの操作は **C-x C-f** となる。

4. ALT を押しながら x を押し（あるいは、ESC を押した後、x を押し）、その後に find-file と打った後、Enter を押す

   [Note] 以下では「ALT を押しながら x を押す（あるいは、ESC を押した後 x を押す）」ことを **M-x** と表記する。また、「Enter を押す」ことを CR と表記する。即ち、ここでの操作は **M-x find-file** CR となる。

といった方法がある（通常は 3 の方法を用いる)。これらのいずれかを行えばファイル名の入力プロンプトが現れ、利用者に対してファイルの入力が求められる。そこでファイル名を指定し Enter キーを押せば、指定したファイルが開く。

ファイル名の入力プロンプトで TAB を押せば既存のファイルの一覧が表示される。また、何文字か指定した状態で TAB を押せば、

- 指定した文字から始まるファイル名が複数存在する場合、その一覧が表示される
- 指定した文字から始まるファイル名のあと、共通する文字列を持つファイルが複数存在する場合、その共通部分が補完された後、一覧が表示される
- 指定した文字から始まるファイル名が 1 つしか存在しない場合、ファイル名が補完される

という動作を行う。これを **ファイル名補完** という。このような補完機能は XEmacs のさまざまな部分で利用可能であり、長い名前の入力の労力を軽減することができる。[1]

この機能は既存のファイルを開く時にも新規にファイルを作成する場合にも有効である。

3 新規文書の作成

C-x C-f を押し、新たなファイル名を指定すれば、新規ファイルが開かれる。

例えば、C-x C-f test.tex CR と押すと、新規ファイル test.tex を編集するためのバッファが開き、画面の一番下に `(New file)` と表示される。

ここで、C-c C-e と押すと

```
Master file:  (default this file) カレントディレクトリ
```

ないしは

```
Environment type:  (default document)
```

というプロンプトが出る。前者の場合、ここでは単に CR を打つこととする。またその場合、次に後者のプロンプトが出る。いずれにせよ、後者のプロンプトが出た時に CR を押すと

```
Document style:  (default article)
```

というプロンプトが表示される。

ここで、jarticle CR と押すと、

```
Options:
```

と表示されるので、a4j CR と打ってみよう。

すると、バッファに

```
\documentclass[a4j]{jarticle}

\begin{document}

\end{document}
```

が挿入される。

[1] よって、GNU Emacs/XEmacs を使う場合、入力の手間を考えて短い名前を使うよりも、長くても判りやすい名前を使うのが良いといえる。

4　章・節の入力

\begin{document} から \end{document} の間の本文領域で C-c C-s を押すと

```
Select level:  (default section)
```

と表示される。ここで CR を押すと

```
What title:
```

というプロンプトが出るので題名を入力し CR を押す。

すると、

```
What label:  sec:
```

というプロンプトが出るが、ここでは CR と押すことにする。

すると、\section{指定した題名} が挿入される。

即ち、C-c C-s test CR CR と打つと\section{test} が挿入できる。

次に、C-u C-c C-s と打つと

```
Select level:  (default subsection)
```

というプロンプトが出る。前回と同様に題名を入力した後 CR CR を押すと\subsection{指定した題名} が挿入される。

その次に、C-u C-c C-s と打つと

```
Select level:  (default subsubsection)
```

というプロンプトが出る。つまり、C-c C-s の前に C-u を付けると、1 つ下のレベルの節を入力することができる。

C-c C-s を押した後、TAB を押すと指定可能な章・節の種類が表示される。ここでは、ファイル指定の場合と同様な **TAB 補完** が利用可能である。例えば、C-c C-s Sec TAB と押すと Sec が Section に補完される。

5　書式の指定

C-c C-f C-b と押すと \textbf{} が挿入される。この中に入力した文字列は**太字 (bold face)** で表示される。この C-c C-f から始まるものは書式に関する命令の入力を表しており、C-c C-f C-b 以外にもさまざまなものが存在する。表 1 にその一覧を示す。

C-c C-f からはじまるキーシーケンスで入力可能な書式命令の一覧は C-c C-f ? を押すことによって表示できる。ここで、MATHFONT とあるのは、数式モード（$...$ で囲まれた範囲など）で利用可能なものである。また、TEXTFONT とあるのは数式モード以外の通常のモードで利用可能なものである（表 1 に挙げたものはこの一覧である）。なお、このヘルプ・ウィンドウは C-x 1 を押せば消える。

キー	TEX の書式指定
C-c C-f C-b	\textbf{ }
C-c C-f C-c	\textsc{ }
C-c C-f C-e	\emph{ }
C-c C-f C-f	\textsf{ }
C-c C-f C-i	\textit{ }
C-c C-f C-m	\textmd{ }
C-c C-f C-n	\textnormal{ }
C-c C-f C-r	\textrm{ }
C-c C-f C-s	\textsl{ }
C-c C-f C-t	\texttt{ }
C-c C-f C-u	\textup{ }
C-c C-f C-d	書式指定の消去
C-c C-f ?	書式関連キーの一覧の表示

表 1: 書式関連キーの一覧

6 命令の指定

C-c C-m（あるいは、C-c CR）を押すと

```
Macro:  (default foo)
```

というようなプロンプトが表示され命令を入力することができる（*foo* の部分は前回入力した命令が表示されるので、毎回変わる）。

単に CR を押すと *foo* の部分に表示されているものが入力されるが、それ以外のものを入力することもできるし、ファイル名の場合と同様に TAB 補完も利用できる（即ち、利用可能な命令の一覧が表示される）。

指定した命令にオプションが存在する場合、各オプションに対する入力プロンプトも表示され、命令の入力を対話的に行うことができ、うろ覚えでも入力しやすい。

7 環境の指定

C-c C-e を押すと

```
Environment type:  (default itemize)
```

というようなプロンプトが表示され環境を入力することができる（C-c CR の場合と同様に、*itemize* の部分は前回入力した環境が表示される）。

命令の入力と同様に、単に CR を押すと *itemize* の部分に表示されているものが入力されるが、それ以外のものを入力することもできるし、ファイル名の場合と同様に TAB 補完も利用できる（即ち、利用可能な環境の一覧が表示される）。

指定した環境にオプションが存在する場合、各オプションに対する入力プロンプトも表示され、環境の入力を対話的に行うことができ、うろ覚えでも入力しやすい。

8 コンパイル

まだ保存してないバッファで C-c C-c を押すと、

```
Save file ファイル名 (y or n)
```

というようなプロンプトが表示される。ここで、[y] を押すとバッファが保存され、

```
Command: (default LaTeX)
```

というプロンプトが出る。既に保存済みのバッファの場合、C-c C-c を押した後すぐに上記プロンプトが出る。

ここで、[CR] を押すと、LaTeX 処理系（latex や platex などのコマンド）が実行される。

この後、C-c C-l を押すと、LaTeX 処理系の実行の様子を見ることができる。なお、この時開いたウィンドウは C-x 1 を押せば閉じる。

LaTeX 処理系の実行開始後、LaTeX 形式上の間違いがなく、DVI 形式への変換処理が正常に終了した場合、

```
LaTeX: successfully formatted {number} pages.
```

という表示が出る（*number* の部分には変換後の頁数が表示される）。

また、なんらかのエラーが発生した場合、

```
LaTeX errors in `*ファイル名 output*'. Use C-c ` to display.
```

という表示が出る。この時、表示に従って C-c ` を押すと

```
ERROR: Undefined control sequence.

--- TeX said ---
l.314 \x

--- HELP ---
TeX が未定義の命令名を発見しました．おそらく入力の誤りでしょう．もしこ
のエラーが LaTeX 命令の処理中に生じた場合は，その命令は間違った位置に置か
れています．例えば，リスト環境の中でないのに\item 命令が使われた場合など
です．また，\documentstyle 命令がない場合にもこのエラーが生じます.
```

というようなエラーの解説が表示され、誤りがあった行にカーソルが移動する。そこで、この解説を参考に誤りを修正する。なお、C-x 1 を押せば、このエラー解説ウィンドウは閉じる。

9 プレビュー

コンパイルが正常に終了し、DVI ファイルができた後、C-c C-c を押すと、

```
Command: (default View)
```

というプロンプトが表示される。ここで、[CR] を押すと、

```
View command: xdvi ファイル名.dvi
```

というようなプロンプトが表示される。ここで、[CR] を押すと、プレビューアー（この例の場合、xdvi）が起動し、変換した DVI の画面イメージを見ることができる。

10 印刷

コンパイルが正常に終了し、DVI ファイルができた後、C-c C-c を押すと、

```
Command:  (default View)
```

というプロンプトが表示されるが、ここで、Print （TAB 補完が効くので、pr TAB ぐらいで Print に補完されるのではないかと思われる）を押すと、

```
Printer:  (default Local)
```

というプロンプトが表示される。ここで、CR を押すと、

```
Print command:  dvips -f ss-latex | lpr
```

というようなプロンプトが表示される。ここで、CR を押すと、印刷が行われる。[2]

11 ファイルの分割

大きな文書を編集する場合、文書全体を 1 つのファイルにしていると編集しづらいことがある。このため、LaTeX はファイルの分割機能を持っており、AUC-TeX もこれを支援するための機能を持っている。

LaTeX におけるファイル分割は、マスターファイルに `\input{`サブファイル名`}` という命令を挿入することで実現される。

こうすると、マスターファイルの上記命令がある位置に、サブファイル名で指定されたファイルの内容が挿入されたものと看做される。

AUC-TeX でファイルを開く時、最初に命令か環境を挿入する時、

```
Master file:  (default this file) カレントディレクトリ
```

というプロンプトが出るが、マスターファイルの場合、CR を打つ。また、サブファイルの場合、マスターファイル名を指定する。そうすると、サブファイルで C-c C-c しても指定したマスターファイルで C-c C-c したのと同じ効果が得られる。

12 相互参照

「第 3 章で述べたように」とか「表 5 に見えるように」というような、文書中の他の要素を参照するような表現を使うことがあるが、この種の表現は編集作業によって章や表の並びを移動させるとおかしくなってしまう。そこで、LaTeX は章や表などの番号にラベルを付けることで、番号に依存しない記述を可能にしている。

4 節では C-c C-s から始まる章の入力において、

```
What label:  sec:
```

というプロンプトが出る時、単に CR と押すことにしたが、ここで、なんらかの文字列を指定すると、それがその章に付けられたラベルとなる。例えば、test を指定すると

[2] なお、実際に印刷が行われるかどうかはプリンターの設定に依存する。

```
\section{test}
\label{sec:test}
```

が挿入される。この `\label{sec:test}` がラベルである。

このラベルは `\ref{sec:test}` によって参照できる。これは、命令の一種であるので、C-c CR を押し、\ref を指定することにより入力できる（この時、TAB 補完が利用できる）。

ラベルは、C-c C-e table CR で対話的に入力できる table （表）環境や C-c C-e figure CR で対話的に入力できる figure（図）環境でも指定できる。

相互参照機能を用いた場合、コンパイル後に

```
You should run LaTeX again to get references right, {number} pages.
```

という表示が出ることがあるが、その場合はもう一度 C-c C-c を押し再度コンパイルを行う。

また、コンパイル後に

```
LaTeX: there were unresolved references, {number} pages.
```

という表示が出た場合、存在しないラベルを参照していることを示している。

存在していないラベルを参照している箇所は、C-c C-l を押すと表示される LaTeX の実行結果ウィンドウの中で

LaTeX Warning: Reference ‘ラベル名’ on page ページ番号 undefined on input line 行番号.

のように表示されるので、これを参考にして修正する。

13 文献の参照

12 節で述べた文章中の他の要素に対する相互参照と同様に、他の文献に対する参照も論文などでは多用するものである。LaTeX では他の文献に対する参照は文章中での相互参照とは別の機能として実現されている。

LaTeX での文献参照（参考文献）支援機能は

- 文献リストも参照番号付けも人間が行う方法
- 文献リストは人間が作り、番号付けをコンピューターに任す方法
- 文献データベース (BibTeX) を用い、全てコンピューターに管理させる方法

があるが、ここでは BibTeX（実際にはそれを日本語化した jBibTeX）を用いる方法について述べる。

13.1 文献データベースの形式

BibTeX ないしは jBibTeX を用いる場合、予め BibTeX 形式の文献データベースファイルを作成しておく必要がある。なお、このファイルの拡張子が `.bib` とする必要がある。

この文献データベースファイルは次のようなものである：

```
@Article{NomuraMasaaki1984,
  author  =      {野村 雅昭},
  title   =      {同字と別字のあいだ},
  journal =      {日本語学},
  year    =      1984,
  volume  =      3,
  number  =      3
}
@Book{TEI-P4,
  editor       = {C. M. Sperberg-McQueen and Lou Burnard},
  title        = {TEI P4: Guidelines for Electronic Text Encoding
                  and Interchange
                  --- XML-compatible edition},
  publisher    = {University of Oxford},
  year         = {2002},
  organization = {TEI Consortium},
  key          = {TEI-P4}
}
@InBook{Mule-and-UTF-2000,
  author =       {錦見 美貴子 and 守岡 知彦
                  and 戸村 哲 and 半田 剣一 and 高橋 直人},
  title =        {bit 別冊「インターネット時代の文字コード」},
  chapter =      {第 9 章「文書編集系における文字コード」},
  publisher =    {共立出版},
  year =         {2001},
  series =       {bit 別冊}
}
@InCollection{fui,
  author =       {増井 俊之},
  title =        {適応 / 予測型テキスト編集システム},
  booktitle =    {インタラクティブシステムとソフトウェア II},
  pages =        {145--154},
  publisher =    {近代科学社},
  year =         {1994}
}
@InProceedings{NW,
  author =       {Yoshi Fujiwara and Yasuhiro Suzuki and Tomohiko Morioka},
  title =        {Network of Words},
  booktitle =    {Artificial Life and Robotics 2002},
  year =         {2002}
}
@Manual{SMPandSIP,
  title =        {Information technology ---
                  {Universal Multiple-Octet Coded Character Set (UCS)}
                  -- Part 2: Supplementary Planes},
  key =          {SMPandSIP},
  organization = {{International Organization for Standardization (ISO)}},
  year =         {2001},
  month =        Nov,
  note =         {ISO/IEC 10646-2:2001}
}
@Misc{IANA-charsets,
  author =       {{Internet Assigned Numbers Authority (IANA)}},
  title =        {CHARACTER SETS},
  howpublished = {ftp://ftp.iana.org/assignments/character-sets},
  note =         {charset の登記簿}
}
```

`@...{...}` で囲まれた範囲が 1 つの文献を表し、`@` から始まり { の前までの部分がその文献の種類を表す。この種類としては

@Article 定期刊行物（雑誌・論文誌）に掲載された論文や記事

@Book 出版社（出版主）のある書籍

@InBook 本の一部

@InCollection 本の一部で、それ自身に表題があるもの[3]

@Proceedings 学術会議の報告書

@InProceedings 学術会議の報告書に掲載された論文

@Manual 説明書

@Misc その他

などがある。この種類を表す識別子は大文字と小文字の区別はない。即ち、`@article` と書いても `@ARTICLE` と書いても構わない。

{ と } で囲まれた範囲は、最初に `fui,` のような文字列が現れ、その後、

```
author =        {増井 俊之}
```

のような『属性名 = 属性値』という形式の欄が並び、各欄の間は , で区切られる。この最初の , までの文字列は文献の参照名（ラベル）を表し、その後に並ぶ『属性名 = 属性値』という形式の欄は属性名で表現される属性を表す。必須となる属性欄は文献の種類によって異なる。また、属性欄の順番は自由である。

文献データベースファイルに書く文献の順番は自由であり、参照しない文献を書いても良い（実際に参照した文献だけがルールに従って並べられる）。よって、複数の論文や文書などで文献データベースファイルを使いまわすことが可能となる。

13.2 bibtex-mode

XEmacs CHISE (GNU Emacs) には BibTeX 形式の文献データベースファイルを編集するためのモードとして、`bibtex-mode` [1] というものがある。

XEmacs CHISE で拡張子が .bib であるファイルを開くと自動的に `bibtex-mode` に入る（例：C-x C-f test.bib CR）。これは新規にファイルを作る場合でも既存のファイルを開く場合でも同様である。

bibtex-mode には文献の種類に応じて文献情報のテンプレートを入力する機能がある。例えば、C-c C-e b を押すと

```
@Book{,
  ALTauthor =     {},
  ALTeditor =     {},
  title =         {},
```

[3] 上記の Mule-and-UTF-2000 は @InCollection を使うべきだったかも

```
  publisher =     {},
  year =          {},
  OPTkey =        {},
  OPTvolume =     {},
  OPTnumber =     {},
  OPTseries =     {},
  OPTaddress =    {},
  OPTedition =    {},
  OPTmonth =      {},
  OPTnote =       {},
  OPTannote =     {}
}
```

のような Book のテンプレートが入力される。@Book の場合、必須欄は author または editor と title, publisher, year であるが、選択的必須欄の前には ALT が付き、必須欄の前には何も付かず、必須でない欄の前には OPT が付く。そこで、それぞれの欄の { と } の間に必要な情報を書く。必要な欄を全部埋めた後 C-c C-c を押すと、欄の前の ALT や OPT や空欄が消されるなどして、適切に整形される。

テンプレートを入力するためのコマンドは C-c C-e b 以外にもさまざまなものが存在する。表 2 にその一覧を示す。

キー	文献の種類
C-c C-e B	@Booklet (@Book と似ているが publisher 欄がない)
C-c C-e I	@InBook (本の一部)
C-c C-e M	@Misc (その他)
C-c C-e P	@PhdThesis (博士論文)
C-c C-e b	@Book (（出版社（主）がある）書籍)
C-c C-e i	@InCollection (本の一部で、それ自身に表題があるもの)
C-c C-e m	@MastersThesis (修士論文)
C-c C-e p	@Proceedings (学術会議の報告書)
C-c C-e C-i or C-c C-e C-p	@InProceedings (学術会議の報告書に掲載された論文)
C-c C-e C-a	@Article (定期刊行物（雑誌・論文誌）に掲載された論文や記事)
C-c C-e C-b	@InBook (本の一部)
C-c C-e C-c	@InCollection (本の一部で、それ自身に表題があるもの)
C-c C-e C-m	@Manual (説明書)
C-c C-e C-s	@String
C-c C-e C-t	@TechReport (学校や研究機関で出す技報 (technical report))
C-c C-e C-u	@Unpublished (著者、表題はあるが公式な出版物でないもの)
C-c C-e M-p	@Preamble

表 2: bibtex-mode でのテンプレート入力用コマンド一覧

13.3 文献データベースの利用

BibTeX 形式の文献データベースファイルに書いた文献情報は、各文献エントリーの参照名を用い、\cite 命令を用いて参照できる。例えば、

```
増井 \cite{fui} は…
```

のようにする。

また、文書（マスターファイル）の最後に

```
\bibliography{m17n}
\bibliographystyle{jplain}
```

のようなものを入れる。ここで \bibliography 命令は使用する文献データベースファイルを指定するためのもので、\bibliography{m17n} というのは使用するファイルが m17n.bib であることを表している。複数の文献データベースファイルを指定することも可能で、例えば、coe21.bib と zinbun.bib を使用する場合、

```
\bibliography{coe21,zinbun}
```

とする。

また、\bibliographystyle 命令は文献リストの整形方法を指定するものである。jplain は標準的なスタイルを表す。他にも jalpha や jname などのスタイルがあり、自分でスタイルを作ることもできる (cf. [2])。

13.4 BibTeX を用いた場合のコンパイル

BibTeX による文献参照機能を用いた場合、コンパイル後に

```
You should run LaTeX again to get references right, {number} pages.
```

あるいは

```
LaTeX: there were unresolved citations, {number} pages.
```

という表示が出ることがあるが、その場合はもう一度 C-c C-c を押す。すると

```
Command:  (default BibTeX)
```

というプロンプトが出るが、その時に、jBibTeX（jB TAB で補完できるのではないかと思われる）と打ち、jBibTeX を起動する。

そして、再度 C-c C-c を押し、LaTeX を再び起動する。すると、

```
You should run LaTeX again to get references right, {number} pages.
```

と出るので、また C-c C-c を押し、LaTeX を起動する。

このようにして、

```
LaTeX: successfully formatted {number} pages.
```

という表示が出るまで、C-c C-c による処理を繰り返す。

13.5 まとめ

AUCTEX は XEmacs CHISE (GNU Emacs) の特徴の 1 つである補完機能をうまく活用した TEX 文書の編集のためのユーザーインターフェースを提供している。

その機能は大別すると

書式 C-c C-f

章・節 C-c C-s

命令 C-c C-m

環境 C-c C-e

実行 C-c C-c

に分けられる。書式関連機能を除く各機能では TAB 補完により可能な選択肢を表示可能な対話的なユーザーインターフェースを備えている。これにより、LATEX 機能の詳細を覚えていなくても比較的容易に入力作業を行うことができる。

XEmacs CHISE (GNU Emacs) はこの他にもスペルチェッカー機能（ispell, flyspell-mode 等）や辞書機能（lookup 等）や動的補完機能 (dabbrev) などが存在し、文書作成をさまざまな形で支援することができる。

LATEX や XEmacs CHISE は一見とっつきにくいように思われるかも知れないが、その操作をある程度覚えれば、これらが提供する文書処理のための機能を利用しワードプロセッサーを用いるよりも効率的にある程度大きな規模の文書を作成・編集することが容易であるといえる。皆様もその機能を活用して頂ければ幸いである。

参考文献

[1] 新堂安孝：BibTeX と bibtex-mode, reftex-mode の使い方, `http://www.fan.gr.jp/~ring/doc/bibtex.html` (2003).

[2] 奥村晴彦：文献スタイルファイル, LATEX2e 美文書作成入門 改訂第 3 版, 11.8 節, p. 170, 技術評論社 (2004).