

# TeX 利用のすすめ

家本 雄二<sup>1</sup>

## 1 はじめに

近年、ワープロ専用機（ワープロソフトを含む）の低価格化・高性能化にともない、広く一般に「美しい文書を…」という機運が高まっている。現在主流のワープロでは *WYSIWYG* (What You See Is What You Get: 文書の編集集中にディスプレイに表示されたイメージがそのままプリントアウトして得られるというもの) 方式と呼ばれるものが採用されており、これは

1. 出力結果をイメージしやすい (編集操作が直接ディスプレイ上に表現される)
2. セットアップが簡単 (ワープロを買ってきてすぐに使用できる)
3. 誰でもすぐに使えるようになる (操作が簡単)

などの特徴を持っているが、その反面

1. 各メーカーごとに独自の操作体系を持つ (操作体系が統一されていない)
2. 各メーカーあるいはワープロの機種ごとに文書ファイルに互換性がないことが多い
3. 複雑な数式や化学式、図などの挿入が困難 (最近では、これらをかなりのレベルでこなすことのできるワープロもある)

などの欠点もある。

### 1.1 TeX とは

これらワープロと対極にあるといわれるのが、TeX (テフあるいはテックと読む) と呼ばれる組版ソフト (印刷ソフト) である。TeX は、米国スタンフォード大学で開発された文書整形ソフトであり、その出力結果の美しさから多くの理科学系学生や技術者に論文やレポートの作成の為に利用されている。

ワープロが *WYSIWYG* 方式であるのに対して、TeX はマークアップ方式 (直接文書ファイル中に制御コードを書き込むというもの) を採用している。このため

<sup>1</sup>情報科学センター, iemoto@isct.kyutech.ac.jp

1. 編集中の文書からは、出力結果をイメージしにくい
2. 初心者にはとっつきにくい
3. コマンドを覚えない限りほとんどなにもできない

などという欠点を持つ。それでも  $\text{T}_\text{E}_\text{X}$  を愛用する人々が多いのは、この欠点を補って余りある  $\text{T}_\text{E}_\text{X}$  の素晴らしさゆえである。以下に、 $\text{T}_\text{E}_\text{X}$  の優れた特徴を記す。

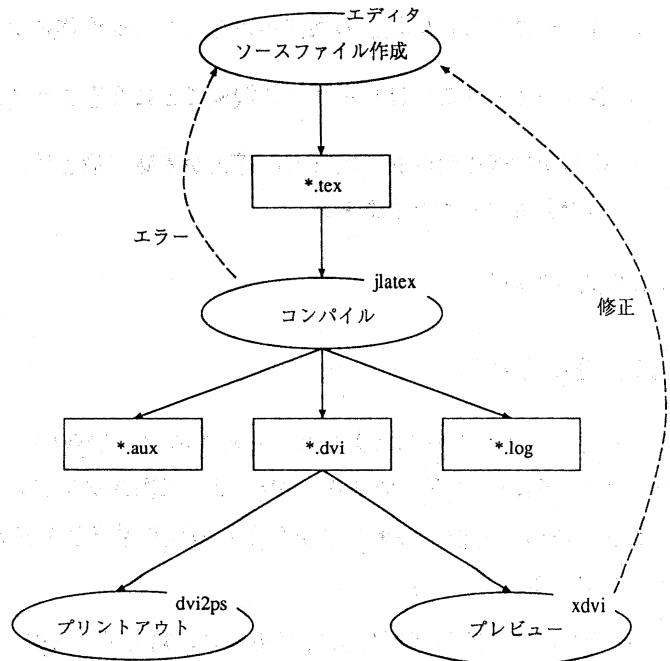
1. 日本語・英語など、様々な言語に対応したものが無料で配布されている
2. パソコンから大型機まで、多種多様なコンピュータで使える（標準化が進んでおり、どのコンピュータの  $\text{T}_\text{E}_\text{X}$  で処理しても、寸分の違いもなく出力できる）
3. 数式については、他のどんなシステムよりも自由度があり美しい出力が可能である
4. 多彩な組版規則が組み込まれており、利用者がそれを意識することなく利用できる

## 1.2 $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$

$\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$  は、 $\text{T}_\text{E}_\text{X}$  をベースに様々なマクロを組み込み機能強化を施した上、文書作成の手間を省く為に文書スタイルを予め用意して初心者にも扱いやすくしたものである（また、その他様々な  $\text{T}_\text{E}_\text{X}$  拡張マクロパッケージが発表されており、比較的簡単に入手できる。）。最近では  $\text{T}_\text{E}_\text{X}$  というよりも  $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$  ( $\text{jL}_\text{A}_\text{T}_\text{E}_\text{X}$ ) をさすことも多い。本文でも、これ以降は  $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$  を中心に解説を行う。

## 2 $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$ 利用概要

UNIX で  $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$  を利用する際の一般的な作業の流れを右に示す。これはどの企業・研究機関の UNIX システムにおいても同様である。しかし以降の事柄については、システムの種類・バージョンによって多少異なる場合もある。ここでは現在本学の研究用及び教育用 UNIX システムで利用されている  $\text{L}_\text{A}_\text{T}_\text{E}_\text{X}$  について解説する。



## 2.1 L<sup>A</sup>T<sub>E</sub>X ソースファイル作成

L<sup>A</sup>T<sub>E</sub>X には美しい出力を得る為の様々なコマンドが準備されており、これは \ (バックスラッシュ、ただし、端末によっては ¥) に続く半角英文字列として記述される。これらを駆使することにより自由自在に文書をレイアウトできる。ここでは、L<sup>A</sup>T<sub>E</sub>X コマンドについては他の文献を参照してもらうこととし、ソースファイル作成に関するもっとも基本的な部分についてのみ解説する。

L<sup>A</sup>T<sub>E</sub>X のソースファイルは次のような形で記述する。

```
\documentstyle[ < OPTION > ]{ < スタイル指定 > }
```

プリアンプル

```
\begin{document}
```

本文

```
\end{document}
```

ここで〈スタイル指定〉は、文書のおおまかな枠組を T<sub>E</sub>X に対して通知するものであり、自動的にいくつかのスタイルファイル<sup>2</sup>を取り込む。L<sup>A</sup>T<sub>E</sub>X は、これから文書整形の形式を判断するので適切なスタイル指定を行なうべきである。〈OPTION〉は省略可能である（各スタイル指定ごとにデフォルトのスタイルファイルが設定される）が、ここに明記することによりスタイル指定に対してさらに細かな指定ができる（スタイルファイルを取り込むことである）。これは複数指定することができ、カンマで区切って並べる。

〈スタイル指定〉

(スタイル名)	同日本語版	
book	jbook	本のような長い文書用
letter	jletter	手紙用
report	jreport	レポート形式

<sup>2</sup>レイアウトを規定するファイル (拡張子.sty)

*(OPTION)*

a4j	日本語 A4 サイズ
b4j	日本語 B4 サイズ
11pt	標準文字サイズを 11 ポイントとしてフォントを使用 (デフォルト値 10pt)
12pt	標準文字サイズを 12 ポイントとしてフォントを使用 (デフォルト値 10pt)
epsf	tgif などで描いた図 (EPS ファイル) を取り込む
twocolumns	左右二段組に出力する
twoside	見開き 2 ページの出力をする

**プリアンプル** はプリンタ出力の際に行間指定や余白の設定、レイアウトパラメータの変更など、文書全体に有効な設定を記述する場所である。また、自前のマクロコマンドやエリアスなどもここで指定することができる (本文中にも記述できるが、プリアンプルで記述することが望ましい)。この部分も省略可能である (デフォルトの設定が自動的に採用される)。

文書は必ず `\begin{document}` と `\end{document}` の 2 行の間に記述しなければならない。というよりも、**本文** は目的の文書記述の場所であるといえる。ややこしいようだがいくつかの注意点をさえておけば、ベタ書きの文章でもある程度は自動的に整形し出力してくれる。例えば、行揃えや改ページなどは見栄えの良いように TeX 側で調整し行なってくれる。

```

emacs @ hiko
% %の後ろはコメント行
\documentstyle[11pt]{jarticle} % 日本語の短文書用スタイルに設定

\begin{document}
\section{はじめに}

近年、ワープロ専用機 (ワープロソフトを含む) の低価格化・高性能化にともない、
広く一般に『美しい文書を $\cdots$ 』という機運が高まっている。現在主流のワー
プロでは WYSIWYG① (What You See Is What You Get : 文書の編集中にディスプレイに
表示されたイメージがそのままプリントアウトして得られるというもの)
方式と呼ばれるものが採用されており、これは $\cdots$ 
\hspace*{70mm} $\cdots$ 

\section{TeX とは}

これらワープロと対極にあるといわれるのが、TeX (テフあるいはテックと読む)
と呼ばれる組版ソフト (印刷ソフト) である。TeX は、米国スタンフォード大学で
開発された文書整形ソフトであり、その出力結果の美しさから多くの理科系学生や
技術者に論文やレポートの作成の為に利用されている。②
\hspace*{70mm} $\cdots$ 

\end{document}
----- Emacs: sample.tex (TeX: LaTeX) -----
Wrote /home/iemoto/ntex/work/sample.tex

```

図 1: TeX ソースファイルサンプル

## 2.2 dvi ファイルの生成

ソースファイル(.tex)は、次に示す `jlatex` コマンドでコンパイルし、dvi 形式と呼ばれる中間形式ファイルに変換する。

```
jlatex ファイル名
```

ソースファイルに誤りがなく、コンパイルが正常に終了すると、次の3つのファイルが生成される。

- \*.dvi : dvi 形式のファイル
- \*.log : コンパイル情報 (エラー情報) の詳細な記録
- \*.aux : セクションや式・図などの番号の情報を記録

log ファイルは画面に表示されるエラー表示よりも、詳しいエラー情報が得られる。これは、単なるテキストファイルであり、`cat` コマンドなどにより簡単に見ることができる。コンパイルに失敗した時 (この時にも log ファイルだけは生成される) には、このファイルを参照することでエラーの原因を突き止められる場合がある。aux ファイルは、セクションや式・図などの番号の情報を記録している。これは相互参照機能<sup>3</sup>と呼ばれるものを行う為に用いられる。L<sup>A</sup>T<sub>E</sub>X では、このファイルを自動的に生成・参照することができるので、簡単に図や表などの参照番号を文章中に記述することができる。

## 2.3 エラーとその対策

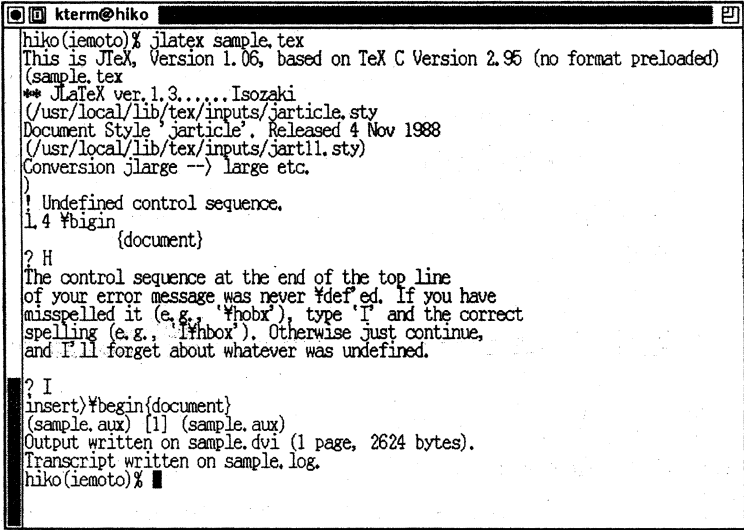
`jlatex` コマンドを実行すると、エラーメッセージ ( `1.n エラー文 <nはソースファイル中のエラー行数 >` の形式でエラー箇所を示す) が表示され、`?` マークのプロンプトになって処理が止まってしまうことがある。これは、設定かソースファイルに誤りがあることを意味する。`?` マークのプロンプトが表示された場合は、次のように対処する。

**RET** 現在示されているエラーを強制的に解釈し、処理を再開させる、いわゆる強制実行である。他にエラーがなければ dvi ファイルを生成するが、まだエラーがあれば、再び `?` のプロンプトとエラー内容を表示して止まる。

**R RET** 動作内容は **RET** の場合と同じだが、これ以降すべてのエラーに対して自動的に強制実行する。(Run without stopping)

<sup>3</sup>相互参照とは、文書の他の部分で現れた式番号や章番号を参照することである。

- Q **[RET]** 動作内容は R **[RET]** の場合と同じだが、エラーメッセージやコンパイル情報などの一切の表示を行わない。(run Quietly)
- H **[RET]** ソースファイルについて、エラーの原因を診断してくれる。時々的外れなメッセージを出してくることもあるが、まずまず信頼できるので、エラーがでた時はまず、これを実行してみると良い。(Help)
- I **[RET]** エラー行の L<sup>A</sup>T<sub>E</sub>X コマンドにタイプミスが見つかった場合などに、コマンドラインからタイプミスを修正する。ただし、ソースファイルを書き換えるものではないので、後からエディタで修正する必要がある。使い方の例を、図 2 に載せた。(Insert)
- E **[RET]** エディタを起動し、ソースファイルを読み込んだうえエラー行にカーソルを移動させる。これは、エディタを起動していない時には便利である。(Edit your file)
- 1 ~ 9 **[RET]** エラー行から n(1 ~ 9) 番目の行までを無視する。
- X **[RET]** jlatex コマンドを抜ける。dvi ファイルは生成されない。(quit)



```

kterm@hiko
hiko(iemoto)% jlatex sample.tex
This is JTeX, Version 1.06, based on TeX C Version 2.95 (no format preloaded)
(sample.tex)
** JLaTeX ver. 1.3.....Isozaki
(/usr/local/lib/tex/inputs/jarticle.sty
Document Style 'jarticle'. Released 4 Nov 1988
(/usr/local/lib/tex/inputs/jart11.sty)
Conversion jlarge --> large etc.
)
! Undefined control sequence.
l.4 \begin
      {document}
? H
The control sequence at the end of the top line
of your error message was never \def ed. If you have
misspelled it (e.g., '\hobx'), type 'I' and the correct
spelling (e.g., '\ifhbox'). Otherwise just continue,
and I'll forget about whatever was undefined.
? I
insert\begin{document}
(sample.aux) [1] (sample.aux)
Output written on sample.dvi (1 page, 2624 bytes).
Transcript written on sample.log.
hiko(iemoto)% █

```

図 2: L<sup>A</sup>T<sub>E</sub>X エラーとその修正例

## 2.4 プレビューアの利用法

グラフィック端末で T<sub>E</sub>X の処理結果を確認する為には、dvi ファイルを作成した後、プレビューアと呼ばれるツールを用いる。プレビューアには様々な種類のものがあるが、ここでは、xdvi コマンドについて説明する。

xdvi [ OPTION ] dvi ファイル名

以下に、xdvi コマンドの代表的オプションを示す。また、この他にも多彩なオプションが存在するが、それらは省略する。 (OPTION)

- +page page ページ目を最初に表示する。
- s shrink ウィンドウに完全に表示できる最大画面の設定 ( $1 \leq shrink \leq 4$  デフォルト値3)。
- S density プレビュー画面の文字濃度を指定する (デフォルト値40)。

xdvi を起動すると、図 3 のようなウィンドウが開く。ウィンドウ右側に並ぶボタンをマウスでクリックすることにより、次のような操作を行なうことができる。

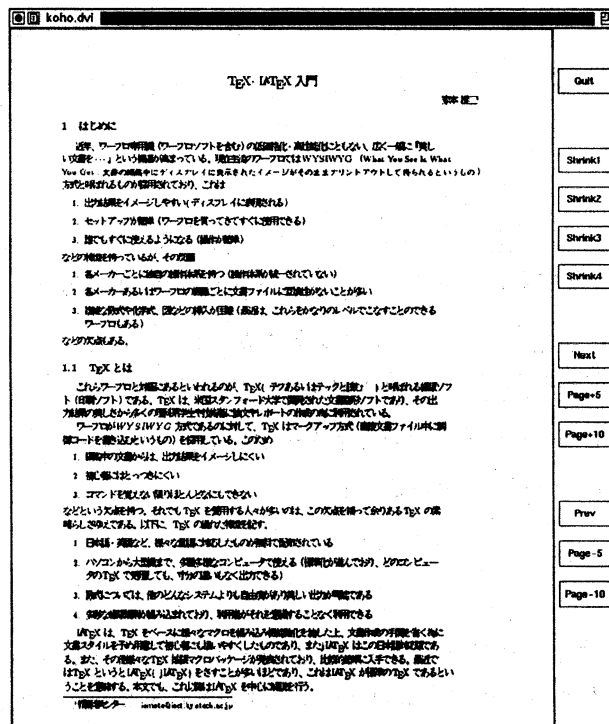


図 3: xdvi 起動画面

**Quit** プレビューを終了する。

**Shrink $n$**  画面の拡大率変更 ( $n$  が小さいほど高倍率) . ( $1 \leq n \leq 4$ )

**Next** 次ページを表示する。

**Page+n**  $n$  ページ先を表示する。 ( $n = 5, 10$ )

**Prev** 前ページを表示する。

**Page-n**  $n$  ページ前を表示する。 ( $n = 5, 10$ )

## 2.5 プリントアウト

プリントアウトは `dvi2ps` コマンドを用いる。ただ、実際には `dvi2ps` コマンドは `dvi` 形式を Post Script 形式に変換するコンバータであり、また結果は標準出力に出力されるので、直接プリントアウトする場合には①、PS ファイルとして出力する場合には②のようにする。

① プリンタに出力      `% dvi2ps [ OPTION ] dvi ファイル名 | lpr`

② PS ファイルに出力    `% dvi2ps [ .OPTION ] dvi ファイル名 > PS ファイル名`

### OPTION

-f 出力開始ページ数 プリントアウトする文書について、何ページ目から出力するかを指定する。デフォルトでは1ページ目から出力する。

-t 出力終了ページ数 プリントアウトする文書について、何ページ目まで出力するかを指定する。デフォルトでは最終ページまで出力する。

-r 出力する順序を逆順にするオプション。このオプションを指定するとページの小さい方から出力される。

## 2.6 おわりに

以上、情報科学センターにおける  $\text{T}_{\text{E}}\text{X}$  の使い方を簡単に説明した。 $\text{T}_{\text{E}}\text{X}$  自体の機能については紙面の都合により触れることができなかった。市販の参考書などと合わせて  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  を使ってみてはどうだろうか？