

センターにおける L^AT_EX による文書作成環境

総合情報処理センター
鶴 正人

E-mail: tsuru@net.nagasaki-u.ac.jp

T_EX 及び L^AT_EX は、近年日本でも広く使われている、世界共通の“文書作成環境”であり、研究者/技術者を中心に、論文やレポート、さらに出版物の作成にも使われている。

本記事では、総合情報処理センター（以下、センター）が UNIX ワークステーションで提供している、ASCII 日本語 T_EX 上の L^AT_EX を中心に文書を作成する環境を概説する。細かい文法やオペレーションは、それぞれの解説書やマニュアルを参照されたい。今後 L^AT_EX の利用を考えられている方の参考になれば幸いである。

目次

1	文書作成/整形用システム L ^A T _E X(T _E X) の特徴	102
2	センターにおける、L ^A T _E X 利用の基本	103
2.1	基本コマンド	103
2.2	ドキュメントスタイル	105
3	グラフや絵の作成と L ^A T _E X 文書への取り込み	106
3.1	L ^A T _E X 文書への取り込み	106
3.2	グラフ化ツール	107
3.3	作画ツール	107
3.4	グラフィックスライブラリ	109
3.5	画像処理ツールと組合せ	109
4	参考文献データベースの活用	110
5	その他、いろいろあります	112
5.1	論文投稿用	112
5.2	OHP やポスター	112
5.3	グラフ的構造	113
5.4	プログラミング言語等のソースファイル	114
5.5	texinfo 形式のマニュアル	116
5.6	HTML 形式への変換	116

1 文書作成/整形用システム L^AT_EX(T_EX) の特徴

T_EX は、Stanford 大学の Donald Knuth 教授が開発した、文書整形/組版用システムであり、L^AT_EX は、Leslie Lamport 氏 [1] が開発した、文書作成の観点から使いやすくした(逆に自由度も減らした) T_EX 上のマクロパッケージである。具体的にいえば、美しく見やすい文書を簡単に作成するための記述言語であり、またその言語の命令を埋め込んで書かれた文書を、実際に表示したり印刷したりするためのシステム(一連のコマンドやフォントデータ等々)である^{†1}。

多彩な字体やレイアウト、数式や記号の美しい印刷、表の作成、さらに他のツールで書いた絵やグラフの取り込み等(図 1、図 2 参照)、の面では、いわゆる“高級ワープロ”と似たようなものであるが、その実現方法が異なり、

- 普通のエディタで普通の UNIX や MS-DOS 上のテキストファイルとして文書を書き、
- その時に文書中に、整形/組版のための指示(命令)を書き込む。

という方法(Mark-up 言語方式)を取る。

<p>数式モード:</p> <p>普通の文中に混ぜるには、\$で囲む。 例えば、$\sum_{i=1}^n \int_0^{\infty} f_i(x) dx$ そこだけ、別の行で表示するには、\[と\]で囲む。</p> $\sum_{i=1}^n \int_0^{\infty} f_i(x) dx$	<p>表の例:</p> <p style="text-align: center;">—主な箇条書きの種類—</p> <table border="1"><thead><tr><th>命令</th><th>意味</th></tr></thead><tbody><tr><td>itemize</td><td>印をつける</td></tr><tr><td>enumerate</td><td>通し番号をつける</td></tr><tr><td>description</td><td>キーワードをつける</td></tr></tbody></table>	命令	意味	itemize	印をつける	enumerate	通し番号をつける	description	キーワードをつける
命令	意味								
itemize	印をつける								
enumerate	通し番号をつける								
description	キーワードをつける								

図 1 L^AT_EX の機能例

つまり、最終的な見たまを画面上で編集し、視覚的にデザインしていく(What You See Is What You Get 方式)のとは違い、論理的にデザインして書いたものを“コンパイル”して、最終結果物を生成(視覚化)する必要がある。すなわち、本質的特徴は、

文書の論理的構造を編集できる。段落、章立て、箇条書や表、グラフ等の論理的構造を指定するだけで、本にするのか小レポートなのか等に応じて、適切な物理的/視覚的表現^{†2}、さらには、目次の生成や参照関係の解決等々まで自動的にやってくれる。

であり、少し複雑な文書を書く場合、特に、再編集や再利用の観点からは、現状の市販の“ワープロ”よりはるかに効率がよい^{†3}。その他にも、

- L^AT_EX の文書(ソース)ファイルは通常の UNIX や MS-DOS のテキストファイルなので、一般のテキストデータをそのまま使え^{†4}、またスクリプト(バッチ)機能と組み合

^{†1}日本語対応したシステムには、(株)アスキーで開発された ASCII 日本語 T_EX と、NTT ソフトウェア研で開発された NTT J_TE_X があるが、本記事ではセンターで利用できる ASCII 版に限定する。

^{†2}配置、字の大きさや字体(フォント)、字間や行間の調整等

^{†3}本記事も、もちろん、L^AT_EX を使って書いている

^{†4}パソコン上の慣れたエディタでテキストを書き、それを UNIX 上の L^AT_EX システムに食わせることも、学内 LAN の整備やパソコンの(マルチ)ウィンドウ化が進んだ現在では、簡単にできる。

わせて、いろいろな自動化も可能である。

- スタイルファイル (2.2節参照) を作る (あるいは他人が作ったのを使う) ことで、定型文書の生成が簡単にできる。例えば、科研費 (文部省科学研究費) の申請書作成用スタイルファイルがあり^{†5}、昔はワープロで書いて手で切り貼りする人が多かったと思われるが、これを使って簡単にオーバーレイ印刷できる。
- 素晴らしいことに、無料 (配布自由) のシステムであり、パソコン、ワークステーションから大型計算機まで、各種マシンに移植されている。そして、賛同する世界中の有志によって^{†6}、次々と機能の拡張や関連したツールの開発が行なわれている [2]^{†7}。

具体的な使い方 (文法等) は、市販の入門書がたくさん出ている ([1],[3],[4]) し、上記の [2] や [5]^{†8}、あるいは、昨年のセンターレポートの“超入門”記事 [6] 等を参照されたい^{†9}。

なお、本記事で触れるソフトウェアは多くの有志の方々が作られたフリーウェアであるが、いちいち作者は示さない。また、以下の説明は、センターが提供している 3 種類の UNIX ユーザ環境 (研究用、教育用、NUNet 用) に共通である。

2 センターにおける、 \LaTeX 利用の基本

2.1 基本コマンド

センターの UNIX システム上で \LaTeX を利用する時の基本的コマンドは、

<code>latex</code>	\LaTeX の文書 (ソース) ファイルをコンパイルし、DVI ファイルへ。ASCII 日本語 <code>TeX1.7j</code> 。
<code>dvi2ps</code>	DVI ファイルを Postscript ファイルへ。日本語化 <code>dvi2ps-1.91j</code> 。
<code>xdvi</code>	DVI ファイルを X-Window 端末で見る。 <code>xdvi-17+miyu.patch</code> 。
<code>ghostview</code>	Postscript ファイルを X-Window 端末で見る。日本語化 <code>gs261+ ghostview-1.4.1</code> 。
<code>dvi2tty</code>	DVI ファイルをテキスト (文字) 端末で見る。日本語化 <code>dvi2tty-5.0</code> 。

であり、これらを使う手順の概略は以下のようになる^{†10}。

なお、以下では、UNIX のプロンプト (入力を促すシステムからの表示) を %、リターンキーの入力を \square で表すことにする。

また、本記事では、UNIX 上の実行コマンドのことを“コマンド”、 \LaTeX の中に記述する指示 (命令) 語のことを“命令”と呼んで区別することにする。

^{†5}94 年度版は、センターの `ftp:pub/prg/TeX/macros/Kaken94` にも置いてある。

^{†6}例えば、`TeXUsers Group` という世界的組織があり、インターネットを通して各種ツールを収集/配布している。

^{†7}[2] の記事の元になった資料は、センターの `ftp:pub/doc/TeX/styleuse` にも置いている。もちろん、雑誌連載の方が内容が詳しく新しい。

^{†8}50 ページほどのよくまとまった (無料の!!) \LaTeX 入門であり、センターの `ftp:pub/doc/TeX/hiro-latex` 下にも置いている。Postscript 形式のファイル (`latex.ps`) を、後述の `ghostview` コマンドを使って読めるし、`lpr` コマンドを使ってセンター設置のポストスクリプトプリンタ等に自由に印刷出力してよい。

^{†9}逆に、英語版 `TeX` (\LaTeX) のことは良く知っていて、ASCII 日本語 `TeX` の詳細を知りたいという方は、センターの `ftp:pub/doc/TeX/asciiJTeX` に DVI 形式でドキュメントが置いてある。

^{†10}コマンドの詳細な使い方は、`latex` コマンド以外は、すべて、`man` コマンドでオンラインマニュアルを見ることができる。

1. 普通のテキストエディタを使って、文書ファイルを作成する(書く)。この時、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ の文法に従って命令を埋め込む。このファイルは、必ず、`.tex` で終るようなファイル名にする。以下このファイルを、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 文書ファイルと呼ぶ^{†11}。

例えば、`sample.tex` という名前だとする。

2. `latex` コマンドによるコンパイル (`dvi` ファイルの生成)

```
% latex sample ▽
This is BigTeX, C Version 2.99 - j1.7e (no format preloaded)
...
Output written on sample.dvi (4 pages, 20212 bytes).
```

メッセージが一杯であるが、成功すれば `sample.dvi` というファイルができる。しかし、文法エラーが見つかったと、以下のようにそこで止まり、`?` が表示される。そこで、`X`(大文字 `X`) を入力してこれを終了し、また、エディタで文書ファイル (`sample.tex`) のエラー箇所を修正しなければならない^{†12}。

```
LaTeX error. See LaTeX manual for explanation.
! Something's wrong--perhaps a missing \item.
1.77 ..... <--- 間違いが見つかった行番号
....
? X ▽
```

なお、正常に終了したように見えても、

```
LaTeX Warning: Label(s) may have changed. Rerun to get cross-references
right.
```

のようなメッセージが含まれていれば、参照が解決していないので、もう一度 `latex` コマンドを繰り返す必要がある。

3. `dvi` ファイルを Postscript 形式ファイルに変換。変換結果は標準出力に出るので、UNIX のリダイレクションを使ってファイルへ落す。

```
% dvi2ps sample > sample.ps ▽
```

ただし、`sample.ps` というファイルが既に存在し、それを上書きしたい場合は、

```
% dvi2ps sample >! sample.ps ▽
```

4. 印刷する以前に中身(刷り上がり)を確認する。`dvi` ファイルを X-Window の画面で表示するには、

```
% xdvi sample ▽
```

Postscript ファイルに変換した後で、中身を X-Window の画面で表示するには^{†13}、

^{†11} 文書ファイルの作成は必ずしも UNIX で行なわなくてもよいが、UNIX 上のテキストエディタとしては、有名な `emacs`、UNIX 標準の `vi`、そして、`emacs` 風キーアサインの簡易エディタ `tiny` が使える。

^{†12} この例は、“`item` 命令が抜けてるらしい、77 行目らしい”とメッセージが出ているので、それを参考に間違いを探す。

^{†13} Postscript 形式のファイルは、 $\text{T}_{\text{E}}\text{X}$ とは関係なしに広く使われており、`ghostview` は Postscript ファイルを表示する一般的コマンドである。

```
% ghostview sample.ps ▽
```

これらは、マウスを使ったメニュー方式で操作する。前者の方が反応が速いが、後者の方が実際の印刷イメージに近い。前者はメイン window の Quit ボタンを、後者はメイン window の File メニューの中の、Quit ボタンをクリックすると終了する。

5. Postscript ファイルをプリンタに出力 (印刷) する。これは、通常のテキストファイルと同じく、lpr コマンドでよい。例えば、センター 1 階の入出力室の opena という名前のプリンタへ出すには、

```
% lpr -Popena sample.ps ▽
```

```
% lpq -Popena ▽
```

lpr コマンドで出力要求後、必ず、lpq コマンドで確認する。

6. DVI ファイルの中身をテキスト端末でみるには、

```
% dvi2tty sample ▽
```

当然、きれいには見えない。文字 q の入力で終了。DVI ファイルをテキストファイルへ (曲がりなりに) 変換する目的にも使える。

```
% dvi2tty sample > sample.txt ▽
```

2.2 ドキュメントスタイル

L^AT_EX で最初を知る必要がある命令が、documentstyle 命令である。

- その L^AT_EX 文書の、全体の文書形式 (紙や文字や空白の大きさ、用途に応じた章節等の構成レイアウト等) を決める
- その L^AT_EX 文書の中で、特別な機能を使える状態にする (命令や変数の定義等)

documentstyle 命令は、documentstyle{主スタイルファイル指定} や documentstyle[オプションの並び]{主スタイルファイル指定} のように使い、例えば、

```
\documentstyle[epsf,egtrees,fancybox,12pt,a4j]{jarticle}
```

は、主スタイルファイル jarticle.sty を指定している。一般にスタイルファイルとは、ある特定の形式/機能のための、長さ等の各種設定値やマクロ (組合せ/拡張命令) の定義をまとめたファイルで、実際のファイル名は .sty で終わるが、documentstyle 命令で指定する時は、.sty は省略する。L^AT_EX 文書の全体形式は、基本的にこの“主スタイルファイル”で決まる。

オプションは、主スタイルファイルに渡され、別のスタイルファイルの読み込み指示等になる。この例の epsf は、3.1 節の画像ファイルの取り込み機能を定義し、12pt は基準の文字の大きさを 12 ポイントにする。

標準的“主スタイルファイル”の例として、

article, jarticle - 普通の記事 (英文用、和文用)

report, jreport - 長いレポート

book, jbook - 本

代表的“オプション”の例として、

- 11pt, 12pt – 基準の文字の大きさ。1 ポイントは約 0.35 mm。
- a4, a4j, a5j, b4j, b5j – A4, B4 等の用紙サイズ^{†14}。
- twocolumn – 2 段組
- titlepage – 表題を独立した 1 ページ (表紙)

3 グラフや絵の作成と L^AT_EX 文書への取り込み

表は、L^AT_EX の機能 (tabular 命令等) で簡単に書けるが、グラフや絵は、別のツールで作って取り込む方が簡単なので、以下で、センターの UNIX 上での、

- 数値データや数式 (関数) をグラフ化する方法
- 対話的に絵や図形を描く方法
- プログラミングで図形を描く方法
- イメージスキャナ等で生成/抽出した画像データ等も含め、それらを組合せて加工する方法
- それらを L^AT_EX 文書ファイル内に取り込む方法

を概観する。

3.1 L^AT_EX 文書への取り込み

グラフ、絵、画像等は、Postscript 形式のうちでも特に Encapsulated Postscript (EPS) と呼ばれる形式になっていれば、epsfile 命令^{†15}を使って簡単に文書中に取り込める。

以下に紹介する作画ツールをはじめ、世の中のグラフィックス表示機能を持つ各種アプリケーション、例えば、数式/統計解析ツール^{†16}、マルチメディア情報表示ツール等の多くは、EPS 形式での出力をサポートしているので、それらを L^AT_EX 文書内に取り込める。例えば、図 2 は、3.2 節で述べる方法でグラフ表示を gnuplot.ps という名前のファイルに出力し、それを、以下のように記述して取り込んだ。

```
\begin{figure*}[htpb]
\begin{center}
\epsfile{file=gnuplot.ps,width=0.62\hsize}
\caption{gnuplot による 3 次元グラフ}
\label{fig:gnuplot}
\end{center}
\end{figure*}
```

なお、カラーの Postscript も扱えるので、色ページ入り L^AT_EX 文書を作成でき、“カラー Postscript プリンタ”^{†17}さえあれば、簡単に出力 (印刷) できる。

^{†14}この指定はあくまで用紙向きのレイアウトを計算するだけであり、印刷時に、B4 等の用紙 (トレイ) をサポートしている Postscript プリンタで、実際にその機能を使用するには、dvi2ps コマンドでファイルを作成する時点で、-o パラメタで用紙サイズを指定する必要がある。

^{†15}ただし epsfile 命令を使うには、先頭の documentstyle 命令で epsf オプション指定が必要。2.2 節参照。

^{†16}例えば Mathematica では、附属の psfix コマンドを併用し EPS 形式にする。

^{†17}センターでは、富士 Xerox 社の Acolor636 プリンタを (有料で) 利用できる。

3.2 グラフ化ツール

数値データや数式(関数)をグラフ化するには、gnuplot コマンドが使える。gnuplot というコマンド名で起動すると、メッセージが出た後、内部プロンプト `gnuplot>` が表示されるので、この状態で `plot` 等のいろんなサブコマンドを入力する。

- 関数のグラフ表示：指定した関数(2次元の場合、 $y = x^2$ とか $y = \sin(x)$ とか)のグラフを表示する。例えば、

```
gnuplot> plot [-5:5] sin(x) ▽  
gnuplot> plot x**2,x**3 ▽
```

は、 $y = \sin(x)$ の2次元グラフで表示範囲を $-5 \leq x \leq 5$ としたもの、 $y = x^2$ と $y = x^3$ を重ねたグラフ、をそれぞれ表示する。

- 数値データのグラフ表示：一行に一点分のデータ(2次元の場合、X座標とY座標の値)を書いたファイルを読み、グラフ表示できる。データファイル名を仮に `sample` として、

```
gnuplot> plot 'sample' with line ▽
```

で折れ線表示する。

- EPSファイルへの保存：例えば、

```
gnuplot> set view 60,120 ▽  
gnuplot> set title "Imaginary part of complex square root function" ▽  
gnuplot> set terminal postscript portrait ▽  
gnuplot> set output "gnuplot.ps" ▽  
gnuplot> splot [-3:3] [-1:1] x**2-y**2,2*x*y,y ▽
```

で、`splot` の結果(表示)が、画面ではなく、指定したファイルへEPS形式で保存できる。また、同様に

```
gnuplot> set terminal tgif ▽
```

としておけば、次節の `tgif-obj` 形式で保存できるので、`tgif` で加工できる。

3.3 作画ツール

X-Window 上での作画には、`draw` 系の `tgif` コマンドと `paint` 系の `xpaint` コマンドが使える。どちらもカラー対応で、(できるものには、、)拡大/縮小/回転が可能である。

使い方は、そのコマンド名で起動すると、メイン window が現れるので、以降、マウスを使ったメニュー方式で操作する。図3参照。

- `tgif` (`tgif-2.14.8` 日本語対応版)

図形エディタと呼ばれるもので、線画/図形を基本とし、高度なアレンジ機能(部品を登録し組合わせて使う等)を持つ。また、`kinput2` を使った日本語入力ができる^{†18}。

入力ファイル形式は、`tgif-obj` と呼ばれる図形データだが、EPS、XBM(X11 bitmap)、XPM(X11 pixmap)の画像も“import”できる。ただし、importされた画像は指定した位置の張り付けることができるだけで、図形オブジェクトに対するような各種加工操作は適用されない。保存(出力)ファイル形式は、再加工可能なデータとしては、`tgif-obj` 形式で保存するが、表示のためには、EPS、XBM等の形式で出力可能。

^{†18}CTRL+SPACE でかな漢字変換モードへ。Shift+SPACE で戻る。

Imaginary part of complex square root function

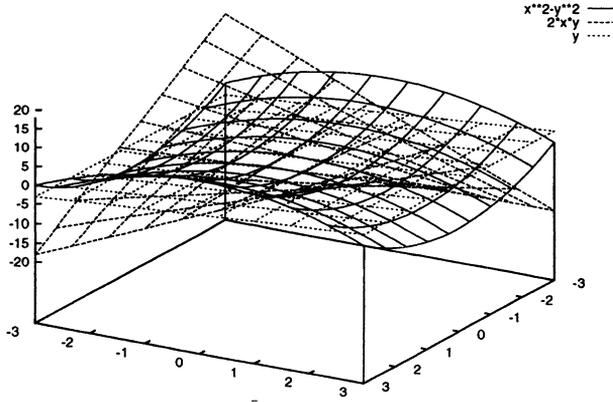


図 2 gnuplot による 3 次元グラフ

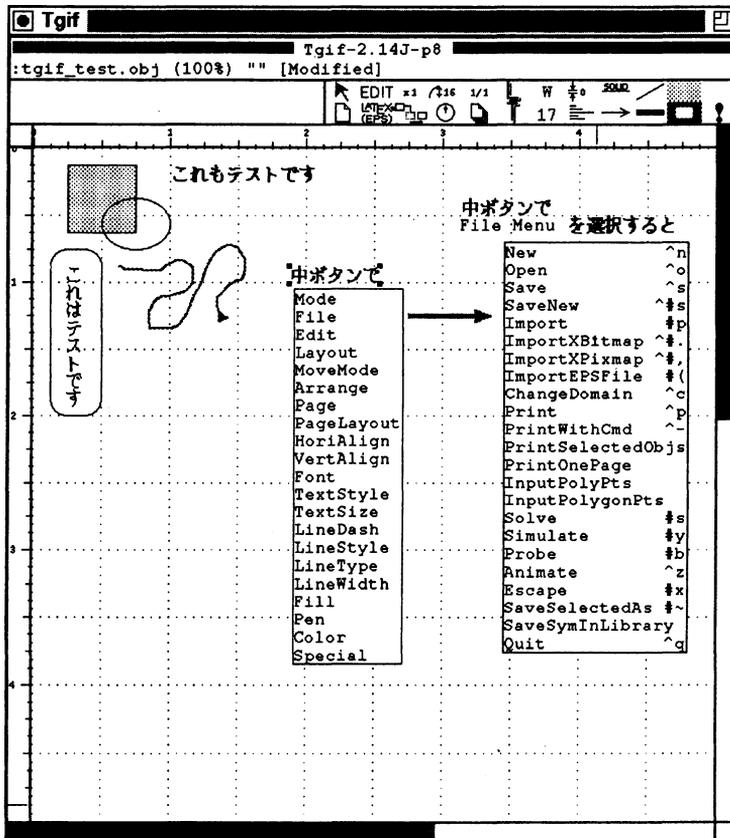


図 3 tgif

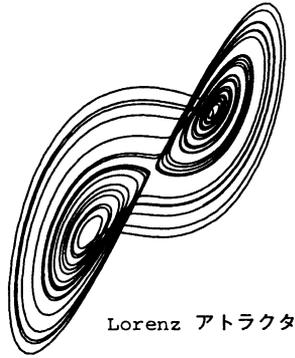


図 4 XeasyGraphic による 2 次元グラフ

- xpaint (xpaint-2.1)
 ペイントツールと呼ばれるもので、フリーハンドを基本とし、パソコン系のペイントブラシと同等の機能だそうである。スプレー (吹き付け) もできる。
 入力ファイル形式は、TIFF, PPM, GIF, XBM, XPM, XWD 等の画像で、保存 (出力) ファイル形式は、それに EPS が加わる。
- EPS ファイルへの保存
 tgif の場合、メイン window 上部にある出力先形式の表示を \LaTeX (EPS) にしておいて、File メニューの中の Print をクリックする。
 xpaint の場合、対象キャンバス window の File メニューの中の *Save as* を選び、その中で、出力形式を選択する。

3.4 グラフィックスライブラリ

グラフィックスライブラリ^{†19}として、教育目的で開発された、フリーウェアの Xeasy-Graphic(10 個の基本描画関数から成るライブラリ) が利用できる。コンパイルには gcc を使い、XeasyGraphics.h を include し、XEG ライブラリを X11 ライブラリと共に指定する^{†20}。EPS 形式でのファイル出力関数をサポートしており、3.1 節の通り、 \LaTeX に張り込める。図 4。

3.5 画像処理ツールと組合せ

3.3 節の作画ツールは、多様な形式のファイルを読み込めるので、イメージスキャナ^{†21}で取り込んだ写真にコメントを添えたり、他のパソコン系のツールでつくった絵/図を再利用

^{†19}C 言語 (や FORTRAN) のプログラムの中でウィンドウやある形式のファイルへ図を描くためのサブルーティン群。GKS, PHIGS 等が有名。

^{†20}例えば、`gcc xxx.c -lXEG -lX11 -o xxx.exe` のように指定。実際の置場所は、ヘッダファイルは `/usr/local/include` に、ライブラリは `/usr/local/X11R5/lib` にある。使い方のドキュメントやサンプルは、`ftp:pub/doc/graphics/lib/xeg` に置いてある。

^{†21}センターでは、富士 Xerox 社の Acolor636⁶ が、カラーイメージスキャナとして利用できる。

したりもできる。そして、最終的に EPS 形式で保存して、3.1節の通り、 \LaTeX 文書に取り込めばよい。

この変換 (加工) 処理に関しては、さらに強力なツールもある。

- netpbm

画像ファイルの形式変換のための数十個の単機能のコマンド群である pbmplus の改良版^{†22}。PBM, PGM, PPM, PNM という中間形式を介して、GIF, TIFF から、MacPaint, Macintosh PICT, XBM 等まで多様な形式の変換を行なう [7]。

- pstoeit

前節の作画ツールも、Postscript 形式のファイルを読み、“加工”することはできない。そこで、制限つきではあるが、Postscript ファイルを tgif-obj 形式に変換 (翻訳) するのが、pstoedit コマンドである。

- xv

version 3 は、多様な形式 (TIFF, GIF, JPEG, BMP, XBM, PBM, IRIS RGB, SUN Raster 等を読み込み、保存はさらに EPS が加わる)、拡大/縮小/回転/切りだし、色や輝度の変更、等々が可能である [7] が、シェアウェア^{†23}であり、今の所は利用できない。

具体例として、X-Window の画面イメージに、tgif で注釈をつけて最終的に EPS 形式ファイルに保存するには、

1. xwd コマンドで、X-Window の画面 (特定 window) を XWD(X11 Window Dump) 形式のファイルに落す。xwd コマンドを打ったら、マウスポインタが + 印に変わるので、取りたい window 上まで移動させ、クリックする。

```
% xwd -frame -screen > my.xwd ▽
```

2. XWD 形式を一度、EPS 形式まで変換。

```
% xwdtopnm my.xwd | pnmdepth 255 | ppmtopgm | pnmtops -noturn > my.ps ▽
```

3. my.ps を tgif の中の File メニューの中の Import ボタンで作画領域へ取り込み、線や文字で注釈をつけ、3.3節の方法で、全体を EPS 形式ファイルに出力する。

あるいは、対象 window が小さいものなら、直接 X-Window の画面上で、起動した tgif の作画領域内に重ね、外側に線や文字で注釈をつけ、その画面イメージを上と同様に (xwd と xwdtopnm 等を使って)EPS 形式まで変換すればよい。

4 参考文献データベースの活用

\LaTeX における参考文献欄作成は、一般には thebibliography 命令でラベルのついた文献情報を定義し、本文の中からは、そのラベルを cite 命令で参照する。しかし、毎回、多量の文献情報を \LaTeX 文書ファイル中に記述するのは無駄である上、投稿先によって、文献情報の表示形式を変える必要もある。そこで、あらかじめ自分用参考文献データベースを

^{†22}man pbm pnm pgm ppm で概要が掴める:-)。

^{†23}常用する場合は、使用料の寄付が求められている。

作成しておけば、BIBTEX というシステムを使って、そこから、文書中で引用されているものの情報を探し、適切な形式で参考文献欄を自動生成することができる。センターでは、jbibtex というコマンド名 (jbibtex-3.1) で利用できる。使い方は、

1. 参考文献データベースを必ず、.bib で終るようなファイル名で作る。中の参考文献レコードは、文献の形式 (論文, 本など) を指定した後に、引用キーや書誌項目を “キーワード=値” の形でカンマで区切って並べる形で記述する^{†24}。
2. L^AT_EX 文書ファイル内で、bibliography 命令で 上記データベースファイル名を指定する。そのファイルを探すディレクトリは、何もしなければ、カレントディレクトリであるが、環境変数 BIBINPUTS でも指定できる。
3. bibliographystyle 命令で表示形式を指定する。xxxx.bst というファイルに定義されている場合、\bibliographystyle{xxxx} と指定する。センターで用意されているのは、abbrv, jalpha, jorsj, plain, unsrt, alpha, jipsj, jplain, tieice, jssst, jabbrv, jname, junsrt, tipsj があり、例えば、tieice は電子通信学会論文誌、jssst はソフトウェア科学会和文誌の規定の参考文献用である^{†25}。それ以外のものを用意して使う場合、そのファイルを探すディレクトリは、環境変数 TEXINPUTS に従う。センター標準のログイン環境では既に設定されているが、必要なら、自分で (setenv コマンドで) 設定しなおせば良い。例えば、本記事の最後は、以下のようになっており、ファイル tex.bib から参考文献情報を探す。なお、tex.bib がカレントディレクトリ以外の場所に置かれているなら、jbibtex の実行前に、環境変数 BIBINPUTS にそのディレクトリを設定する。

```
\begin{document}
.....
\bibliographystyle{junsrt}
\bibliography{tex}
\end{document}
```

4. 文書ファイル (xxx.tex) を一度 latex コマンドにかけ、aux ファイル (xxx.aux) を生成する。
5. 次に jbibtex コマンドで、bbl ファイル (xxx.bbl) を生成する。
6. 再度、latex コマンドを、参照解決するまで (2回) かける。つまり、

```
% latex xxx ▽
% jbibtex xxx ▽
% latex xxx ▽
% latex xxx ▽
```

とすると、最終的な xxx.dvi ができる。

7. なお、jbibtex でエラーになった場合 (例えば、bib データベースがおかしく)、xxx.bbl というファイルを消してから、最初の latex からやり直す。

^{†24} 作り方のドキュメントやサンプルは、センターの ftp:pub/doc/TeX/jbibtex にも置いてある。

^{†25} 5.1節のスタイルファイルと併用する。

5 その他、いろいろあります

スタイルファイルは、標準以外にも世界中で作られた優れ物が多数あるが、以下では、センターで利用できるスタイルファイルや L^AT_EX 関連コマンドの中のいくつかを紹介する^{†26}。

以下の例の詳しい使い方は、

- スタイルファイルに関しては、[2] に載っているが、センターのスタイルファイル置場^{†27} からそのスタイルファイルまたはそのサンプルを見つけて中身を読むこともできる。だいたい、スタイルファイルは、適切なサンプルがあれば、それを latex にかけて印刷してみれば、使い方はほぼわかる。また、スタイルファイル自身の中に、コメントとして使い方やサンプルが書かれている場合も多い。
- コマンドに関しては、man コマンドでマニュアルが引ける。また詳しい情報がある場合は脚注等にポインタを与えた。

5.1 論文投稿用

単なる 2 段組というなら、twocolumn オプション (2.2 節) でもよいが、学会誌等の投稿論文には、かなり細かい形式の指定がある。そこで、最近では、それ専用のスタイルファイルを (公式または非公式に) 公開している場合も多い。また、シンポジウム/ワークショップ等は proceedings の編集時間の短縮のため、L^AT_EX および統一スタイルファイルの利用がますます一般化すると予想される。例えば、ソフトウェア科学会和文誌の場合、japanese-style@jssst.or.jp 宛に電子メールを送ると、必要なスタイルファイルや書き方のサンプルが送り返されてくる。

5.2 OHP やポスター

OHP やスライド向きの文字の大きさやレイアウト等の専用のスタイルファイルとして、

A Short-Cut to Your Slides:

eslides や lslide がある。例えば、eslides(.sty) をドキュメントスタイルオプション指定し、slide 命令を使うと、右のような感じで、さらに実際には、字は大きく、上下に枠線が引かれ、外にタイトルやロゴが書ける。

- take your finished article file
- add the style option eslides
- initialize \magnification and \conference
- put some \begin{slide}s and \end{slide}s around the parts you want to present
- comment out the rest
- run L^AT_EX

会場の張り紙やポスター等のように、普通の用紙を張り合わせて大きなものを作る場合、poster(.sty) をドキュメントスタイルオプション指定して、Poster 命令を使うと、指定サ

^{†26} 世界中の研究者が T_EX を使っているので、この手の必要を満たすマクロ定義 (スタイルファイル) や外部ツール (コマンド) は、ほとんど誰かが既に作っているわけである。

^{†27} /usr/local/tex/lib/macros, /usr/local/share/tex/macros.new

イズに拡大された文書の各部分が、A4の紙に分割されてレイアウトされる。それらを出し、のりで張り合わせればよい。

また、OHP やポスターを書く場合等には、効果を狙った様々な囲み枠付けが欲しくなる。上の OHP の例では、`fancybox(.sty)` をドキュメントスタイルオプションで指定して、影付きの枠 `\shadowbox{...}` を使ったが、他に、`\ovalbox{...}` や `\doublebox{...}` 等が使える。

5.3 グラフ的構造

論理的 (規則的) なグラフ構造は、作画ツールではなく、もっと形式的な言語レベルで書きたい。例えば、ツリー構造を書くには、`egtrees(.sty)` がある。図 5。

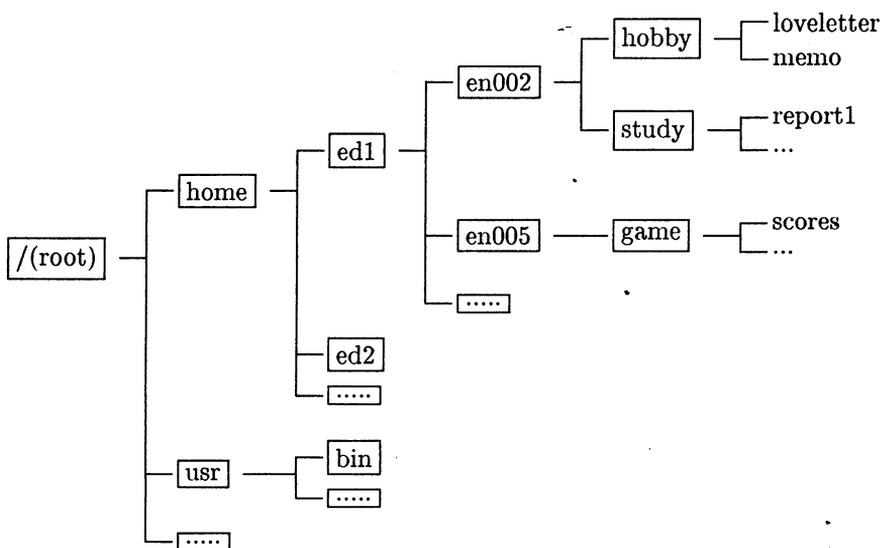


図 5 `egtrees.sty` による木構造グラフの例

フローチャートを書くには、 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ のスタイルファイル (`Flow.sty` 等) で直接書く方法もあるが、一定の書式でフローチャートの内容を記述し、それをあるコマンドで $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ の `picture` 環境での命令列に変換する方法があり、後者の方が高機能である。センターでは `flow2latex` というコマンド名で使える^{†28}。

このコマンドを使って生成した $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}(\text{picture})$ 部分ファイル (仮に名前を `example.pic` とする) を本文に取り込むには、

```

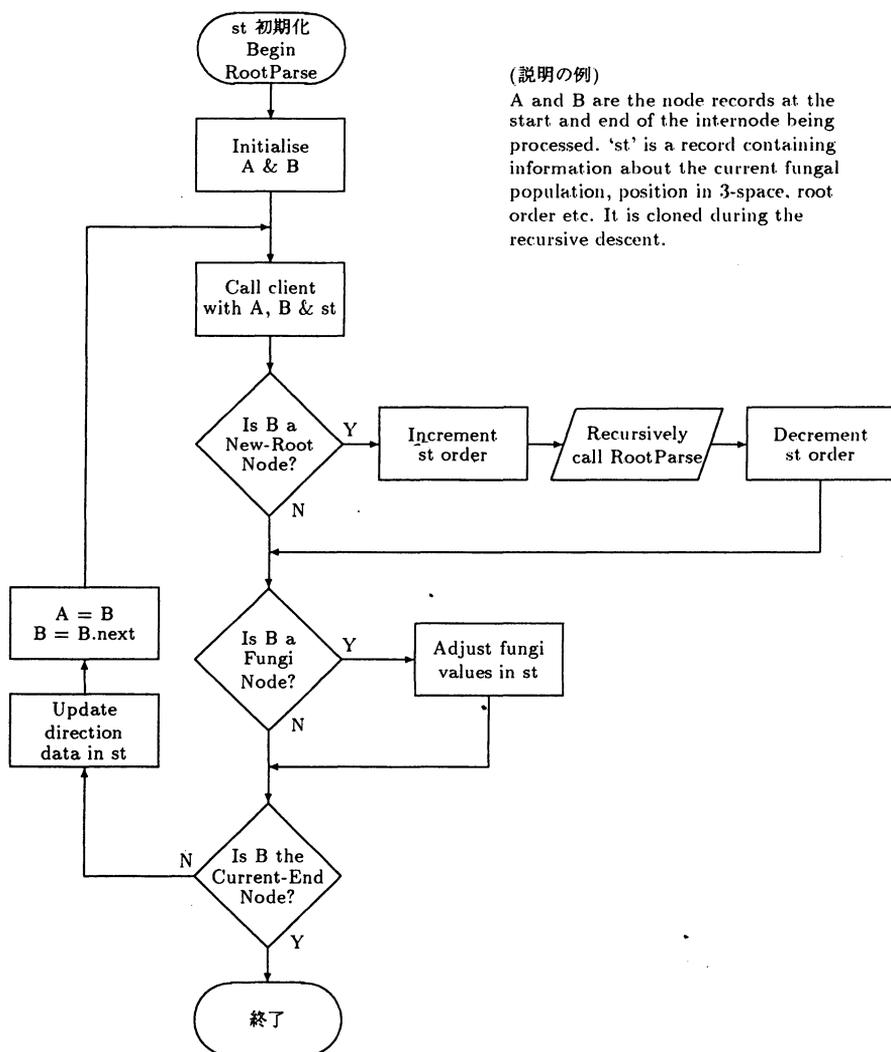
\setlength{\unitlength}{1.8em}
\input{example.pic}

```

のように、`unitlength` を設定し、`input` 命令で取り込めばよい^{†29}。図 6。

^{†28} オリジナル名は `flow`。ftp:pub/prg/Tex/misc/flow に使い方とサンプルがある。

^{†29} さらに、ドキュメントスタイルオプションに、`epic`, `eepic` を (この順で) 指定した方がきれいに表示されるようである。



(説明の例)

A and B are the node records at the start and end of the internode being processed. 'st' is a record containing information about the current fungal population, position in 3-space, root order etc. It is cloned during the recursive descent.

図 6 flow2latex によるフローチャート

5.4 プログラミング言語等のソースファイル

文書中にプログラミング言語等のソースファイルを引用したい場合、言語ソースプログラムは \LaTeX に (単なるドキュメントとしての観点から) 勝手にレイアウトされたら困るので、単純には書いたままの位置に表示される verbatim 環境 (verb 命令) を使えばよい。しかし、便利なマクロやコマンドもある。

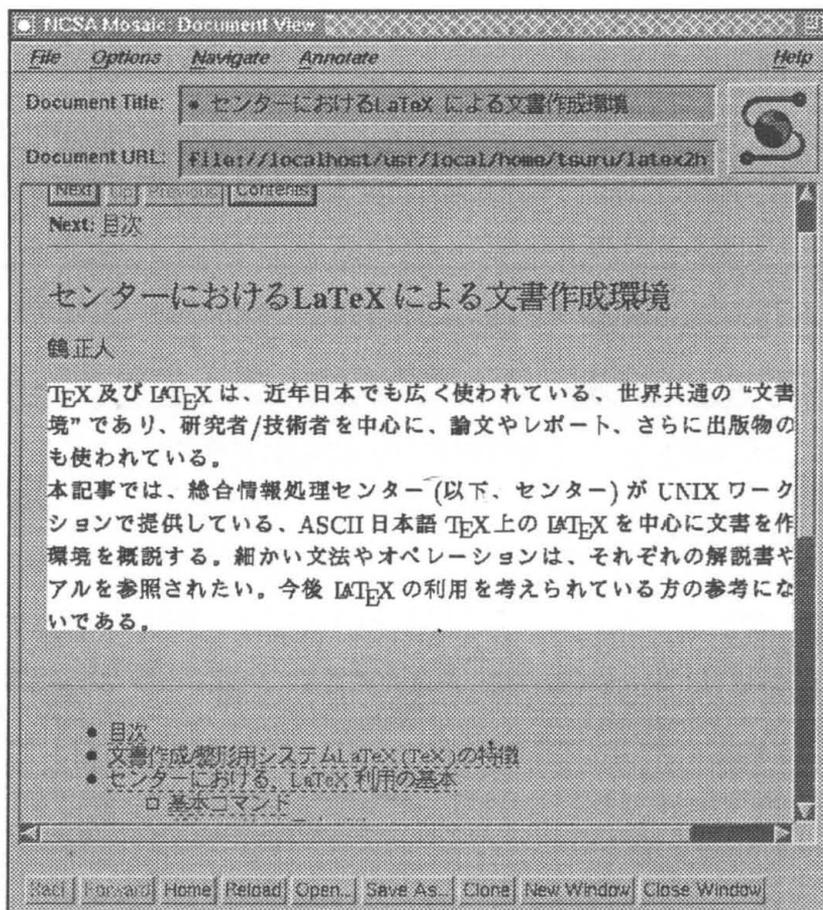


図 7 jlatex2html による変換

- `verbatimfiles(.sty)` を使うと、LaTeX 文書内には、取り込むべきソースプログラムのファイル名だけ書いておくと、`latex` コマンド実行時に (`verbatim` 環境で) 取り込んでくれる。その時左端に行番号を付けることもできる。ソースプログラムはしばしば変更されるので、これを使うと保守が楽になると思われる。
- `c++2latex` というコマンドは、ANSI-C または C++ のソースプログラムを与えると、きれいにインデントされ、C(C++) のキーワードが強調フォントで表示されるような、LaTeX の部分ファイルを生成する。この生成されたファイルを、本文の文書ファイルから `input` 命令/`include` 命令で取り込めばよい。
見栄えがするので、C 言語教育の教材テキストを作成する場合等には好んで使われるかも知れない。

さらに最近、LaTeX 文書の中で、ソースプログラムを表示する (取り込む) のではなく、逆に、ソースプログラムファイルの中 (コメント欄) に、LaTeX の文法 (機能) を使ったいろいろ複雑なコメントを記述できるようにする `src2tex`, `src2latex` というコマンドも公開され、センターでも利用できる。単一のソースファイルで説明用ドキュメントもソースプ

ログラムも兼用することで保守性を高めることができる。

5.5 texinfo 形式のマニュアル

GNU project 等の開発するフリーソフトウェアを中心に、ソフトウェアのオンラインマニュアルが、従来の man(roff に基づく) 形式ではなく、texinfo と呼ばれる形式で作成されるようになった(内部的に T_EX が利用されている)。通常、xxxx.texi とか xxxx.texinfo 等の名前のファイルであり、それを直接利用するのではなく、

- makeinfo コマンドを使って、emacs 系エディタ上でのオンラインマニュアルシステムである info 形式 (xxxx.info, xxxx.info-1, ...) に変換できるので、それを、emacs 上で引く。

```
% makeinfo xxxx.texi ▽
```

- texi2dvi コマンドを使って、DVI 形式 (xxxx.dvi) に変換できるので、それを、(T_EX と同様に) 画面で見たり、プリンタに印刷したりする。

```
% texi2dvi xxxx.texi ▽
```

- 形式的には、man と違って、階層構造を持てるのが特長である。

5.6 HTML 形式への変換

近年の WWW(World Wide Web) サーバの大流行で、L^AT_EX 文書を HTML(HyperText Markup Language) 形式へ変換するシステムも開発され^{†30}、センターでも利用できる。L^AT_EX 文書ファイルを yyyy.tex として、

```
% jlatex2html yyyy.tex ▽
```

とすると、yyyy というディレクトリが自動的に切られ、その中に yyyy.html をトップとする一連の html ファイルやインライン画像ファイルが生成される。例えば、本記事を HTML 形式にして、Mosaic で見た様子が、図 7 である。

目次、章節の階層構造や脚注等は、リンクとして張られ、マウスのクリックで jump する。また、数式、箇条書や表等は、イメージとして埋め込まれる。

参考文献

- [1] L. Lamport, (大野俊治他訳). 文書処理システム L^AT_EX. アスキー, 1990.
- [2] 岩熊哲夫, 古川徹生. 連載 スタイルファイル活用法. *UNIX MAGAZINE*, 1994-5.
- [3] 野寺隆志. 楽々L^AT_EX. 共立出版, 1990.
- [4] 磯崎秀樹. L^AT_EX 自由自在. サイエンス社, 1992.
- [5] 鈴木裕信. ひろのぶの L^AT_EX 入門. *a copy-free document on INTERNET*, 1990.
- [6] 花田英輔. L^AT_EX 入門の入門. 総合情報処理センターレポート, 1993.
- [7] 中村眞. イメージと画像 (1)-(4). *UNIX MAGAZINE*, Vol. 9, No. 1-4, 1994.

^{†30}将来的には、T_EX と HTML 等が融合されるかもしれないが。