

情報処理 2 第 6 回

TEX (1)

かつらだ まさし
桂田 祐史

2007 年 5 月 22 日

この授業用の WWW ページは <http://www.math.meiji.ac.jp/~mk/syori2-2007/>
いよいよ TEX の解説をはじめます。卒業研究レポートを TEX を使って書きませんか？

1 連絡事項

- 0508 号室のパソコンの Windows XP 環境に、(仮称) 十進 BASIC が正式にインストールされました。プログラムをダブルクリックすれば十進 BASIC が起動しますし、オンライン・ヘルプも利用できます。
- 0508 号室のパソコンの Windows XP 環境に、TEX 関係のソフトウェア一式がインストールされました。これからの授業で使い方について説明します (来年度からはすべての教室のパソコンで利用できるようになると思われます)。

システム管理課の方達に、各種ソフトウェアをインストールしていただきました。学期中のため相当の手間がかかったはずですが、どうもありがとうございます。

2 TEX とは？

(駆け足で説明する。)

TEX は組版ソフトである

TEX は著名なコンピューター科学者であるドナルド・クヌース (Donald Knuth¹, ウィキペディアにも載っています) の開発した文書整形システム (組版²システム) です。TEX は日本では「てっく」または「てふ」と呼ばれています³。

¹<http://www-cs-faculty.stanford.edu/~knuth/>

²組版とは、文字や図版などの要素を配置して、紙面を構成することで、もともとは活版印刷において、活字を組み上げることから来ている。

³「てっくす」とは読みません。ちなみに TEX の解説書に “Joy of TEX” という本があって、それは英語圏の国では有名な本のパロディだったそうです。昔、テレビで深夜映画を見ていたら、元ネタの本が出て来て、思わず見入ってしまいました。

当初、数式を含む英文を清書することを目的に、従来の組版技術の歴史を入念に調べた上で、それをコンピューター上で実現することを目標に開発されたそうです。

(ワープロとの違いを説明すべきだけど...)

TeX はフリーソフトである

Knuth 自身は TeX に関する情報を完全に公開して (書籍になっています)、ソフトウェアを無償で利用することができます。また、多くのボランティアの活動により、TeX を補助、発展させるためのソフトウェア、データもほとんどは無償で利用可能です。例えば、TeX 本体や周辺ソフトウェアの C 言語への変換、表示用ドライバー、印刷用ドライバー、PDF への変換ソフトウェア、日本語対応、日本語フォント (やそれを利用する仕組み)、Windows 環境への移植、インストーラーなど。これら成果物はすべてインターネットから無償で入手できます。

TeX は数学の世界では標準である

数学者村では標準の文書作成ソフトウェアです。理工系の多くの分野で利用されていますが、それだけでなく文系の研究者が利用した例もあります (発音記号や、ややマイナーな言語などを扱う場合)。

TeX で高品位の文書が作成できる

組版技術をしっかり研究した上で作られたものであるため、高品質な仕上がりが得られます。異なる環境下での再現性も抜群です。英語圏ではもちろん、日本でも理工系の多くの書籍 (中学高校の教科書や問題集なども含む) で採用されています。

TeX で作った文書は PDF にして配布が楽々

それ自身は文書の配布フォーマットとして適当ではありませんが (表示、印刷に専用のソフトウェアが必要なため)、簡単に PDF (portable document format) に変換できるので、配布に際して問題はありません。

この授業では、TeX の一種である LaTeX (正確にはその日本語対応版 pLaTeX) を使ってもらうことにします。

3 Windows 環境での利用法 (1)

今回は、“コマンドプロンプト”の中でキーボードからコマンドを打ち込むという、最も基本的な (原始的な?) 使い方を説明します。

(TeX は複数のプログラムを利用して目的を達成するソフトウェアです。完全なブラックボックス)

クスにしてしまうと「わけが分からなく」なるので、個々のプログラムが目に見える「コマンドによる操作」に慣れてもらいます。後でより便利な方法を紹介するつもりですが、基本的な方法は、他の環境 (Linux, Mac, ...) に行ったときも応用が効きますし、トラブルが発生したときに自分で解決する力をつけられる可能性も高くなります。)

3.1 コマンド・プロンプトの準備

1. まずコマンドプロンプトのショートカットを用意しましょう。情報処理 2 用の作業フォルダ (マイドキュメントの中に “syori2” のような名前で作成しました) を開いてから、[スタート] [すべてのプログラム (P)] を開き⁴、[コマンドプロンプト] をマウスの右ボタンを押しながら作業フォルダまでドラッグしてコピーします⁵。
2. 試しにショートカットのアイコンをダブルクリックして起動すると、Z:¥> のようなコマンドプロンプトが現われるはずです。試しに

(以下の “syori2” は自分で選んだ名前に置き換えて下さい。なお、コマンドプロンプトで日本語を入力するには、Alt-漢字 とします。)

```
Z:\>dir
```

(色々表示される)

```
Z:\>cd .windows2000
```

(マイドキュメントに移動する)

```
Z:\.windows2000>dir
```

(マイドキュメント内のファイルの名前が表示されるはず)

```
Z:\.windows2000>cd syori2
```

(マイドキュメント内のフォルダ syori2 に移動する)

```
Z:\.windows2000\syori2>dir
```

(syori2 内のファイルの名前が表示されるはず)

```
Z:\.windows2000\syori2>type first.BAS
```

(BASIC プログラム first.BAS があったとして、それを表示してみる。)

3. 今後の TeX に関する作業は、マイドキュメント内の作業フォルダで行うのがよいので、コマンドプロンプトを実行した直後に Z:¥.windows2000¥syori2 が作業フォルダになるように設定します。具体的には、[コマンドプロンプト] のアイコンを右クリックして現われるメニューの [プロパティ] を選んで、「作業フォルダ」を “%HOME%\HOMEPATH%” から “%HOME%\HOMEPATH%\¥.windows2000¥syori2” に変更します。もちろん “syori2” の部分は自分で選んだ名前で置き換えてください。OK としてから、アイコンをダブルクリックして最初から

⁴[すべてのプログラム (P)] に [コマンドプロンプト] があるのは、情報処理教室のパソコンの特殊事情です。普通は [すべてのプログラム (P)] [アクセサリ] [コマンドプロンプト] と辿ります。

⁵[コマンドプロンプト] を右クリックして、[送る] で [マイドキュメント] を選択すると、[マイドキュメント] にショートカットが作れるので、それを移動しても良いでしょう。

```
Z:\.windows2000\syori2>
```

となったら成功です。

3.2 秀丸で .tex ファイルを作成する

1. 秀丸エディタを起動して、次のような内容を入力してください。

```
tamago.tex  
\documentclass[12pt,leqno]{jarticle}  
\begin{document}  
\end{document}
```

これを `tamago.tex` という名前で保存しましょう。そのためには、[ファイル] [名前をつけて保存 (A)] で、

- 「保存する場所 (I)」をマイドキュメント内の作業フォルダ (`syori2`) にして、
- 「ファイル名 (N)」を “`tamago.tex`” にして、
- 「ファイルの種類 (T)」を「すべてのファイル (*.*)」に

します。

注意「ファイルの種類 (T)」をデフォルトの「テキストファイル (*.txt)」のままに保存すると、環境によっては、“`tamago.tex.txt`” というファイルが出来てしまいます。これは使えません。

2. 一度秀丸エディタを終了して、作業フォルダ (`syori2`) 内の `tamago.tex` のアイコンをダブルクリックして起動してみましょう。少し書き足してみます。

```
first.tex  
\documentclass[12pt,leqno]{jarticle}  
\begin{document}  
こんにちは。  
  
\[  
  \int_{-\infty}^{\infty} e^{-x^2}\;dx = \sqrt{\pi}.  
\]  
  
\end{document}
```

これを `first.tex` という名前をつけて保存しましょう。[ファイル] [名前をつけて保存 (A)] とすると、「ファイルの種類 (T)」は「その他 (*.TEX)」となっているはずですが、「ファイル名 (N)」のところだけ `first.tex` として OK です。

3. 今後は、手順 2 の部分、つまり `tamago.tex` を開いてから、「名前をつけて保存 (A)」で名前を決める、というやり方で作業するのが私のお勧めです。

3.3 .tex ファイルから .dvi ファイルを作る

1. 作業フォルダ内のコマンドプロンプトのアイコンをダブルクリックして、コマンドプロンプトを開きます。
2. キーボードから `platex first.tex` と入力し、最後に `Enter` を打ちます (最後の `.tex` は省略できるかも知れません)。

```
platex first.tex Enter
```

うまく行けば次のように次のコマンド待ちの状態になるはずです。

```
Z:\.windows2000\syori2>platex first.tex  
  (色々表示が出る)  
Z:\.windows2000\syori2>
```

`first.dvi` というファイルが出来ているはずです。

3. `first.tex` の内容に打ち間違いがあった場合、エラーになり、`platex` コマンドはどうすればよいか、`?` というプロンプトを出して尋ねてきます。それに対しては、`x` `Enter` (即終了) あるいは `q` `Enter` (バッチモードでとにかく最後まで処理) として、終了させてください (それぞれ `eXit`, `Quit` ということでしょう)。エラーメッセージを頼りに `first.tex` を修正してから (上書き保存 `C-s` を忘れずに)、2 に戻ります (`platex first.tex` とする)。

3.4 .dvi ファイルを表示する/印刷する

1. コマンドプロンプトから

```
Z:\.windows2000\syori2>dviout first.dvi
```

と打ちます (最後の `.dvi` は省略できるかも知れません)。新しいウィンドウが現われ、`.dvi` ファイルが表示されるはずです。

2. 作業中は `dviout` を終了する必要はあまりないでしょう。`first.tex` を書き換えて、`platex` で処理し直しせば、`dviout` は新しい内容を表示してくれます。
3. 印刷したい場合は `dviout` のメニューから行えばよいでしょう。

3.5 .dvi ファイルから .pdf ファイルを作る

課題のレポートとして `.dvi` ファイルを送ってもらうので十分なこともあるのですが、`.dvi` ファイルには図のデータを含めることが出来ないことと、`.dvi` ファイルが読めない人も多い (`dviout` はどこにでもインストールされているものではない) ので、人に渡すときは、PDF (portable document format) に変換したものを渡すのがお勧めです。

それには `dvipdfm` あるいは `dvipdfmx` というコマンドを用います。

```
Z:\.windows2000\syori2>dvipdfm first.dvi
```

または

```
Z:\.windows2000\syori2>dvipdfmx first.dvi
```

で `first.dvi` から `first.pdf` が作成されます。

出来上がった `first.pdf` は Adobe Reader 等で表示・印刷が可能です。

なお、Adobe Reader で読んでいる間は、`first.pdf` を変更することができません。`dvipdfm` (`dvipdfmx`) を実行する時は、一旦 Adobe Reader を終了して下さい。

4 課題 5A

自分が作った `first.tex` と `first.pdf` を送って下さい。Subject は “情報処理 2 課題 5A レポート” です。

A 参考: Linux 環境での L^AT_EX の使い方

T_EX は UNIX の世界での利用実績が長いので、UNIX 環境では使いやすくなっている、と言えるでしょう。情報処理教室の Linux 環境ではどうすれば使えるか、簡単に説明します。

まず emacs のようなエディターでファイル名末尾が “.tex” であるファイルを作ります。例えば “first.tex” であれば、

```
icr3-1008% emacs first.tex &
```

とようにすれば OK です。

YaTeX (野鳥) のような便利なソフトもありますが、それはまたの機会に紹介するとして、ここでは基本的な T_EX 関係のコマンドの使い方を紹介します。

.tex ファイルを .dvi に変換するために platex コマンドを使うのは、Windows の場合と同じです⁶。

.tex を .dvi に変換 (「コンパイル」)

```
icr3-1008% platex first.tex
```

“first.tex” に (L^AT_EX の文法上の) 誤りがなければ、“first.dvi” という名前のファイルができます。

画面に (組版結果を) 表示する (「プレビューする」という) には `pxdvi` コマンドを用います⁷。

⁶現在、日本語が使える L^AT_EX としては、pL^AT_EX が標準的で、コマンドの名前は普通 platex である。

⁷UNIX 環境ではプレビューアとして `xdvi` が定番で、`pxdvi` はその派生物らしい。なお、Windows 環境では、`dviout` というソフトが定番である。

.dvi ファイルをプレビュー (画面表示)

```
icr3-1008% pxdvi first.dvi &
```

印刷するには、情報処理教室の場合、例えば次のようにします⁸。

.dvi ファイルを印刷

```
icr3-1008% pdvips -f first.dvi | lp -d プリンター名
```

(両面印刷機能を持ったプリンターでは、途中で適当なフィルターを挟むと両面で印刷されるようになります。例えば `double`⁹ を用いて、`pdvips -f first.dvi | double | lpr` で両面印刷できます。)

dvi ファイルを PDF に変換する方法については後日説明します (`dvipdfm` がありますが、そのままでは日本語うまく行かないみたいです)。

B L^AT_EX の書き方 (1)

数式の書き方は置いておくとして、上の見本で使ったような基本的事項を説明しておく。

L^AT_EX 文書で最低限必要なのは次の内容である。

```
\documentclass[12pt]{jarticle}% スタイルの指定
\begin{document}
\end{document}
```

レポートでは、タイトル、著者名、日付が必須なので、次のようなものが必要と思って良い。

```
\documentclass[12pt]{jarticle}% スタイルの指定
\begin{document}
% この行は注釈。次の4行でタイトル、著者名、日付を表示する
\title{レポート課題 X}
\author{2年16組99番 桂田 祐史}
\date{2007年4月26日}
\maketitle
\end{document}
```

特殊文字以外は `\begin{document}` と `\end{document}` の間に書けば表示される。

⁸dvi ファイルを PostScript に変換するためのソフトウェアとしては `dvips` が定番で、`pdvips` はその派生物であるらしい。

⁹<http://www.math.meiji.ac.jp/~mk/syori2-2007/double>

```

\documentclass[12pt]{jarticle}% スタイルの指定
\begin{document}
% この行は注釈。次の4行でタイトル、著者名、日付を表示する
\title{レポート課題 X}
\author{2年16組99番 桂田 祐史}
\date{2007年4月26日}
\maketitle

ここにフツの文字で書いたものは出力される。
\end{document}

```

実際には色々な記号が $\text{T}_\text{E}\text{X}$ の命令と解釈される特殊な文字となっている。プログラムなどを表示するには、`verbatim` 環境で利用するのが簡単である。

```

\documentclass[12pt]{jarticle}% スタイルの指定
\begin{document}
% この行は注釈。次の4行でタイトル、著者名、日付を表示する
\title{レポート課題 X}
\author{2年16組99番 桂田 祐史}
\date{2006年4月26日}
\maketitle

ここにフツの文字で書いたものは出力される。

% verbatim 環境の中は特殊文字であってもそのまま出力される。
\begin{verbatim}
#include <stdio.h>
int main()
{
    printf("Hello\n");
    return 0;
}
\end{verbatim}
\end{document}

```

参考文献

- [1] 奥村晴彦, $\text{L}^{\text{A}}\text{T}_\text{E}\text{X} 2_\epsilon$ 美文書作成入門 改訂第4版, 技術評論社 (2007).