

LaTeXによる文書作成(その1)

プログラミング演習 1#11

平成 26 年 6 月 27 日

1 設定変更のお願い

これから 4 回にわたる LaTeX の演習では, CentOS にデフォルトで収録されていない新しいバージョンの LaTeX 関連ソフトウェアである TeX Live 2013 を用います. 最初の方の回で設定した設定ファイルのままだと, その辺がどうもうまく行かないようなので, 以下のように変更して下さい.

1.1 シェルの設定変更

TeX Live 2013 のコマンドは, 演習室の環境では `/usr/local/texlive/2013/bin/x86_64-linux/` 以下にインストールされておりますので, コマンドパスに加えておいて下さい. `bash` をお使いであれば, `.bashrc` の一番最後にでも

```
export PATH=/usr/local/texlive/2013/bin/x86_64-linux/:$PATH
```

の一行を, `tcsh` をお使いであれば, `.cshrc` の一番最後にでも

```
setenv PATH "/usr/local/texlive/2013/bin/x86_64-linux/:$PATH"
```

の一行を加えておくとよいです.

1.2 Emacs の設定変更

この演習では, LaTeX ソースを編集するのに(基本的には)Emacs を使用せず, Kile という専用エディタを使用します. しかし, Emacs でちょっとした変更ができたほうがよい場合もあります.

Emacs で LaTeX のソースを編集する上で, 設定する必要があることは特にありません. しかし, 日本語の混じった LaTeX ソースファイルを演習室環境でコンパイルする場合, UTF-8 でエンコードされている必要があります. そこで, `.emacs` ファイルに以下の記述をしておくと, `.tex` ファイルを開くたびに UTF-8 に自動的に変換してくれるようになります.

```
(add-hook 'latex-mode-hook
  (lambda ()
    (set-buffer-file-coding-system 'utf-8)
  ))
```

実は去年までの LaTeX 環境だと文字コードとして EUC-JP を使う必要があったので, 最初の方の授業で, `.tex` ファイルを開くたびに EUC-JP に自動変換する `.emacs` ファイルを皆さんにコピーさせていただきました. すみません. `.emacs` の

```
(add-hook 'latex-mode-hook
  (lambda ()
    (set-buffer-file-coding-system 'euc-jp-unix)
  ))
```

の箇所の `'euc-jp-unix` を `'utf-8` に変更すればよいです.

2 L^AT_EX による文書作成手順

L^AT_EX(ラテフ, ラテック) は、文書整形システム T_EX(テフ, テック)¹を使い易く拡張したものです。L^AT_EX は、いろいろなスタイル指定するだけで、文章の内容を変更せずにそのスタイルに沿ったレイアウトを出力します。従って、L^AT_EX による文書作成は、通常のワープロと違い、ディスプレイに表示された文字配置がそのまま印刷結果になるわけではありません。準備したソースファイルを L^AT_EX で処理することで文章中に埋め込んだ各種コマンドを反映した dvi ファイルが作成されます。一部の印刷機ではこの dvi ファイルをそのまま受け付けることができます。画面上で整形された文書を表示させるには、演習室の環境では pxdvi コマンドを使用します。また、Postscript プリンタで印刷するには dvips コマンドを使って dvi 形式のファイルを Postscript 形式のファイルに変換します。

L^AT_EX による文書作成の具体的な手順は下記のようになります。

1. L^AT_EX のソースファイルを作成する。

まず、emacs などのエディタを用いて文章を入力し、文章中にレイアウトに関するコマンドを記述します。L^AT_EX 用に作ったファイルは、report.tex のように、ファイルの拡張子を tex にします。このファイルを L^AT_EX のソースファイルと呼びます。

2. platex コマンドでソースファイル进行处理 (コンパイル) する。

L^AT_EX のソースファイルをコンパイルするには、

```
% platex report.tex
```

のように、platex コマンドの引数に、L^AT_EX のソースファイルを指定します。

3. コンパイルの結果、dvi ファイルが得られる。

エラーがなければ dvi ファイルが生成されます。L^AT_EX のソースファイルが report.tex であれば、report.dvi というファイルが作られます。エラーがあった場合は、そのメッセージをよく確認してソースファイルを正しく修正します。

4. dvi ファイルを画面で確認する。

意図した通りに文書が作成できているかを、pxdvi コマンドによりディスプレイ上で確認できます。

```
% pxdvi report.dvi
```

のように、evince コマンドの引数に、dvi ファイルを指定します。

5. プリンタで印刷するために Postscript 形式のファイル (PS ファイル) に dvi ファイルを変換する。

dvi ファイルを PS ファイルに変換するには、dvips コマンドを用います。

```
% dvips report.dvi
```

とやれば、report.ps という PS ファイルを作ることができます。

なお、PDF 形式のファイルに変換したい場合は dvipdfmx コマンドを用います。

¹T_EX を開発した Donald Knuth によると、「語源が *τεχνη*(テフニー) なのでテフと嫁」という事です。でも古典ギリシャ語読みだとテクネーなのでテクと読んでもいいような気もする。

```
% dvipdfmx report.dvi
```

本日のやってみよう#1

私のホームページの例の場所にある `fun.tex` を保存して、 \LaTeX でコンパイルして、`pxdvi` で確認してみよう。さらに、`lv` コマンドで元のファイルの中身を確認してみよう。

ちなみに、「プログラミング演習1」で今まで配ってきた資料は、すべて \LaTeX を使って作成してます。参考までに、本資料の \LaTeX ソース (の少し古いバージョン) を私のホームページの例の場所² に置いてありますので、保存したあと \LaTeX コンパイルしてみてください。

3 \LaTeX エディタ Kile の準備

\LaTeX は一度使い慣れてしまうと非常に便利なのですが、使いこなせるようになるまでにいろいろなコマンドを覚える必要があり、初学者にとっては敷居が高いようです。特にこの科目では、数回の授業で習得していただく必要があることから、少しでも皆様の負担を軽減するために Kile という \LaTeX 専用エディタを用いることにします。

Kile は、Gnome メニュー/K メニューの「オフィス」の下の「Kile」を選択するか、端末から `kile` と打ち込むことで起動します。そのままの設定でも \LaTeX 文書の編集には使えるのですが、コンパイル等を行う上で、日本語に対応していない外部コマンドを呼び出す設定になっています³。演習室の日本語環境に合わせて次の様に設定変更しておくといいでしょう。

1. Kile のメニューの「設定 (S)」をクリックし、「Kile を設定 (C)...」を選択。
2. 設定ダイアログ左のツリーの「ツール」の下の「ビルド」を選択。
3. 「ツールを選択 (T)」中の“DVItOPDF”を選択し、「コマンド (M)」欄の“dvipdfm”を“dvipdfmx”に変更。さらに「設定を選択 (C)」で“Landscape”や“Black and White”を選んで同様に“dvipdfmx”に変更。
4. 「ツールを選択 (T)」中の“LaTeX”を選択し、「コマンド (M)」欄の“latex”を“platex”に変更。
5. 「ツールを選択 (T)」中の“TeX”を選択し、「コマンド (M)」欄の“tex”を“ptex”に変更。
6. 「ツールを選択 (T)」中の“ViewDVI”を選択し、「コマンド (M)」欄の“kdvi”を“pxdvi”に変更。

それではさっそく文書を作成してみましょう。Kile のメニューの「ウィザード (W)」をクリックして「クイックスタート」を選択すると「クイックスタート」ダイアログボックスが起動します。ここでいろいろなオプションを選択できますが、あまりにたくさんのオプションがあるためパニックってそのまま OK をクリックしてしまうかもしれません。それで結構です。

試しに `\usepackage` と書いてある行を全部削除して、`\begin{document}` と `\end{document}` の間に適当な日本語の文章を入れて、`Alt+1`⁴ を押してみましょう。自動的にコンパイルされて `pxdvi` が立ち上がるはずですよ。

²<http://nautilus.cs.miyazaki-u.ac.jp/~skata/Penshu1/2014/>

³デフォルトでは、例えばコンパイルを行うのに、日本語対応の `platex` ではなく未対応の `latex` コマンドを用いる設定になっています。

⁴`Alt` キーを押しながら `1` キーを押す

4 L^AT_EX ソースファイルの作成

L^AT_EX のソースファイルの大まかな骨組みは下記の様になります。

<code>\documentclass[a4j,12pt]{jarticle}</code>	文章のスタイルを指定 (必須) 2 段組みにする場合は角括弧内に <code>twocolumn</code> を入れる
<code>\title{プログラミング演習 1 レポート}</code>	文章の表題を記述
<code>\author{宮崎 太郎}</code>	文章の著者を記述
<code>\date{平成 26 年 6 月 27 日}</code>	日付を記述 {} を省略すると今日の日付になる
<code>\begin{document}</code>	文章の始まりを指定 (必須)
<code>\maketitle</code>	タイトルの表示を指示 (準必須)
ここから本文です。レポートを書きます。	
<code>\section{はじめに}</code>	第 1 節の表題
今回提出するレポートは.....	
<code>\subsection{目的}</code>	第 1.1 節の表題
書く目的は下記のキーワードです。	
<code>\begin{itemize}</code>	箇条書き環境の指定
<code>\item IT 革命</code>	箇条書きの項目指定
<code>\item 情報化社会</code>	
<code>\item インターネット</code>	
<code>\end{itemize}</code>	箇条書き環境の終了指定
<code>\section{おわりに}</code>	第 2 節の表題
今回のレポートで分かったことは.....	
<code>\end{document}</code>	文章の終わりを指定 (必須)

最初の `\documentclass` は作成する文書の基本構造を指定します。続く `{}` 内に文書の基本構成を、`[]` 内にオプションを指定します。この例では `jarticle` 形式⁵を文書の基本構成とし、オプションで A4 版を使用すること (`a4j`)、文書中の文字の基本サイズを 12 ポイント (12pt) と指定します。

次に必要なのは `\begin{document}` とファイルの末尾にある `\end{document}` になります。この 2 つの行の間に文書の本文を、様々なコマンドを混じえて記述していきます。

⁵英語の文書を作成する時は `article` を用います。なお、Kile のデフォルトは `jarticle` ではなく `article` です。

4.1 文書構造の指定

文書やレポートには、読み手が理解しやすいよう章立てをする場合が多くなります。L^AT_EX には章立てをするコマンドが用意され、大変使い易くなっています。以下に章立てを作成するのに用いる主なコマンドを示します。これらは、Kile では左上ツールバーの `section` と書いてあるドロップダウンをクリックすることで選択可能です。

```
\part          \chapter      \section
\subsection   \subsubsection \paragraph
```

最初の2つ (`part` と `chapter`) は、`jbook` スタイルと `jreport` スタイル⁶でのみ使用できます。これらのコマンドを使用すると自動的に章番号が付され、コマンドが現れるたびに自動的に番号は増やされていきます。また、始めの方にあるコマンドほど上位階層であることを表し、下部階層を表すコマンドが現れた場合、上位階層の番号、下部階層の番号というふうに番号が付けられます。

L^AT_EX では、ソースファイル中の改行は空白扱いされます。文書中の空行は一般に段落が変わることを意味しますので、段落分けを行いたい場合には空行を一行以上ソースファイル中に入れることで実際の文書中で改行が行われ、新しい段落の先頭の文字は一文字分下げが自動で行われます。

4.2 環境の使用

L^AT_EX では、様々なレイアウトを実現するためには「環境」を使用します。環境を記述するための一般的な書式は下記の通りです。

```
\begin{環境名}
  この中の文章は、指定された環境に沿ったレイアウトになります。
\end{環境名}
```

指定できる環境の代表的なものは表 1 にまとめてあります。これらはすべて、Kile では Latex メニューから選択可能であり、一部の環境にはボタンも用意されています。

表 1: L^AT_EX でしばしば使用される環境

環境名	機能
<code>verbatim</code>	文章をソースファイルそのままのレイアウトで表示
<code>quote</code>	文章の左右の余白を狭めて表示
<code>itemize</code>	箇条書き (バレット「•」) で表示
<code>enumerate</code>	箇条書き (番号) で表示
<code>tabular</code>	表を作成
<code>center</code>	文章をセンタリング

⁶英文の場合は `book` と `report` .

itemize 環境と enumerate 環境は、\item で箇条書きの中身を記述します。tabular 環境については、<http://tdh8025/Lecture/H25-2013/PE1/> の中のファイルを参考にして下さい (例えば、この資料 (lec10-latex-1.tex) には表が複数載ってます)。

文字の大きさや字体を指定することもできます。表 2 に、代表的なコマンドを挙げておきます。これらの指定は環境名と異なり、一度指定すると、後の文章がすべて影響を受けます。この場合は、必要な範囲を { } で囲みます。例を挙げると、{\large 大きく書きたい文章} のように指定します。

Kile では、文字の大きさは「normalsize」と書いてあるドロップダウンで、字体はその横の B とか i とか書いてあるボタンで指定できます。

表 2: 文字の大きさと字体

コマンド	表示
\tiny	とても小さい
\small	少し小さい
\normalsize	普通の大きさ (何も指定しない)
\large	少し大きい
\Large	大きい
\huge	とても大きい
\bf	Bold (太字)
\rm	Roman (何も指定しない)
\it	<i>Italic</i>
\tt	Typewriter

文章中に書く数式は、\$と\$で式を囲みます。数式中で使用できる主なコマンドを表 3 に示します。これらは Kile では、ツールバーのボタンか、そこになれば左の「記号」タブを使って選んで入力することができます。ただ、 \LaTeX の利点の一つは「マウスに手を伸ばさなくても数式がサクサク書ける」ということなので、慣れてきたらなるべくコマンドを覚えて打ち込むようにしましょう。

表 3: 数式中で使用できる主なコマンド

機能	コマンド	表示
右肩添字	$\$x^{2e}\$$	x^{2e}
右下添字	$\$x_{1}\$$	x_1
分数	$\$\frac{1}{2}\$$	$\frac{1}{2}$
積分	$\$\int_0^{\infty}\$$	\int_0^{∞}
和	$\$\sum_{k=1}^N\$$	$\sum_{k=1}^N$

これらを組み合わせると、 $a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3}}}$ や、 $\sum_{k=1}^n k = \frac{n(n+1)}{2}$ などが表示できます。文章中の数式が小さくなって見づらい時には、数式を、文章とは独立した別の行で表示することもできます。その際には、\$の代わりに、\[と\] で区切ります。先の例だと、以下のようになります。

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3}}}$$

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

ちなみに、上記を表示するためのコマンドは、それぞれ下記です。

```
\[ a_{0}+\frac{1}{a_{1}+\frac{1}{a_{2}+\frac{1}{a_{3}}}} \]
```

```
\[ \sum_{k=1}^n k=\frac{n(n+1)}{2} \]
```

「行列 $\mathbf{A} = \begin{pmatrix} -a & a & 0 \\ a & -a & 0 \\ 0 & 0 & 0 \end{pmatrix}$ ($a > 0$) について、次の間に答えなさい。」を表示させるには、以下のようになります。

```
行列  $\mathbf{A} = \left( \begin{array}{ccc} -a & a & 0 \\ a & -a & 0 \\ 0 & 0 & 0 \end{array} \right)$  ( $a > 0$ )$ について、次の間に答えなさい。
```

上の例で行列を表す大きな括弧を表示させるため、`\left`(と`\right`) を使っています。数式内で括弧をこのように表現すると、 \TeX は括弧内の数式の高さを計算し、括弧の大きさを自動で調節します。

二項演算子 (= など) を揃えて書く場合は、`eqnarray` 環境を使います。例えば下のような数式は、その下に示すようなソースになります。

$$\begin{aligned} I(A \cap B) &= -\log P(A \cap B) \\ &= -\log P(A) - \log P(B) \\ &= I(A) + I(B) \end{aligned} \tag{1}$$

```
\begin{eqnarray} I(A \cap B) &= & -\log P(A \cap B) \ \nonumber \\ &= & -\log P(A) - \log P(B) \ \nonumber \\ &= & I(A) + I(B) \end{eqnarray}
```

`eqnarray` 環境では行の区切りに`\`を用いますが、そのとき自動で式番号が付きます。ある行で式番号を付けたくない場合には、行末記号の前に`\nonumber`と記述します。

本日のやってみよう#2

- 私のホームページの例の場所²にある文書 `short.pdf` を \LaTeX で書いてみよう。
- 新しくディレクトリを作って, `darcs` でバージョン管理しながら作っていくこと。最終的な課題提出物として, バージョン履歴も確認する。
- イタリック体 (数式など) に注意すること。
 - 文中の式は `$` で囲む。
 - 1 行だけで番号付きの式は, `\begin{equation}` と `\end{equation}` で囲む。
- 太字 (ボールド体) になっている個所などに注意すること。
 - 太字にしたい個所を `\textbf{太字にしたい所}` とする。
- 各章の見出しには `\section` を利用すること。
 - 見出しにある章番号などは自動で付くので手で入力しなくてよい。
- 文字としての A, B, \dots と, 数式中の変数としての A, B, \dots の違いに注意すること。
 - `$` で囲むと, 斜字体になるべきものは自動的に斜字体になるので, 明示的に斜字体指定にする必要はない。
 - 本来数式であるものに全角文字を使ってはならない。
- \leq は `\leq`, \geq は `\geq` と数式モード中で入力する。
- 改行や改頁の位置, (数式以外での) 句読点の形 (「。」と「」や「、」と「,」) は気にしなくてよい。
 - 段落を変えたいときは, 空行を入れる。