情報処理 2 第 1 回

アンケートと TEX (1)

桂田 祐史 (かつらだ まさし)

2006 年 4 月 19 日

ホームページは http://www.math.meiji.ac.jp/~mk/syori2-2006/

今週から水曜 5 限、0508 教室で授業を行います。

1 連絡事項

- 先週 1 回やっているし、「この授業で何をやるか、どうやるか」について詳しく説明することはしません。暇なときに前回資料[1] を眺めてください。

「やることは大体シラバスに書いてある通りだが、順番は変える可能性が高く、
1 回インターネット講習会をやるかもしれない。」

- 以下はあえて繰り返します:

1. この講義科目のための WWW ページ http://www.math.meiji.ac.jp/~mk/syori2-2006/ を用意するので、受講中「お気に入り」などに登録することを勧めます。
2. 成績評価はレポートで行います。レポートは電子メールで提出して下さい[2]。
   提出先アドレス syori2@math.meiji.ac.jp
   質問用アドレス mk@math.meiji.ac.jp
   もちろん直接桂田を捕まえて質問しても構いませんが、その場合、メールで日時を約束するのが確実です。
3. 昨年度利用していたユーザーのアカウントについては、ユーザー名、パスワードはこれまで通りでログオン (ログイン) できます。なお、このアカウントを 2006 年度中も使い続けるためには「情報科学センター生田分室 2006 年度利用登録ページ」[3] で更新することが必要です。

[1] http://www.math.meiji.ac.jp/~mk/syori2-2006/jouhousyori2-2006-00/

[2] レポートが届いたかどうか WWW ページで確認できるようにする。ただしアクセスにはパスワードが必要である。パスワードは必要になった時点で口頭で伝える (資料には記さない)。

[3] https://ikuta-m.isc.meiji.ac.jp/cgi-isc/regist/reg_2006.cgi

2　レポート課題 0

(課題というよりはアンケートです。)

情報処理 2 の WWW ページ[4]から、アンケート用紙を入手して、それに書き込みをして、メイルで送信する。

(1) WWW ブラウザーで、情報処理 2 のページにアクセスし、questions-win.txt (Windows で書く場合) または questions-unix.txt (Linux で書く場合) を表示して、「ファイル」メニューから「名前をつけて保存」を選択して保存する。「保存する場所」はとりあえず以下がお奨め。

(a) Windows では「デスクトップ」または「マイ ドキュメント」

(b) Linux ではホームディレクトリィ (“~” = /home/ユーザー名)

(Windows では “z:¥” が Linux のホームディレクトリィ “~” と同じものである。)

(2) アンケートの答を書く。

Windows 上の秀丸で書く場合

(a) スタートメニューから秀丸を起動する。

(b) 「ファイル」から「開く」を選んで、「ファイルの場所」を「マイ ドキュメント」にし、questions-win.txt を選択して編集する。

Linux 上の emacs で書く場合

```
isc-xas06% cd
isc-xas06% emacs questions-unix.txt &
```

(最初のコマンドはホームディレクトリィにいない場合に、ホームディレクトリィにもどるためのもので、不要な場合が多い。)

(3) (2) で書いたファイルをメイルにインクルードまたは添付して、syori2@math.meiji.ac.jp まで送る。Subject: (表題, 件名) は「情報処理 2 課題 0」とする。GraceMail でファイルを添付するやり方は以下で説明する (一度、自分自身に送って実験することを勧める[5])。

[4] http://www.math.meiji.ac.jp/~mk/syori2-2006/

[5] 添付ファイルを開いたら中身が空のようだった、ということが起こるが、添付ファイルを直接開くのではなく、一度ファイルとしてデスクトップにでもセーブして、それから秀丸などで開けば良い。

GraceMail でファイルを添付する方法

1. 「新規」ボタンを押して、メイルを書き始める。
2. 「添付」ボタンをクリックする。
3. 現れた「添付ファイル」ウィンドウで、「参照」ボタンをクリックする。
4. 現れた「ファイルの選択」ウィンドウで、「デスクトップ」または「マイドキュメント」フォルダを選択し、アンケートのファイルを選択し、「一覧に追加」ボタン、「閉じる」ボタンをクリックする。
5. 後は通常と同様にして送信する。

3 (改めて) TEX の紹介

3.1 TEX とは？

(駆け足で説明する。後から加筆する可能性が大きい。)

TEX は著名なコンピューター・サイエンティストである Donald Knuth の開発した文書整形システム (組版システム) である。TEX は日本では「てっく」または「てふ」と呼ばれている。

当初、数式を含む英文を清書することを目的に、組版技術を十分に調べた上で、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。

Knuth 自身は TEX に関する情報を完全に公開していて、ソフトウェアを無償で利用することができる。それを補助、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公開されている。例えば、TEX 本体や周辺ソフトウェアの日本語対応版、日本語フォント (やそれを利用する仕組み)、Windows 環境への移植版、インストーラーなど、すべてインターネットから無償で入手できる (後日説明予定)。

数学者村では標準の文書フォーマットである。理工系の多くの分野で利用されているが、それだけでなく文系の研究者が利用した例もある (発音記号やマイナーな言語などを扱う場合)。

もともと組版技術を十分に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る。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) で採用されている。

それ自身は文書の配布フォーマットとしては適当でないが (表示、印刷に専用のソフトウェアが必要なため)、簡単に PDF に変換できるので、PDF ファイルにしてから配布することができる。

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する。

3.2 Linux 環境での LATEX の使い方

(Windows XP 用の TEX もありますが、情報処理教室にはインストールされていないので... 交渉してみますが、あまり期待しないでください。)

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る。例えば “report1.tex” であれば、

```
icr3-1008% emacs report1.tex &
```

YaTeX のような便利なソフトもあるが、それは後日紹介するとして、ここでは基本的な TEX 関係のコマンドの使い方を紹介する。

.tex ファイルを .dvi に変換するのには platex コマンドを使う[6]。

```
icr3-1008% platex report1.tex
```

“report1.tex” に (LATEX の文法上の) 誤りがなければ、“report1.dvi” という名前のファイルができる。

画面に (組版結果を) 表示する (「プリヴューする」という) には pxdvi コマンドを用いる[7]。

```
icr3-1008% pxdvi report1.dvi &
```

印刷するには次のようにする[8]。

```
icr3-1008% pdvips -f report1.dvi | lp -d プリンター名
```

(もしかすると pdvips -f report1.dvi | double | lp で両面印刷できるかもしれない。未確認であるが。)

dvi ファイルを PDF に変換する方法については後日説明する。

4 レポート課題 1

1 年生の時に情報処理・演習 1 または 2 で作成した C のプログラムを一つ選んで (簡単なものでよい)、以下の内容の TEX 文書を作成し、DVI 形式のファイルに変換して、できあがったファイル (“report1.dvi” とせよ) をレポート受け付け用アドレス syori2@math.meiji.ac.jp まで送信する。Subject は “レポート課題 1” として、メール本文の先頭に学年・組・番号・氏名を書くこと。今回は授業のスタートにてこずったので、締め切りは 4 月一杯またはもっと遅らせる予定である。

1. 文書のタイトルは「課題 1 レポート」とする。学年・組・番号・氏名と日付 (西暦) を忘れずに。

2. それが何をするためのプログラムか説明する。

3. C のソース・プログラムを TEX 文書に取り込む。

[6]現在、日本語が使える LATEX としては、pLATEX が標準的で、コマンドの名前は普通 platex である。

[7]UNIX 環境ではプリヴューアーとして 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。

[8]dvi 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。

4. プログラムのコンパイルの仕方、実行の仕方を説明する。

5. プログラムの実行結果を (少なくとも一つ) TEX 文書に取り込む。結果の簡単な説明を書く (入力データがある場合、それも忘れずに)。

何らかの理由で 1 年生の時のプログラムが一つも残っていないという場合は、以下のプログラムを用いよ (しかし出来る限り 1 年生の時に書いたプログラムを用いること)。何をするためのプログラムか、普通はコメント (注釈) に書くが、このプログラムではわざと書いていない。簡単であるからそれを読み取ること。

sample1.c

```
/*
 * sample1.c --- サンプル・プログラム
 *  (普通は何のプログラムか、目的、使い方など書くものだけど、
 *   それは課題のうちなので、わざと省略します。)
 */

#include <stdio.h>
#include <math.h>

int main()
{
  double a, b, c, D, real, imaginary;

  printf("a, b, c を入力してください: "); scanf("%lf%lf%lf", &a, &b, &c);
  D = b * b - 4 * a * c;
  if (D == 0) {
    printf("%g\n", - b / (2 * a));
  }
  else if (D > 0) {
    printf("%g, %g\n", (- b - sqrt(D)) / (2 * a), (-b + sqrt(D)) / (2 * a));
  }
  else {
    real = - b / (2 * a);
    imaginary = sqrt(- D) / (2 * a);
    printf("%g ± %g i \n", real, imaginary);
  }
  return 0;
}
```

注意 4.1 (参考: ソースプログラム、実行結果等の TEX への取り込み) ソースプログラムやプログラムの実行結果など、いわゆるテキスト・ファイルは短くて完成したもの (頻繁に書き換えないもの) ならば、マウスでコピー&ペーストして verbatim 環境の中に入れれば良いが、そうでない場合は \verbatimfile 命令で取り込むことを推奨する。そのためには、プリアンブル (\begin{document} の前のこと) で

```
\usepackage{verbatimfiles}
```

としておいて、テキスト・ファイル (ここでは “myprog.c” という名前にする) を取り込みた

い場所で

```
\verbatimfile{myprog.c}
```

と書けばよい。 ■

5　最初だから見本を

```
% コメントはパーセント記号 % の後に書く。
\documentclass[12pt,leqno]{jarticle}

\usepackage{verbatimfiles}% 後述の \verbatimfile 命令を使うため

\begin{document}

\title{\textbf{課題 1 レポート}}% 以下 4 行はまとめて一つの指示と思って下さい
\author{2 年 16 組 99 番　桂田　祐史}
\date{2006 年 4 月 19 日}
\maketitle

\section{プログラムの目的}% \section{} は新しいセクションの始まり

(何をするプログラムか説明する。)

\section{ソース・プログラム myprog.c}

\verbatimfile{myprog.c}% これで myprog.c の内容を「そのままの形」で取り込める。

\section{プログラムのコンパイル・実行方法}

(プログラムのコンパイルの仕方、実行の仕方を説明する。)

\section{プログラムの実行結果と説明}

プログラムの実行結果は次のようになった。

% verbatim 環境の中は何でも「そのままの形」で取り込める。
\begin{verbatim}
icr3-1008% ./myprog
a, b, c を入力してください: 1 2 3
-1 ± 1.41421 i
icr3-1008%
```

```
\end{verbatim}

(ここでは verbatim 環境を使ったが、
実行結果を myprog.out という名前のファイルに記録しておいて、
\verbatimfile{myprog.out}
としてもよい。

(実行結果の説明を書く。)

\end{document}
```

6 LaTeX 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。

LaTeX 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{2006 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。

```
\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

  ここにフツー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 TEX 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

  ここにフツー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}
```

A　おまけ: シラバスの写し

2006 年 4 月 19 日現在、Oh-o! Meiji に載せてあるシラバスを引用する (多分、色々変更されるであろうが、逐一 WWW ページでアナウンスする)。

[第 1 回] 電子メールの使い方

電子メールの特徴，使用上の注意を学び，GraceMail を使ってレポートをメールで提出する方法を習得する。特に携帯電話のメールとは異なる特性を理解し使い分けられるようになることが必要である。

[第 2,3,4 回] TeX によるレポート作成法

数学の世界での標準の文書フォーマットである TeX に習熟し，TeX でレポートを書けるようになることを目標とする。

TeX の由来，インストールの方法，TeX に関する情報の入手，PDF 化，各種画像の埋め込み法，スクリーン・キャプチャー，ソース・プログラムや計算結果の挿入法，データのグラフ化などを説明する。

[第 5,6,7,8 回] 「(仮称) 十進 BASIC」による数学的プログラミング

極めて数学向きで，手軽に導入できるフリーの BASIC インタープリターを用いて簡単なプログラミングができるようになることを目標とする。

(仮称) 十進 BASIC の紹介（入手・インストール法を含む)，各種演算モード (10 進 1000 桁，有理数計算，複素数計算) とグラフィックス機能の説明をした後，プログラミング演習を行う。題材としては，非線形方程式の数値解法，連分数，等角写像の図示などを予定している。

[第 9,10 回] Octave による数値線形代数

MATLAB 互換な数値計算インタープリターである Octave を用いて，行列の数値計算の世界を紹介する。

LU 分解の概念，計算量の概念，行列の疎性の利用，行列の解析的性質，線形計算ソフトウェアについて説明する。

[第 11,12,13 回] Mathematica による数式処理入門

代表的な数式処理系である Mathematica を用いて，数式処理を体験する。

Mathematica の電卓的な利用法（特に多項式計算，微分積分，極限)，グラフィックス（2 変数関数の可視化，代表的な曲面の描画)，プログラミングを

取り上げる。