コンピューター実習

桂田 祐史

2018 年 9 月 28 日

http://nalab.mind.meiji.ac.jp/2018/20180928jissyu/

1 今日の実習の内容

- (全員) 桂田から出したメールを読んだか確認 (メールに慣れること)。
- (全員) emacs をインストールする。
- (全員) C 言語の勉強の環境を整える。余った時間はプログラミング実習。
- (MATLAB の利用申請をした人) インストールの相談。

2 C 言語再入門

- C 言語ってどんな言語。
  1972 年誕生 (割と古い)。ベル研デニスリッチーが創った。当初 UNIX OS の記述に使われたことで普及。その後、多くの目的 (たとえば数値計算) に使用されるようになる。
  知っていて損はない。
  コンピューターの機械語に (ある意味で) 近い「低水準言語」。
  言語仕様は国際規格になっていて、何度か改定されている。C99 くらいが相場かも。
- C 言語で書いたプログラムを**コンパイル**して作った機械語形式のプログラムを実行する、という使い方が普通である。

hello.c

```
// hello.c --- 有名な Hello world プログラム

#include <stdio.h>

int main(void)
{
  printf("Hello, world.\n");
  return 0;
}
```

やってみよう (ターミナルで)

```
bash-3.2$ mkdir zemi
bash-3.2$ cd zemi
```

(zemi というディレクトリィを作り、そこに移動する。)

hello.c を作る。emacs を使って作るには

```
bash-3.2$ e hello.c &
```

(通常は emacs というコマンドですが、付録のやり方で設定すると、e 一文字で起動します。emacs で保存は C-x C-s とする。)

```
bash-3.2$ ls
hello.c
  (ls は存在するファイルの名前をリストするコマンド)
bash-3.2$ cc hello.c
  (cc でコンパイルすると、a.out という実行可能プログラムが出来るはず)
bash-3.2$ ls
a.out   hello.c
  (確かに出来ている)
bash-3.2$ ./a.out
Hello, world.
  (カレント・ディレクトリィ . にある a.out を実行すると "Hello, world." と表示される)
bash-3.2$ cc -o hello hello.c
  (実行可能プログラムの名前を -o 名前 で指定できる)
bash-3.2$ ls
a.out   hello   hello.c
bash-3.2$ ./hello
Hello, world.
bash-3.2$
```

- (参考情報: C コンパイラー事情) Mac で使える C コンパイラーとして

  (1) Clang+LLVM (Xcode の cc)

  (2) GCC

  がポピュラー (いずれも無料で導入可能)。現象数理学科 Mac には、(1) はインストール済み。MacPorts がインストールされているので、(2) のインストールも簡単。例えば GCC Version 6 をインストールするには

GCC version 6 のインストール

```
bash-3.2$ sudo port install gcc6
bash-3.2$ sudo port select --set gcc mp-gcc6
```

とする。gcc というコマンドが使えるようになる。
注 1: MacPorts の更新を勧められるかも。時間に余裕のあるときに次のコマンドを実行する。

MacPorts の更新 (時々やっておくこと)

```
bash-3.2$ sudo port selfupdate
bash-3.2$ sudo port upgrade outdated
```

注 2: 実は Xcode に gcc というコマンドがあるが、その正体は Clang+LLVM である。MacPorts でインストールした gcc であることをチェックするには

gcc コマンドは、MacPorts でインストールされた gcc か？

```
bash-3.2$ which gcc
/opt/local/bin/gcc
  (/opt/local/bin/gcc と表示されれば MacPorts でインストールされたもの\\
   /usr/bin/gcc と表示されれば Xcode に含まれるもの)
```

- C のプログラムを入力・編集するためにテキスト・エディターと呼ばれるプログラムが使える。
  ここでアンケート。C のプログラムの作成に何を使っている？
  (1) Xcode (2) mi (3) emacs (4) Atom (5) テキスト・エディット (6) Vim
  (桂田は (3), (6) を使っている。)
  何か一つに慣れておこう。(3) または (4) がおすすめ。
  (1) はプログラミングのための統合環境で、汎用性に問題があるので[1]、テキスト・エディターとしては勧めない。

  – (3) emacs のインストール
    全員インストールしよう。使うか、使わないかは個人の自由だが。Emacs の中から TEX を便利に使う YaTeX とか、Emacs の中でプログラムのデバッグを行う gdb を紹介する予定。

  – (4) Atom については、例えば

    http://nalab.mind.meiji.ac.jp/~mk/knowhow-2018/node20.html

    (ネット上に色々情報がある。)

- (C 言語の勉強) プログラミング言語を学ぶとき「すべてマスターする」などと考えないこと。必要なことを少しずつ確実に、がお勧め。疑問点があったら解消すること。

  「C 言語これくらい覚えよう」
  http://nalab.mind.meiji.ac.jp/~mk/labo/text/cminimum/

  の 3 節 (3.10 くらいまで)、まずはやってみよう (読んでプログラムを入力・実行する — これを各自が出来るようになることを確認)。

  3.11, 3.12 はちょっとなやましいので、後日解説する。

  「4 練習問題」をこの文書の末尾 (最後の 1 枚) に引用しておく。

A 現象数理学科 Mac に Emacs 25.2 をインストール

全員にやってもらう。

(i) まず emacs の入手

https://osdn.net/projects/macemacsjp/releases/p15427

[1] C のプログラムを入力・編集する場合には、特に問題はないが、それ以外のファイルで混乱することがある。

から emacs-25.2-unflickered.zip をダウンロード (1,2 分程度)、ダブルクリックして現れた Emacs.app をアプリケーション・フォルダに移す (Finder でドラッグ&ドロップ)。

(ii) ターミナルから次のようにして、設定ファイル (桂田が用意) をインストール。(bash-3.2$ から後をターミナルにコピペして実行しよう。)

```
bash-3.2$ curl -O http://nalab.mind.meiji.ac.jp/~mk/20181007/dot.emacs.d.tar.gz
  (ネットからファイルを入手する。2 行ほど表示が出る)
bash-3.2$ tar xzf dot.emacs.d.tar.gz -C ~
  (.emacs.d というフォルダを作り、init.el というファイルを置く)
bash-3.2$ echo alias e=/Applications/Emacs.app/Contents/MacOS/Emacs>>~/.profile
  (.profile に別名定義を書き込み)
bash-3.2$ source ~/.profile
```

(2018/10/7 改訂: ¥ キーを押すだけで、バックスラッシュが入力出来るような設定を加えました。)

(iii) アプリケーション・フォルダから Emacs.app を実行出来るか確認。またターミナルで次のようにして実行できるか確認する。

```
e hello.c &
```

(訂正: 配ったプリントには、emacs hello.c & と書いておいたが、上の別名定義は e となっているので、e hello.c & が正しい。)

B　練習問題

各自やってみよう。全員に [8] くらいまでやってもらいたい。

何かを参考にしても、人に尋ねても良い。自分自身が納得して、似たような問題が解けるようになることが大事。

ときどき「どこまで出来ました？」尋ねます。

最後の [11], [12] はそのうち時間を取ってやるかも。

いつでも気軽に (メールでも可) 質問・相談してください。

[1] 自然数 n を入力されたとき、$1+2+\cdots+n$ を計算して、その結果を表示するプログラムを書け[2]。

[2] 実数 a, 自然数 n を入力されたとき、a^n を計算して、その結果を表示するプログラムを書け[3]。

[3] 自然数 k, n を入力されたとき、1^k, 2^k, $\cdots$, n^k を表示するプログラムを書け[4]。

[4] $a_1 = 1$, $a_{n+1} = 3a_n + 2$ $(n \geq 1)$ で定まる数列の最初の 100 項を計算して表示するプログラムを書け (n が大きいときは近似値で構わない — 何桁くらい正しいでしょう？正確に計算するにはどうしたら良いでしょう？)。

[5] $a_0 = 1$, $a_1 = 1$, $a_{n+2} = a_{n+1} + a_n$ $(n \geq 0)$ で定まる数列 (フィボナッチ数列) $\{a_n\}$ に対して、$n = 1, 2, \ldots, 100$ の順に a_n と a_n/a_{n-1} を計算して表示せよ。(n が大きいときは近似値で構わない …)

[番外] a_n/a_{n-1} の極限は何か？ (簡単な数学の問題)

[6] n を与えられたとき

$$S_1(n) := \sum_{k=1}^{n} \frac{1}{k}, \quad S_2(n) := \sum_{k=1}^{n} \frac{1}{k^2}$$

を計算するプログラムを書け。計算結果は `double` の精度一杯まで表示せよ。

[番外] $\lim_{n\to\infty} S_2(n) = \pi^2/6$ であることが知られている。点 $(n, \pi^2/6 - S_2(n))$ を両側対数目盛でプロットすることによって、収束の速さを調べよ。

[7] 自然数 n を入力されたとき、$S_n = \sum_{k=0}^{n} \frac{1}{k!}$ を計算して結果を表示するプログラムを書け ($1/k!$ は漸化式で計算するのが簡単)。

[8] 関数 $f\colon [a, b] \to \mathbb{R}$ のグラフを描くプログラムを書け。ただし f を計算する関数をプログラム中に書くものとする。特に

$$f(x) := \begin{cases} x & (0 \leq x \leq 1/2) \\ 1 - x & (1/2 \leq x \leq 1) \\ 0 & (\text{それ以外}) \end{cases}$$

の場合に $-0.2 \leq x \leq 1.2$ の範囲のグラフを描け。

---

[2] もちろん $n(n+1)/2$ を計算すれば簡単なわけだが、繰り返しの練習なので、正直に 1 から n まで足すこと。

[3] もちろん冪乗を計算する `pow()` を使えば簡単だが、繰り返しの練習なので…

[4] `for` を二重に用いるプログラムを書こう。

[9] (ここまでの応用) 与えられた自然数 n, 実数 x に対して

$$s_n(x) := \sum_{k=1}^{n} \frac{(-1)^{k+1}}{(2k-1)!} x^{2k-1}$$

を計算する関数を書き、$n = 1, 2, 3, \cdots, 10$ に対して $s_n(x)$ $(0 \leq x \leq 2\pi)$ のグラフを描け (「Taylor 級数のプログラミングには漸化式を」)。

[10] $\mathbb{R}^3$ における極座標をデカルト座標 (直交座標) に変換するコードは簡単で、

```
x = r * sin(theta) * cos(phi);
y = r * sin(theta) * sin(phi);
z = r * cos(theta);
```

とすれば良い。この逆をする (つまりデカルト座標を極座標に変換する) 関数 `d2p()` を作って、動くことをチェックせよ[5]。

```
d2p(x, y, z, &r, &theta, &phi);
```

のようにして呼び出せるようにすること。

[11] 常微分方程式の初期値問題を Euler 法、Runge-Kutta 法などの数値解法で近似的に解けることは重要である。以下の初期値問題を解くプログラムを作れ。結果を可視化せよ。

(1) $x'(t) = x(t)$, $x(0) = 1$.
$x(1) = e$ となるはずだが、計算で得た値と比較せよ。

(2) $x''(t) = -g - \gamma x'(t)$, $x(0) = H$, $x'(0) = 0$. ただし $g > 0$, $\gamma \geq 0$ は定数とする。
これは速度に比例する空気抵抗が存在する場合の自由落下を記述する微分方程式である。実は (抵抗が速度の自乗に比例する) $x''(t) = -g - \gamma x'(t)^2$ が正しいという説もある。

(3) $\dfrac{d}{dt}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}$, $x(0) = x_0$, $y(0) = y_0$. ここで a, b, c, d は与えられた定数である。さまざまな (x_0, y_0) に対して解軌道を描け。

[12] 方程式 $\cos x - x = 0$ の実数解を、二分法または Newton 法で計算するプログラムを作れ。

($f(x) := \cos x - x$ は狭義単調減少で、$f(0) = 1 > 0$, $f(1) = \cos 1 - 1 < 0$ であるから、区間 $(0, 1)$ にただ 1 つの実数解を持つことが分かる。)

[5] ちょっと短かすぎて悪い名前だが、Decartes 座標 to Polar 座標、のニュアンス。