

1 C プログラムの作成から実行まで

以下は UNIX 系 OS における方法であるが、Windows 等で行う場合も基本的な流れは同じである。

1. エディタを使ってソースコード (source code) を書き、ファイルに保存する。このファイルをソースファイル (source file) という。C のソースファイルでは、ファイル名の終りに .c を含める。エディタとして GNU Emacs を使う場合、次のコマンド実行する (X Window System が使えない場合は & を打たない)。

```
emacs &
```

Emacs が起動したら、まず作成するソースファイルを開いて (C-x C-f) から、ソースコードを入力し、保存する (C-x C-s)。

2. ソースファイルをコンパイル (compile) して実行可能ファイル (executable file) を作る¹。コンパイルを行うプログラム (コマンド) をコンパイラ (compiler) という。この授業で使うコンピュータには GNU C compiler がインストールされており、起動のためのコマンド名は gcc または cc である。

```
gcc filename
```

コンパイルの途中でエラーメッセージが出たら、1. に戻ってソースファイルを修正する。エラーが無ければ、a.out という実行可能ファイルができる。

3. プログラムを実行し、実行結果を確認する。

```
./a.out
```

プログラム実行時にエラーが発生したり、実行結果に不具合があれば、1. に戻ってソースファイルを修正して保存し、2. のコンパイルも行う。

なお、プログラムに潜む欠陥をバグ (bug) といい、バグを取り除く (プログラムを修正する) 作業をデバッグ (debug) という。

1.1 例題

1. プログラムのコンパイル・修正・実行: 次の内容をもつソースファイル ex1.c を作成し、コンパイルしなさい。

```
/* A first program in C */
#include <stdio.h>

int main()
{
    printf("Hello world!\n");

    return 0;
}
```

¹正確には、コンパイルの後にリンク (link) が行われて実行可能ファイルが出来上がるのであるが、ここでの「コンパイル」という言葉には、リンクのステップも含むこととする。

コンパイル時にエラーメッセージが出ることを確認したら、`printf` で始まる行の行末に ; (セミコロン) を加えてから再度コンパイルし、プログラムを実行しなさい。画面に `Hello world!` と表示されるはずです。

2. プログラム実行時のエラー: 次の内容のソースファイルを `ex2.c` として作成しなさい。

```
/* Runtime error */
#include <stdio.h>

int main()
{
    int i = 2;

    printf("6 / %d = %d\n", i, 6 / i);

    return 0;
}
```

その後、下記の指示に従いなさい。

- (a) ソースファイルをコンパイルし、エラーメッセージが表示されないことを確認してから、このプログラムを実行しなさい。 $6 / 2$ の計算結果が表示されます。
- (b) ソースファイル中で 2 と記述されている箇所を 0 に変更し、ファイル名を変えずに保存しなさい。このソースファイルをコンパイル・実行し、実行時にエラーメッセージが出ることを確認しなさい。

コンパイルに成功しても、作成したプログラムが正しいとは限りません。プログラムの実行時にエラーが起きた場合にも、ソースファイルを修正してコンパイルし直す必要があります。

- (c) 先に変更した 0 を 3 に変えてからコンパイルし、正しく実行できることを確認しなさい。

3. 1. のソースコードを参考にして、画面に自分の学生番号と氏名（ローマ字でよい）を出力するプログラムを作成しなさい。ソースファイル名は `ex3.c` とする。