C 言語の基本的な知識とコンパイルの練習

山本昌志*

2004 年 4 月 13 日

1 本日の学習内容

C 言語の基本的な知識の習得とコンパイルの練習を行う。ここでは、

- プログラム毎のディレクトリーの作成
- エディターを立ち上げてプログラムの打ち込み
- プログラムのコンパイル
- プログラムの実行

ができることが目標である。あわせて、C 言語の非常に基本的なとも理解する。

2 基本事項

2.1 大文字と小文字の区別 教科書 p.7

C 言語では、大文字と小文字は、区別される。変数名 hogehoge と Hogehoge、hoGehoge は異なる。

```
#include <stdio.h>
int main(){
  int hogehoge, Hogehoge, hoGehoge;

  hogehoge = 1;
  Hogehoge = 2;
  hoGehoge = 3;

  printf("hogehoge = %d\n",hogehoge);
  printf("Hogehoge = %d\n",Hogehoge);
  printf("hoGehoge = %d\n",hoGehoge);

  return 0;
```

*独立行政法人秋田工業高等専門学校電気工学科

```
}
```

このプログラムは、次のようになっている。

- `#include <stdio.h>`は、当面おまじないだと思って欲しい。
- `int main()` もおまじない。この後の中括弧 { } の中身が `main` と言う関数の本体である。当面は、この中に書かれたものが実行されると考えて欲しい。
- `int hogehoge, Hogehoge, hoGehoge;` で整数型の変数 (`hogehoge, Hogehoge, hoGehoge;`) を用意している。
- `printf` で変数のデータを書き出している。

これをコンパイルして、実行させると以下のような出力が画面に現れる。これを見ると、大文字と小文字を区別していることが理解できる。

実行結果

```
hogehoge = 1
Hogehoge = 2
hoGehoge = 3
```

2.2 注釈 (コメント文) 教科書 p.11

コメント文は、プログラムの内容をわかりやすくするために記述するものである。これは、人間のためのもので、コンパイラーは無視する。プログラムを維持・管理するときの参考に用いる。良いプログラムは、コメント文が大量に書かれている。FORTRAN の場合、第一カラムが”*”、または”C”の場合、その行はコメント文となるのは、依然学習したとおりである。C 言語の場合は、/*〜*/で囲まれた部分が、コメント文となる。行をまたいでも、それは有効である。

プログラムのデバッグのとき、スキップしたい行をコメントアウトすることはよく使うテクニックである。

```
/* ========================================== */
/* ==    円の面積の計算                       */
/* ========================================== */
#include <stdio.h>
int main(){
  double pi;
  double r, s;

  pi = 3.141592;           /* 円周率 */
  r = 1.0  /* 円の半径 */

  s = pi*r*r               /* 円の
     次の行にまたがっても良い 面積計算 */
```

```
    printf("s = %f\n",s);    /*出力*/

    return 0;

        }
```

2.3 識別子 教科書 p.12

識別子とは、変数、記号定数、関数などにつける名前のことである。名前に用いることが出来る文字は決まっており、以下のとおりである。

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r t t u v w x y z
0 1 2 3 4 5 6 7 8 9 _（下線、アンダースコアー、アンダーバー）

2.4 全角空白と半角空白 教科書 p.13

全角の空白と半角の空白は、まったく異なる。全角の空白は、日本語を表示する時以外、使うことは有りません。全角の空白を書くと、非常に分かりにくいバグの原因となることがあるので極力使わない方がよいであろう。

2.5 フリーフォーマット 教科書 p.13

FORTRAN は、7 カラム目から、プログラムは記述するという約束がる。しかし、C には、どこからでも、プログラムを書くことが出来るので便利である。そのため、プログラムは、字下げをしてわかり易い記述ができる。字下げは、[Tab] を使うと自動的にできる。字下げを上手に使って、わかりやすいプログラムを書くことが、上達の早道である。

```
#include <stdio.h>

int main(){

  int i, j;
  int kai;

  for(i=1; i <= 5, i++){
    kai = 1;

    for(j=1; j <= i, j++){
      kai = kai*j;
    }

    print("%d の階乗 = %d\n",kai);
```

```
  }

  return 0;
}
```

実行結果

```
1 の階乗 = 1
2 の階乗 = 2
3 の階乗 = 6
4 の階乗 = 24
5 の階乗 = 120
```

2.6 セミコロン

C はフリーフォーマットで記述できますので、文の区切りの記号が必要である。その区切りの記号にセミコロンを用いる。

3 コンパイルの練習

3.1 最小のプログラム (nothing.c)

何も、実行しないプログラムである。

```
main(){
}
```

main は関数名。C のプログラムの中に、必ず、1 つ必要。関数名のあとの ( ) の中に、引数を書く。引数とは、その関数の変数みたいなのもの。これについては、後の授業で説明する。

コンパイル `gcc -o nothing nothing.c` あるいは、`gcc nothing.c`
実行 `./nothing` あるいは、 `./a.out`
結果 なんにも、起こらない。

3.2 1 行出力 (hello_world.c)

画面に、1 行出力させる。この、Hello World のプログラムは世界で、一番、書かれているプログラムである。

```
#include <stdio.h>

int main(){
```

```
  printf("Hello World !!");

  return 0;
}
```

#include <stdio.h> これは、ヘッダーファイルと呼ばれるもので、関数のプロトタイプがかかれている。プロトタイプとは、関数の引数の数と型をチェックするものである。ここでは、printf 関数を使うため、これが必要である。とりあえず、おまじないと思って、常に先頭に書く習慣をつける。

printf() は、名前のとおり、標準出力に出力する関数である。標準出力とは、通常、ディスプレイが割り当てられている。

コンパイル	gcc -o world hello_world.c あるは、gcc hello_world
実行	./world あるいは、 ./a.out
結果	Hello World !!

3.3 2 行出力 (hello_akita.c)

画面に、2 行出力させる。1, 2 行目に改行マークがある。

```
#include <stdio.h>

int main(){

  printf("Hello World !!\n");
  printf("from Akita National College of Technology !!\n");

  return 0;
        }
```

printf の中の \n エスケープシーケンスと言われ改行を示す。重要な拡張表記については、教科書の p.28 に示されているので、見るのがよいだろう。通常、よく使用されるのは、\n と $backslash$t である。

3.4 足し算 (add.c)

π+e の計算を行い、その結果を示す。

```
#include <stdio.h>

int main(){
  double a, b, c;

  a = 3.1415927;
  b = 2.7182818;
  c = a+b;
```

```
  printf("pi = %10.7f\n",a);
  printf("e = %13.9f\n",b);
  printf("pi+e = %20.12f\n",c);

  return 0;
}
```

double によって、変数の型を指定している。double は、倍精度実数型である。それ以外のデータの型については、教科書の P.34 に示されているので確認すること。

%10.7f は、書式つき出力を示している。printf("pi = %10.7f\n",a) は、a を 10.7f 書式で出力する命令。f は浮動小数点のこと。10.7 の 10 はフィールド幅を示し、7 は小数点以下の桁数を示している。フォートランと同じ。

3.5 ファイルへの書き込み (outf.c)

sample.txt というファイルを作り、その中に、Hello World と書く。

```
#include <stdio.h>

int main(){
  FILE *out_file;

  out_file = fopen("sample.txt", "w");
  fprintf(out_file, "Hello World");
  fclose(out_file);

  return 0;
}
```

FILE は、変数の型指定みたいなも。out_file の型がファイルを示す。out_file の前についている*は、ポインターをしめす。これについては、おまじないと思って、ファイルを使う場合は、FILE *filename; と記述すればよい。

fopen は、ファイルを開く関数です。引数は、ファイル名とオープンモードである。オープンモードについては、教科書の p.382 に書かれている。

fprintf() は、printf() とほとんど同じ。第一引数に、出力先を指定する。もし、stdout と指定すると、標準出力 (通常ディスプレイ) に出力される。

fclose は、ファイルを閉じる関数である。

プログラムをコンパイル・実行の後、作成されたファイルを調べよ。

3.6 ファイルからの読み込み (inf.c)

sample.txt というファイルを読み込み、その内容を出力する。

```
#include <stdio.h>

int main(){
  FILE *in_file;
  char ss[256];

  in_file = fopen("sample.txt", "r");

  fgets(ss, 256, in_file);
  printf("%s",ss);

  fclose(in_file);

  return 0;
}
```

char ss[256] は、文字型変数、256 バイトのメモリー領域を確保しろという命令である。

fgets 関数で、ファイルの中身を読み込みむ。最初の引数は読み込んだ文字の格納領域、次は最大読み込む文字数、最後は読込先のファイルポインターである。