関数 printf() と変数

山本昌志*

2004 年 5 月 21 日

1　本日の学習内容

本日の学習は、教科書の p.56 の 2.3.3「結果を表示する関数 printf() の詳細」から、p.68 の「プログラムを実行しながら、さまざまな結果を得るには?」の前までです。具体的には、以下のことについて、学習します。

- 結果を表示する関数 `printf()` の使い方
- 計算結果の表示の仕方
- 計算のための数値を入れる変数について

2　これまでの復習

2.1　プログラムの作成順序

これから、諸君は多くのプログラムを作る。それらは、分かりやすく分類しなくてはならない。そのため、それぞれのプログラム用にサブディレクトリーを作成し、その中にプログラムを入れておくことが望ましい。サブディレクトリーの作成から、ソースファイルの作成、コンパイル、実行が一連のプログラム作成の手順である。ここで、

サブディレクトリー名	work
ソースファイル名	hogehoge.c
実行ファイル名	fugafuga

とした場合の具体的なプログラム作成方法を示す。ただし、名前は適当に変えてよい。

1. サブディレクトリーの作成と移動
    - コマンド「`mkdir work`」と入力し、サブディレクトリーを作成する。
    - コマンド「`cd work`」 と入力し、作成されたサブディレクトリーに移動する。

*国立秋田工業高等専門学校　電気情報工学科

2. ソースファイルの作成

   - コマンド「`iedit hogehoge.c`」と入力する。
   - エディターの編集画面が出てくるので、プログラムを打ち込む。
   - プログラムを打ち込み終わると、その内容のセーブ (保管) する。

3. ソースプログラムのコンパイル

   - コマンド「`cc -o fugafuga hogehoge.c`」と打ち込み、ソースファイルをコンパイルする。このコマンドで、ソースファイル「hogehoge.c」を翻訳して機械語の「fugafuga」が出来上がる。
     - 「`cc`」がコンパイルをしなさいという命令である。
     - 次の「`-o`」は出力先を示すオプションで、空白を空けた直後に機械語に翻訳される実行ファイル名を書く。
     - 最後に、ソースファイル名を書く。
   - もし、エラーがあればソースプログラムを再編集して、保管、コンパイルを行う。

4. 実行ファイルの実行

   - ターミナルで、実行ファイルがあるか確認をする。コマンド「`ls`」を打ち込むと、実行ファイル「*」が表示される。ファイル名の後ろにアスタリスクが付いているのは、実行ファイルの印である。
   - コマンド「`fugafuga`」と打ち込こむことにより作成したプログラムを実行できる。
     - もし、作成した実行ファイル名が UNIX のコマンドと同じ場合、UNIX コマンドが優先されて実行できない。
     - 「`./fugafuga`」のように強制的に、カレントディレクトリーを指定して、実行させる。

2.2 何かを表示するプログラム

これまでに学習したプログラムは、printf() 関数を使って、ダブルクォーテーションの間に書かれた文字をディスプレイに書き出すプログラムでった。それは、次のようなものである。

```
 1  #include <stdio.h>
 2
 3  int main()
 4  {
 5    /* 表示をする */
 6    printf("Yes My Master\n");
 7    printf("May I help you\n");
 8
 9    return(0);
10  }
```

このプログラムの各行の役割は、次の通りである。コンピュータープログラムは、各行がきちんとした役割を持ち、厳密に動作することを理解しなくてはならない。

- 1 行目は、今のところ、おまじないと思って、C 言語のプログラムの最初に書くものだと考える。これでは、気がすまない人は、以下を読むこと。
  - `#include <stdio.h>` は、`printf()` 関数等の定義が書かれているヘッダーファイル stdio.h を挿入する。
- 3 行目は関数 `main()` を示しており、4 行目の { から 10) 行目の } がその範囲である。`int` は、integer(整数) の略で、その関数が実行されると、整数の値が返されるということです。その整数の値は、9 行目の `return` 文で 0 を返している。しばらくは、3 と 4、9、10 行目はおまじないとして、この通りに書くのが良いであろう。
- 5 行目はコメント文 (注釈文) である。`/*`と`*/`で囲まれた部分は、`/*`と`*/`を含めコンパイラーは無視をする。要するに、機械語に翻訳されないので、プログラムの動作には全く影響しない。その目的はプログラムの内容をわかり易くするために書く。そうすると、
  - 保守性がよくなる。
  - 拡張するときに、参考になる。
  - プログラムミスの発見がしやすい。

  など、有益である。
- 6 行目と 7 行目の `printf()` 関数で、画面にメッセージを書き出している。この関数は、括弧内のダブルクォーテーション内の文字をそのまま、ディスプレイに書き出す。ただし、`\n` はエスケープシーケンスと言って、これは改行を表す。

以上がプログラムの内容の全てである。実際にプログラムが動作している部分 (実行文) は、5 と 6、9 行目であるが、それを実行させるために、いろいろ準備が必要である。これ以外の行は、その準備のために必要である。

あとはプログラムは一目見て分かりやすいように、空行 (2 と 8 行) を入れたり、インデント (5〜9 行) を行っている。空行やインデント[1]が無くてもプログラムは動作するが、プログラマーの処理の塊がすぐに分かるようにすることは重要である。長いプログラムを書くときに、ミスをかなり減らすことができる。

3 結果を表示する関数 printf() の詳細

3.1 簡単な計算 (教科書 List 2-5)

プログラムの説明の後、以下を実行せよ。

- 適当なサブディレクトリーを作成せよ。

[1] Tab キーを押すことによりインデントができる。

- そのディレクトリーの中に移動せよ。
- そこで、教科書の p.57 の List 2-5 のプログラムソースを作成せよ。今までとは異なり、エディターは、「`iedit hogehoge.c`」として立ち上げよ。この方が、アイコンをダブルクリックするより便利である。ただし、`hogehoge.c` はソースファイル名なので、適当に変えよ。
- 出来上がったソースファイルをコンパイルせよ。
- 実行ファイルを実行させて、教科書と同じ出力であることを確認せよ。

以上の課題が完了したら、`1+2` の部分を変えて、計算結果を確かめよ。この和の計算は、どこの部分で指定しているか良く考えること。

3.2 出力変換仕様 (教科書 List 2-6)

プログラムの説明の後、先ほどと同様に教科書 p.58 の List 2-6 のプログラムのソースファイルを作成せよ。そして、コンパイル・実行を行い、教科書と同じ結果が得られることを確認せよ。

以上の課題が完了したら、計算内容を変えて、出力変換仕様 (%d, %f %lf 等) の動作を理解せよ。

4 値を記憶しておく箱を利用して結果を求めて、表示する

教科書 p.61 の List 2-7 のプログラムを実行せよ。実行結果は、明らかに問題がある。どのようにすれば、問題が無くなるか考えよ。

教科書の p.62 の List 2-8 のプログラムを実行せよ。ここで、変数というものをしっかり理解する必要がある。変数 `total` の役割を考え、その動作を理解せよ。

教科書の p.66 の List 2-9 のプログラムを実行させよ。ここでは、変数と変数の型をしっかり理解する必要がある。いろいろプログラムを変えて、それを理解すること。