

C言語の学習

標準ライブラリー関数・その他のライブラリー関数

山本昌志*

2004年6月23日

1 本日の学習内容

本日の内容は、教科書の16章と17章である。コンソール入出力と数学関数について、学習する。

16章 コンソール入出力関数

17章 入出力以外の標準ライブラリー関数

2 コンソール入出力関数(16章)

コンピューターのもっとも基本的な入出力装置であるキーボードとディスプレイをコンソール(操作卓)と呼ぶ。これらのコンソール入出力を利用した関数のプログラムの学習をする。とは言え、諸君はこれらの関数はかなり使ってきている。ここではそれら内容を整理して、これらの使い方の復習を行う。

2.1 標準ライブラリー関数(p.282)

2.1.1 プログラマー作成の関数とライブラリー関数

これまでの学習で、C言語のプログラムは関数の集まりであることが理解できていると思う。この関数には、プログラマーが作成する関数とともともとコンパイラーが用意している関数がある。例えば、リスト1のプログラムでは、

- `printf()` は、コンパイラーが用意している関数
- `main()` と `plus_print()` はプログラマーが作成した関数

である。

*独立行政法人 秋田工業高等専門学校 電気工学科

リスト 1: 加算結果を表示。

```

1 #include <stdio.h>
2
3 void plus_print(int i, int j);
4
5 /*=====*/
6 /*  main 関数                               */
7 /*=====*/
8 int main(void){
9
10     plus_print(1, 4);
11
12     return 0;
13 }
14
15 /*=====*/
16 /*  plus_print 関数                           */
17 /*=====*/
18 void plus_print(int i, int j){
19     int sum;
20
21     sum = i+j;
22
23     printf("%d+%d=%d\n", i, j, sum);
24 }
25

```

2.1.2 標準ライブラリー関数とコンパイラ依存のライブラリー関数

これまでの学習で、`printf()` や `scanf()` など、コンパイラが用意している関数をしばしば使用した。これらの関数を、一般にライブラリー関数と呼ぶ。ライブラリー関数には、ANSI C で定められている関数と、それ以外に処理系ごとに特別に用意された関数とに分けることができる。前者は、ANSI C の規格にもとづいた C 言語のコンパイラには必ず用意されている関数で、`printf()` や `scanf()`、`sin()`、`cos()` などがこれに含まれる。一方、後者はコンパイラに依存する関数である¹。例えば、マイクロ秒単位でプログラムの実行を停止する関数 `usleep()` が、これにあたる。例えば、リスト 2 のプログラムでは、

- `printf()` は、標準ライブラリー関数
- `usleep()` は、コンパイラに依存するライブラリー関数

である。

リスト 2: 標準ライブラリー関数でない `usleep` を使った例

```

1 #include <stdio.h>
2 #include <unistd.h>
3
4 int main(void){
5     int i;
6     unsigned long int t;
7
8     t=1;

```

¹コンパイラが異なると、その関数は使えない可能性がある。

```

9
10     for ( i=1; i < 23; i++){
11
12         t*=2;
13
14         usleep (t);
15         printf ("%lu\tHello World\n" ,t);
16     }
17     return 0;
18 }

```

2.1.3 まとめ

これまでの関数の種類について、まとめると次のようになる。プログラムの構成を関数の種類から考えると、図1のように分類できる。

- C 言語のプログラムは関数から構成される。
- プログラムで使われる関数は 3 種類ある。
 - プログラマーが作る関数
 - 全ての C 言語のコンパイラーが用意している ANSI C で規格が決められている標準ライブラリー関数
 - コンパイラーに依存するライブラリー関数

本講義では、標準でないコンパイラーが独自に用意した関数を使うことはない。諸君は、教科書の付録 (p.408-466) に書かれている標準ライブラリー関数を使って、プログラムを作成することに心がけよ。

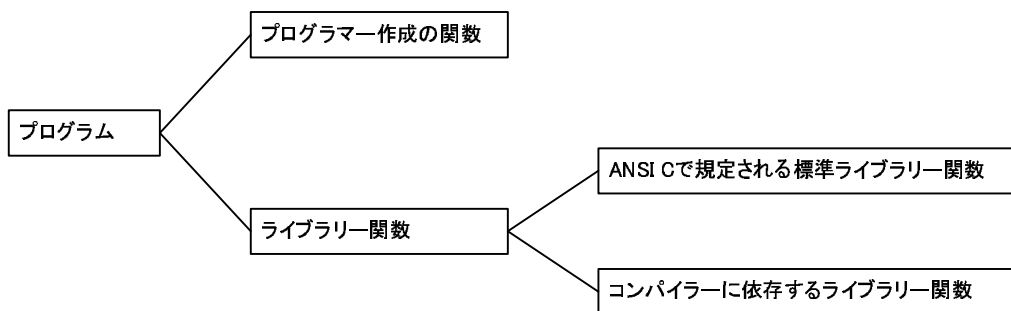


図 1: C 言語のプログラムで使われている関数の分類

[練習 1] 標準ライブラリーにどのようなものがあるか、教科書 p.408 ~ 466 を軽く眺めよ。

2.2 標準入力と標準出力

コンソール入出力のことを、標準入出力と言うことがある。この標準入出力には、ファイルの入出力先に指定ができ²、そのために名前が付いている。標準入力を `stdin`、標準出力を `stdout` と言う。これらの関係を、以下に示す。

```
キーボード → 標準入力 → stdin
ディスプレイ → 標準出力 → stdout
```

2.3 書式付き出力 printf(p.320)

書式付き出力 (`printf`) を使うと、任意の形でデータを出力できる。教科書の P.320 ~ 325 を読み、以下の練習問題を実施せよ。

[練習 1] 円周率を、いろいろなフォーマットで出力せよ。ただし、円周率は `math.h` の中で `M_PI` で定義されている。それは、リスト 3 のようにすれば表示できる。

- 通常の `%lf` で表示せよ。
- 小数点以下、3 桁で表示せよ。
- 小数点以下、10 桁で表示せよ。
- 指数形式で表示せよ。

[練習 2] 整数の 22446688 を 8 進数で表示せよ。また、16 進数で表示せよ。

[練習 3] 円周率と円周率の 2 乗の両方を小数点以下、10 桁で表示せよ。ただし、2 つの数値の間はタブ区切りとする。

リスト 3: π の表示

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void){
5
6     printf("%lf\n", M_PI);
7
8     return 0;
9 }
```

2.4 書式付き入力 scanf(p.326)

キーボード入力の場合、書式付入力の `scanf` という関数を使う方法が最も簡単である。この関数の引数は、ポインタ³で、キーボードからのデータを入れる変数のアドレスを指定する。

データを入力する場合、データを入れた後に、完了を示す [Enter] キーを押す。この [Enter] キーをしめす文字 `\n` も読み込まれようとして、悪さをすることがある。この、改行コード `\n` を読み捨てる必要がある。そのためには、4 通りの方法がある。

²C 言語では、キーボードやディスプレイもファイルとして取り扱われる。

³値を代入する変数のアドレスのこと。

- ダミーの文字型変数を用意して、それに改行コード'\n'読み込む方法。
- 代入を抑止する方法。
- fgets で読み込む方法。
- rewind 関数⁴でバッファの内容を捨てる。

私は、最初の方法をよく使いますが、どれを使っても良い。

以下に、実際に、書式入力 (scanf) を使ったキーボード入力を示す。

2.4.1 1文字の場合

キーボードから1文字を入力し、変数 a に格納する。変数宣言は

```
char a;
char temp;
char dmystr[128];
```

とする。[Enter] キーの問題を回避した4通りの入力方法は、以下の通りである。

- `scanf("%c%c", &a, &temp);`
- `scanf("%c%c", &a);`
- `scanf("%c", &a);`
`fgets(dmystr, 128, stdin);`
- `scanf("%c", &a);`
`rewind(stdin);`

2.4.2 文字列の場合

キーボードから文字列を入力し、配列 a[] に格納する。変数宣言は

```
char a[128];
char temp;
char dmystr[128];
```

とする。[Enter] キーの問題を回避した4通りの入力方法は、以下の通りである。配列名はアドレスを示すので、&をつける必要はない。文字列の中に空白があると `scanf()` は使えない。空白はデータの区切りと解釈され留殻である。そのときは、`gets()` を使う。

- `scanf("%s%c", a, &temp);`
- `scanf("%s%c", a);`

⁴ファイル位置指定子を先頭に設定する関数

- `scanf("%s", a);`
`fgets(dmystr, 128, stdin);`
- `scanf("%s", a);`
`rewind(stdin);`

2.4.3 整数 (10 進数) の場合

キーボードから整数を入力し、変数 a に格納する。変数宣言は

```
int a;
char temp;
char dmystr[128];
```

とする。[Enter] キーの問題を回避した 4 通りの入力方法は、以下の通りである。

- `scanf("%d%c", &a, &temp);`
- `scanf("%d%c", &a);`
- `scanf("%d", &a);`
`fgets(dmystr, 128, stdin);`
- `scanf("%d", &a);`
`rewind(stdin);`

2.4.4 倍精度実数の場合

キーボードから倍精度実数を入力し、変数 a に格納する。変数宣言は

```
double a;
char temp;
char dmystr[128];
```

とする。[Enter] キーの問題を回避した 4 通りの入力方法は、以下の通りである。

- `scanf("%lf%c", &a, &temp);`
- `scanf("%lf%c", &a);`
- `scanf("%lf", &a);`
`fgets(dmystr, 128, stdin);`
- `scanf("%lf", &a);`
`rewind(stdin);`

2.4.5 2個以上読み込む場合

キーボードから2個の倍精度実数を入力し、変数 a と b に格納する。変数宣言は

```
double a, b;  
char temp;  
char dmystr[128];
```

とする。[Enter] キーの問題を回避した4通りの入力方法は、以下の通りである。

入力データの区切りは、空白あるいは tab、改行です。

- `scanf("%lf%lf%c", &a, &b, &temp);`
- `scanf("%lf%lf*c", &a, &b);`
- `scanf("%lf%lf", &a, &b);`
`fgets(dmystr, 128, stdin);`
- `scanf("%lf%lf", &a, &b);`
`rewind(stdin);`

[練習 1] 2つの整数をキーボードから読み込み、その和と差、積、商を表示せよ。

[練習 2] 整数と倍精度実数をキーボードから読み込み、積を指数形式と小数点以下8桁で表示せよ。

3 入出力以外の標準ライブラリー関数 (16章)

3.1 数学関数 (p.351)

教科書 p.351 の表に書かれているような数学関数を使うときには、ヘッダーファイル<math.h>を以下のように書いて、インクルードする必要がある。

```
#include <math.h>
```

さらに、さらにコンパイルする場合には、libm というライブラリーをリンクして、実行ファイルを作成する必要がある。このライブラリーが数学関数の実行ファイルである。数学関数が使われているソースコードをコンパイルするときには、

```
gcc -lm -o 実行ファイル名 ソースファイル名
```

のようにする。

[練習 1] 倍精度実数を読み込み、以下の関数の値を指数形式と小数点以下、8桁で表示せよ。

- 三角関数 (sin, cos, tan)
- 指数関数
- 自然対数関数と常用対数関数
- 平方根と立方根

[練習 2] π^π を計算して、指数形式と小数点以下、8桁で表示せよ。

3.2 数学に関係すること諸々

3.2.1 複素数

以前の C 言語は複素数がサポートされていなかった。数値計算をする場合、複素数が使えないとかなり不便を強いられる。そのため、複素数が使える FORTRAN から抜け出せない人が多くいた。新しい C 言語では、複素数がサポートされるようになってきている。

その使い方は、教科書の p.471 を参考にして欲しい。

[練習 1] $1 + i$ の複素数の絶対値を計算せよ。複素数の絶対値を計算する関数は `cabs()` である。

[練習 2] $2 + i\pi/4$ の三角関数の値を計算せよ。複素数の三角関数は、`csin()`、`ccos()`、`ctan()` を使う。

3.2.2 乱数

乱数も数値計算ではよく使われる。乱数の発生については、教科書の p.440(`rand`) と p.449(`srand()`) を見よ。

[練習 1] 諸君が使っているシステムの乱数発生のプログラムで、発生できる最大の乱数を調べよ。

[練習 2] 乱数の一様性を調べたい。どうすれば良いか考えよ。