マイコンを正しく操縦するための作法

基礎から学ぶC言語講座

岡田 好一
Yoshikazu Okada

第5回 関数とマクロを攻略しよう

今回は，C言語プログラムに欠かせない関数と関数に外見がよく似ているマクロについて解説します．

関数とは

プログラム中で同様の処理が別の場所に出てきたときに，まとめて記述する仕掛けが関数やサブルーチンです．印字や三角関数値など，ある特定の機能を使いたいときに，関数名やサブルーチン名を書いておけば，指定された機能を「呼び出す」ことができます．

計算機言語では両者はほぼ同等で，関数が「値」を返すもの，手続き/サブルーチンは値を返さないものです．C言語には関数しかなく，値が不要な場合は，形式的に`void`型の値を返す関数として記述するか，使う側で返り値を無視します．

● 関数の主作用と副作用

計算機言語に慣れてしまうと上述の説明は素直に頭に入ると思います．しかし，もともと関数とは，

$$y = ax^2 + bx + c$$

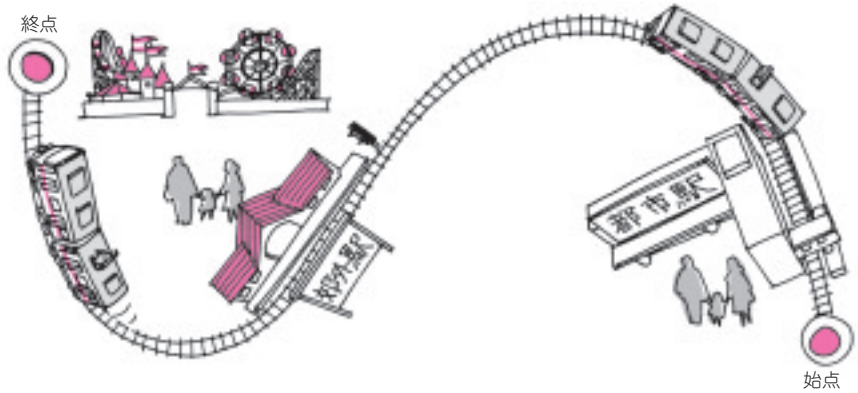

図5-1 関数の主作用と副作用
列車が始発駅から終着駅まで進むのが主作用，乗客が目的地まで行くのが副作用

Keyword 1 関数(function)

数学的には定義域の値から値域の数値を対応させることです．この定義では，入力は任意の集合の要素であればよく，例えば平面図形など，数値とは限りません．出力は通常，複素数まで含まれます．私の記憶では，本来の計算機言語の関数も同様でした．

C言語には，形式的には手続き/サブルーチン(procedure/subroutine)がないので，その代わりとして副作用主体の関数が使われます．

R8Cをはじめとする16ビット以上のCPU(CISC)では，関数呼び出しで使われる局所変数のためのスタック・フレームが機械語レベルで支援されます．コンパイラも関数呼び出しにはハードウェア支援のスタック・フレームを利用します．

アセンブラでは単純なcall/return，つまり手続きサブルーチン風の利用が主体です．マイコンを状態遷移的に使うのなら，こちらで十分です．最適化を指定すると，コンパイラは可能な範囲でフレーム生成を避けるようです．

みたいな数学上の概念です．機能は「写像」で，定義域の値から，数値を対応させることでした．C言語でも数学関数としてsin(x)やatan2(x, y)が用意されています．もちろん，計算機言語でも本来の関数はこちらです．引き数から値に変換するのが関数の「主作用」です．

C言語の学習者が最初に目にする関数printf()は出力を期待する関数です．値を返す以外の機能，つまり代入や入出力は関数の「副作用」と呼ばれます．イメージで言えば，列車が始発駅から終着駅まで進むのが主作用，乗客が目的地まで行くのが副作用，といった感じです(**図5-1**)．

つまり，関数には副作用を期待するものがあります．副作用主体の場合，働きは手続き/サブルーチンそのものなので，以下，サブルーチンの用語も使います．

● 関数の宣言

変数に宣言が必要なように，関数も使用する前に宣言が必要です．先ほど出てきたsin関数なら，math.hファイル内に，

```
double sin(double x);
```

という宣言があります．使う場合には，この1行が参照できるように，

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

とのプリプロセス・コマンドが必要です．この位置に，math.hファイルの内容がごっそりコピーされます．これで，

```
y=sin(x);
```

などと，関数を使うことができます．

宣言からわかるのは，sin関数は引き数に倍精度浮動小数点数が1個要求され，返り値の型も倍精度浮動小数点数だということです．関数の型は返り値の型です．

sin関数はライブラリ関数なので，自分で定義を書く必要はありません．リンカはライブラリから適切に該当のコードを抜き出して実行形式(最終コード)に埋め込みます．

▶ 可変数の引き数を持つprintf関数

printfライブラリ関数も見ておきましょう．stdio.h内に次の宣言があります．

```
int printf(const char _far *format, ...);
```

最初の引き数はchar型へのポインタです．ポインタの先のchar型データをprintf関数が書き換えないことを示すconstが付いています．_farは，通常はfarポインタ，つまり20ビット・アドレスが使われる，ということです．単なるfarでないのは，内部的な使用だからです．

printf関数は引き数の数がformatの指す文字列の内容で変化するので，第2引き数のところに「...」と書かれています．ここに0個以上の引き数が書かれます．可変数の引き数をもつ関数を作る場合は，stdarg.h内の宣言を利用します．可変数の引き数の関数を作ることはほとんどないと思われるので，本連載では作り方を省略します．

printf関数の型はintです．書き出された文字数，またはエラーを返します．何か作業する目的の関数は，このように，結果を知らせる値，つまりリターン・コードを返す設計が多いようです．

● 関数の型

関数の型，つまり返り値の型はvoid，整数，浮動小数点数，ポインタにします．配列は返せません．構造体も可能ですが，大きなデータを返すよりはポインタが好まれます．

voidは値を返さない関数の宣言です．値を利用しなければ同じことになるので，初期のC言語ではとりあえずint型で宣言していました．値を返さないのですから，周辺機器に出力，大域変数へ代入，時間つぶしなどの「副作用」があるはずです．

Keyword 2　副作用(side effect)

計算機言語における関数の副作用とは，値を求める以外の入出力や大域変数への代入を指します．言葉とは裏腹に，望んだはずの効果です．

副作用が目的の関数では，値を求める主作用はソフトウェアの推進役，つまり裏方に回ります．値を求める方向に機械は動作し，求まった時点で仕事は完成しています．

ただし，印字など，実用的な副作用には決定論的な動作，つまり1回に限り成功する動作が求められます．また，一般に複数の副作用の時間の順序はたいせつです．関数の値のように，試行錯誤してでも，どのような経路をたどっても，いつか正しい答えが算出されれば良い，というものではありません．

したがって，副作用が正しく行われるためにはある時点で過去に戻れなくなる，つまり同期の概念が必要です．私の知る範囲では，この点が美しく解決されている計算機言語はありません．計算機がなんとなく実時間処理に向いていない感じがする理由と思います．

副作用のない言語は理論検証目的には存在し，ながめるにはおもしろいですが，ほとんど役立たないようです．