

C言語の文法を本質から理解しよう!

森 孝夫

C言語も言葉の一つである。文法を理解しなければ正しく動くプログラムは記述できない。本章ではC言語の基本文法を基礎から解説する。理解が難しいとされるポインタと配列については、練習問題を使用して実際に手を動かしながら理解を深める。
(編集部)

C言語は、シンプルで理解しやすい文法構造を備えています。さらに、関数や構造体のようなモジュール化やデータを抽象化するしくみと、ハードウェアに近い低階層のプログラミングを可能にするしくみも持っているという特徴があります。

本章では、C言語の基本文法と、組み込みプログラミングによく使われる表現を紹介します。

1. 変数——メモリ上に確保されるデータの入れ物

変数は、メモリ上に確保されるデータの入れ物で、プログラムの基本的な構成要素です。

組み込みシステムではリソースに制限があるため、メモリを無駄に使用できません。変数の大きさもプログラム作成時に最適なものを選ぶ必要があります。

● 型——データの種類や範囲を規定する

変数を使用するときは、型を宣言する必要があります。型とはデータの種類やその値の範囲を規定するものです。基本となる型を表1に示します。

表1 主要な型

型	内容
char	文字型 (Character). 8ビット幅
int	整数型 (Integer). 通常、処理系の標準の大きさとなる
float	単精度浮動小数点数
double	倍精度浮動小数点数
short	処理系の標準より短い、もしくは同じ幅を持つ整数型
long	処理系の標準より長い、もしくは同じ幅を持つ整数型

文字型 char や整数型 int に関しては、符号付きと符号なしを修飾子 signed と unsigned で指定します (表2)。たとえば符号なし整数の型は、

```
unsigned int
```

のように記述します。なお、下記のように単に型名だけを記述した場合に、signed となるか unsigned となるかは処理系 (コンパイラ) によって変わります。

```
int
```

また、あるコンパイラにおける変数型のビット幅や最大値、最小値は、通常、標準ライブラリの limits.h に記載されています。limits.h は、組み込みプログラミング用のコンパイラでも定義されていることがほとんどです。表3に、limits.h に記載されている文字型と整数型の最大値と最小値の定義を示します。

● 宣言——型や性質を明示する

宣言は、変数や関数の型や性質を明示するためのものです。変数は、使用前に必ず宣言されなければなりません。たとえば、

```
a = b + 20;
```

という命令文の前には必ず、

```
int a, b;
```

という宣言が必要です。宣言がないと、コンパイラは使用している変数の型などがわからないのでコンパイルができ

表2 符号を表す修飾子

修飾子 (qualifier)	内容
signed	符号付き。通常、最上位ビットが符号ビットとして使用される
unsigned	符号なし。常に0以上の値をとる