

» 文法の曖昧さを理解して確実性と再利用性を高める

# マイコンC言語 転ばぬ先のつえ

第10回 演算子①…「関数呼び出し」に使う小括弧

鹿取 祐二

本連載では、C言語の文法の曖昧な部分と、それにより起こる問題を解説します。再利用性と効率が高く、安全かつ安心して使えるソフトウェアが開発できるようになることを目指します。

今回からは、演算子に関する注意すべき項目や一見不思議な文法を紹介します。表1は、優先順位の高いものからC言語の演算子を並べています。中には本連載で紹介しないものもあります。本連載では、優先順位の高い演算子から紹介します。

## 16 演算順を決めるものとは別物! 演算子の小括弧は「関数呼び出し」に使う

### ■ こんな演算子

#### ● 演算の順序を決める小括弧とは違う

筆者の主観ですが、C言語プログラムの10～20%が間違っていて覚えていると思っていることがあります。それは表1の1番左上にある( )です。この小括弧は、次のような演算の順序を決めるためのものではありません。

```
a = ( b + c ) * d
```

この小括弧は演算子ではないので、表1に記載されるはずがありません。この小括弧は、演算子の優先順位に関係なく、最優先で演算を行うことを意味しています。

それならば表1の左上の小括弧は何なのでしょう。その答えは関数呼び出しの小括弧です。文法だと関数呼び出しは演算と考えていて、その記号が( )なのです。また、この( )が使える被演算数は、関数のアドレスを意味するものに限られています。

#### ● 関数名は関数の先頭アドレスを意味する

関数の名前は、それだけで関数の先頭アドレスを意味します。これは、配列の名前がそれだけで配列の先頭アドレスを意味することと同様です。

入門書に記載されているような次の説明は、正しくありません。

関数呼び出し → 関数名(実引数)

正しい関数呼び出しの説明は次の通りです。

関数呼び出し → 関数のアドレス(実引数)

このように解釈しておかなければ、関数を指すポインタを使った関数呼び出しのコーディングが理解できなくなります。

### ■ 使い方

リスト1に示すのは、関数を指すポインタを使った関数の呼び出し例です。

abs関数はint型の絶対値を求める関数です。ここではabs関数をポインタ変数経由で呼び出します。そのポインタ変数の宣言がmain関数内に宣言されているpointです。

#### ● 宣言の特徴…返り値も引数もint型の関数しか指せない

関数を指すポインタは、小括弧を使った少し難しい宣言となります。

表1 C言語における演算子の優先順位

上から優先順位の高い順に並べている。今回からは、この中でも特に注意すべき項目や、一見理解しにくい不思議な文法などを解説する

演算子	結合規則
( ) [ ] . -> ++ --	左から右
! ~ ++ -- + - * <sup>注A</sup> & sizeof	右から左
(type)	左から右
* / %	左から右
+ -	左から右
<< >>	左から右
< <= > >=	左から右
== !=	左から右
&	左から右
^	左から右
	左から右
&&	左から右
	左から右
? :	右から左
= += -= *= /= %= &= ^=  = <<= >>=	右から左
,	左から右

注A：単項の+、-、\*は二項形式より高い優先度を持つ