

» 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8回 派生型③…ちゃんと使えばメモリ容量の削減に！ 共用体とは

鹿取 祐二

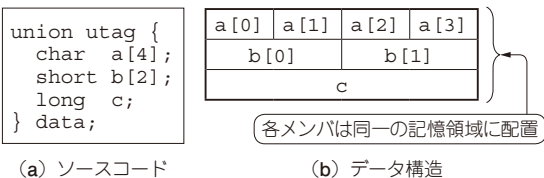


図1 共用体のデータ構造の例

各メンバが同一の記憶領域に配置されるので、使用メモリ量を削減できる。その代わりに、各メンバは同時に存在できない

C言語は、誕生から50年近く経つ今でも、分野を問わず、さまざまな場面で使われるプログラミング言語です。言語仕様（文法）は標準規格化されていますが、曖昧な部分が多く存在します。

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。

今回は前回（本誌2021年7月号）に続いて派生型、いわゆる基本型から派生した配列、ポインタ、構造体、共用体、ビットフィールドについて紹介します。各派生型が移植性を重視した組み込み系システムで利用できるものなのかを理解しましょう。（編集部）

14 共用体は適正に利用すれば メモリ量の削減に効果を発揮する

● MISRA-Cでは禁止だが…場合によっては活用すべき派生型

車載機器や産業機器などの組み込みソフトウェア開発で広く利用されているC言語のコーディング規約MISRA-Cでは、共用体に対して厳しい利用制限を設けています。そのため、共用体を使っている組み込みプログラマはほとんどいないと思います。

ただ、筆者は違う見解を持っています。共用体は、言われているほど敬遠するものではなく、むしろ積極的に活用すべきものだと考えています。本稿の読者の中には、共用体を全く知らない人もいると思うので、今回は共用体の記述方法から利用目的、禁止事項などを順番に紹介します。

● 記述方法

記述方法は、構造体と非常によく似ています。構造体はキーワードでstructを使いますが、共用体はunionを使います。それ以外の部分は構造体と同じです。もちろん、typedefを使って型名を宣言することもできます。

```
union utag {
    char a[4];
    short b[2];
    long c;
} data;
```

● 構造体との違い

構造体とは何が違うのでしょうか。名前の示す通り、各メンバが記憶領域を共用する点が違います。結果、前述の共用体データの場合は、図1のようになります。今回の例では、各メンバのサイズが同じになるように記述しましたが、特に同じでなくても構いません。サイズが異なる場合、一番大きな記憶領域が必要となるメンバのサイズで確保されます。

▶ 利用目的…使用メモリ量の削減

共用体の利用目的は、使用メモリ量の削減です。これ以外にはありません。

従って、使用メモリ量の削減を必要としないことの多いPCやワークステーションのプログラムでは、利用されることが少ないです。しかし、メモリ容量の少ない組み込み系のプログラムでは、利用を検討すべき文法であると言えます。

▶ 注意事項…各メンバは同時に存在できない

注意事項は、各メンバは同時に存在することができない点です。記憶領域を共用するので、当然といえば当然です。結果、次のように特定のメンバで値の更新を行うと、他のメンバとしての値は失われます。

```
data.a[0] = 'a';
// 配列bと変数cのメンバは失われる
data.b[1] = 1000;
// 配列aと変数cのメンバは失われる
data.c = 0x12345678;
```

第1回 整数型①…値の範囲と負の内部表現（2021年1月号）

第2回 整数型②…表現できない値を代入したときの規則（2021年2月号）

第3回 整数型③…整数の格上げと算術変換規則（2021年3月号）