≫ 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21回　静的変数領域の削減④…具体的な削減方法

鹿取 祐二

リスト1　初期状態ゼロから始めたい静的変数に0を記述してはならない
処理系によっては，0を記述すると初期値があると判断して，ROM領域とRAM領域の両方に割り付けられてしまい，無駄に静的領域を消費してしまう

```
int a;
int b = 0;
```
（bを初期値0にしたい）

（a）ソースコード

```
                .glb    _a
                .glb    _b
; 以下が変数aの結果
                .SECTION B,DATA,ALIGN=4
_a:             .blkl   1
; 以下が変数bの結果
                .SECTION R,DATA,ALIGN=4
_b:             .blkl   1
                .SECTION D,ROMDATA,ALIGN=4
                .lword  00000000H
```
（RAM領域）（ROM領域）

（b）0を記述すると初期値があると判断する処理系のコンパイル結果（RXの処理系）

```
                .PUBLIC  _a
                .PUBLIC  _b
                .SECTION .bss,BSS
_a:             .DS      (2)
_b:             .DS      (2)
```

（c）0を記述しても初期値がないと判断する処理系のコンパイル結果（RL78の処理系）

本連載では，C言語の言語仕様（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．

第18回（本誌2022年12月号）から，静的変数領域の削減について紹介しています．「静的変数」の一言で，どの変数であるかが理解できていればよいのですが，宣言場所や記憶クラスも絡んでおり，多少複雑です．前回まで，静的変数の文法的な説明をしてきました．今回は，最終目的である静的変数領域の削減方法を紹介します．（編集部）

これまで紹介した静的変数の性質，およびコンパイル結果が理解できれば，その領域の削減方法はおのずと判明します．今回は，静的変数領域の削減方法を説明します．

33 ケース①…初期値がゼロなら記述しない

●「0」を記述すると「初期値あり」とみなされる

静的変数の宣言で注意する1つ目の内容は，初期状態ゼロから始めたい変数です．初期値をゼロにしたくても，**リスト1**（a）の2行目のように記述してはいけません．

ここでは静的変数を単純な大域変数で宣言しています．比較するために初期値のない大域変数を1行目で宣言しました．これをRX（ルネサス エレクトロニクス）の処理系でコンパイルし，最終的なROM化のイメージで示すと，**リスト1**（b）のようになります．

初期値がゼロでも，0を記述してしまうと初期値があるとみなされてしまいます．結果，他の初期値のある静的変数と同様にROM領域とRAM領域の両方に割り付けられてしまいます．

初期値のない静的変数は，RAM領域だけの配置であり，スタートアップ・ルーチンがゼロ・クリアを行ってくれます．従って，初期状態ゼロから始めたい変数に，bのように初期値の0を記述する必要はありません．初期値がゼロであれば，変数aのように初期値なしで宣言するのが正解です．

これは，一度理解すれば二度と繰り返すことはないように思えますが，初期値の0をマクロ名などに置き換えていると気づかないことが多いです．マクロ名を乱発している場合，その値がゼロかどうかを確認して，コメントなどに対策を施したことを記載しておいた方がよいでしょう．

● 処理系によっては対策されている場合もある

本件に関しては，一部の処理系では対策が行われている場合があります．例えば，**リスト1**（a）のソースコードをRL78（ルネサス エレクトロニクス）の処理系でコンパイルし，最終的なROM化のイメージで示すと，**リスト1**（c）のようになります．初期値の設定があっても，それが0であれば初期値を記述していな

第9回　派生型④…移植性は無いが可読性バツグン! ビット・フィールド（2021年11月号）
第10回　演算子①…「関数呼び出し」に使う小括弧（2021年12月号）
第11回　演算子②…[ ] と * の意味，++aとa++の違い（2022年1月号）