

» 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25回 スタック・サイズの算出①…スタック領域は処理系が適当に確保しているだけ

鹿取 祐二

リスト1 デフォルトで作成されるサンプル・プログラムにおけるスタック領域の確保例

このサイズはいずれも処理系のデフォルト設定であり、どれも適当なサイズで確保されている（開発環境や処理系のバージョンによって変わる場合がある）

<pre>#pragma stacksize 0x200</pre>	<pre>#pragma stacksize 0x400</pre>
(a) H8のスタック領域…512バイト	(b) SHのスタック領域…1024バイト
<pre>.SECTION .stack_bss, BSS _stackend: .DS 0x200 _stacktop:</pre>	<pre>#pragma stacksize su=0x100 #pragma stacksize si=0x300</pre>
(c) RL78のスタック領域…512バイト	(d) RXのスタック領域…256バイト+768バイト

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を目指します。（編集部）

マイコンのプログラムを動作させるには、スタック領域が必要です。しかも、ほとんどの組み込みシステムは仮想記憶の機構を持っていないので、システム動作中にはスタック領域を増やせません。結果として、スタック領域はプログラム作成時に確保したサイズ以上に使用することが許されません。つまり、スタック領域は確保したサイズ内に収まっているのかを検証する必要があります。今回からは、その算出方法を紹介しましょう。

43 スタック領域は処理系によって適当に確保される

● スタック領域の確保例

スタック領域は、マイコンが動作するためには必ず存在しなければなりません。筆者が携わってきたマイコンの中で、スタック領域を必要としないものはありませんでした。それほどスタック領域は重要なのです。

ここでは、ルネサス エレクトロニクス社製のH8、SH、RL78、RXの4つのマイコンを例に、スタック領域の具体的な確保例を見ていきます。リスト1にデフォルトで自動生成されるサンプル・プログラムにおけるスタック領域の確保例を示します。

リスト1(a)のH8とリスト1(b)のSHは、`#pragma stacksize`の後の数値が具体的なスタック・サイズです。H8は0x200なので512バイト、SHは0x400で1024バイト確保されています。

リスト1(c)のRL78はアセンブリ言語で確保されていますが、`.DS`制御命令のあとの0x200がスタック・サイズで、512バイト確保されています。リスト1(d)のRXは、スタック領域をユーザ・モード用とスーパーバイザ・モード用の2つに分離できます。`su=`の後の数値がユーザ・モード用で、0x100の256バイト確保されています。また、`si=`の後の数値がスーパーバイザ・モード用のサイズで、0x300の768バイト確保されています。

● デフォルトの設定で必ず動作するの？

▶ いずれも適当に確保されているだけ

同じプログラム、同じマイコンであっても、開発環境によって必要となるスタック・サイズが変化します。同じメーカーの開発環境であっても、処理系のバージョンが異なるとサイズが変わる可能性があります。全システム共通のサイズなどは存在しません。

▶ サイズが十分か判断するには計算や検証が必要

リスト1の例は、どれも適当なサイズでスタック領域が確保されています。このサイズでスタック領域が収まるかどうかは、プログラマがサイズを計算し、検証しなければなりません。組み込み業界では「デバッグが完

第19回 静的変数領域の削減②…動的変数と静的変数の性質をプログラムで確認する(2023年1月号)

第20回 静的変数領域の削減③…静的変数のコンパイル結果を見定める(2023年2月号)

第21回 静的変数領域の削減④…具体的な削減方法(2023年4月号)