≫ 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26回　スタック・サイズの算出②…関数呼び出し時のスタック領域の使われ方

鹿取 祐二

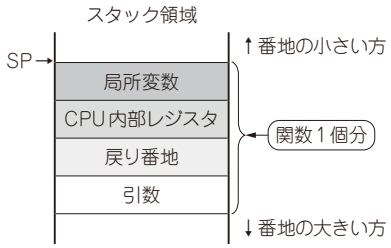

図1　スタック領域に保存されるデータの内容
関数が呼び出されたときの保存内容

リスト1　このプログラムの関数を実行したときのスタックの使われ方を見てみる

```
void main(void)
{
    関数（ 引数 ）;      // 引数の格納
}                        // 関数呼び出し

void 関数（引数の宣言）
{
局所変数の宣言           // 局所変数の確保
    文                   // レジスタの退避
    文
}
```

第25回（2023年10月号）から，スタック領域の算出方法を紹介しています．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．もし確保したサイズを超えると，プログラムが正常に動作せず，思わぬ事故を招くことがあります．そのため，事前に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．

今回はスタック領域の使われ方について解説します．（編集部）

44 スタック領域の使われ方

● 関数呼び出し時の動作を見てみる

C言語のプログラムの場合，スタック領域には**図1**に示すデータが保存されます．もちろん，使うマイコンや開発環境によって多少の違いはありますが，ほぼ同じと考えて構いません．また，保存されるタイミングは関数呼び出しが行われたときです．関数が呼び出されると，これらのデータが番地の大きい方から小さい方へと保存（または確保）されていきます．関数が終了するとき，番地の小さい方から順番に必要なデータは復旧され，必要のないデータは破棄されながら，関数読み出し前の状態に戻ります．

また，マイコンのエンディアンがビッグ/リトルのどちらであっても，不思議なことにスタック領域は番地の大きい方から小さい方に向かって使用されます．またスタック領域は，ほとんどのマイコンでSPと名の付くスタック・ポインタと呼ばれるCPU内部レジスタで管理します．

ここまで説明した内容を確認します．大ざっぱですが，**リスト1**の例でスタック領域の使われ方を解説します．関数呼び出し前のスタック領域は**図2**（**a**）の通りとします．

● ステップ①…引数を格納

関数呼び出しのとき，最初にスタック領域に格納されるのは**図2**（**b**）が示すように引数です．

C言語での引数の渡し方はコピー渡しなので，引数はスタック領域にコピーしながら格納します．**図2**（**b**）では，引数が1つしか記載されていませんが，複数個が指定された場合は複数個格納されます．

▶全ての引数が格納される訳ではない

全ての引数がスタック領域に格納されるとは限りません．近年の処理系は，コピー先としてスタック領域ではなくCPU内部の汎用レジスタを使うことが多いです．スタック領域は単なるメモリなので，汎用レジスタを利用した方が性能的に優れているという訳です．

▶格納規則は複雑で処理系により異なる

ただし，汎用レジスタには数に限りがあります．引数の型（サイズ）にも影響を受けます．結果，引数の汎用レジスタへの格納規則は非常に複雑です．同じマイコンであっても処理系が異なると規則が違うことがあります．例えば，ルネサス エレクトロニクスの4つの処理系だと，**表1**のような規則となっています．