

≫ 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

最終回

第32回 動的スタック領域の削減④…局所変数を削減する

鹿取 祐二

リスト1 int型10個の配列を局所変数として宣言するプログラム

```
void sub(int array[]);

void func(void)
{
    int array[10];

    sub( array );
}
```

50 局所変数の削減

● 汎用レジスタに割り付く程度なら気にしなくてOK

局所変数も引数と同じく、ほとんどのものは汎用レジスタに割り付きます。汎用レジスタに割り付いた局所変数は、スタック領域を使いません。問題は汎用レジスタに割り付け可能な局所変数のサイズと個数です。

引数と同じように「何個までなら大丈夫」とか「型はどのサイズまで許される」といった、全てのマイコンや処理系に共通の内容は存在しません。引数と同じように、int型で2～4個程度であれば確実に汎用レジスタに割り付くと思います。ただし、それ以上だと分かりません。引数のような明確な割り付け規則が存在する訳でもありません。

このことから、局所変数に関してはあまり気にしても仕方がないと考えてください。マイコンを問わずに処理系が頑張ってくれるので、1桁程度の個数であれ

ば問題ないと考えてよいと思います。

● 汎用レジスタに割り付かないとスタックを大量に消費する

局所変数として汎用レジスタに割り付く型は明確ではありませんが、割り付かない型は明確です。配列や構造体、共用体などの集成型は汎用レジスタに割り付きません。もちろん小さなサイズの集成型であれば汎用レジスタに割り付く可能性もありますが、それは特殊な例です。一般的に集成型の局所変数は、レジスタに割り付かないと考えて構いません。

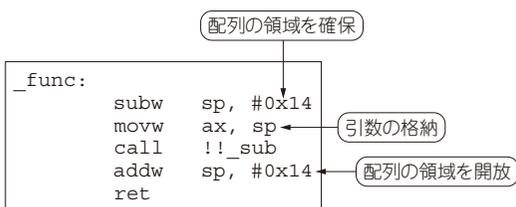
▶ 配列を局所変数として宣言すると…

具体例で確認します。int型10個の配列を局所変数として宣言したときのコンパイル結果を見てみます。

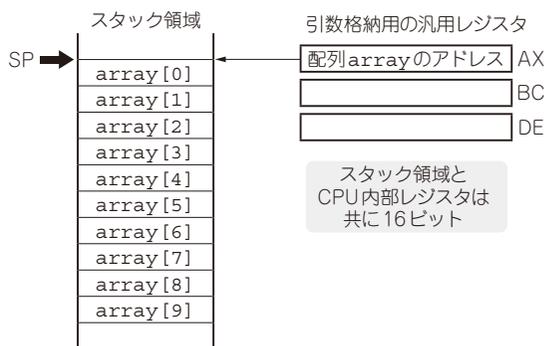
リスト1にソースコードを示します。図1にRL78、図2にRXのコンパイル結果を示します。

RL78(int型のサイズが2バイト)の例だと、関数が呼ばれたときに16進数0x14、10進数で20バイト分、配列arrayの領域をスタックに確保しています。

RX(int型のサイズが4バイト)の例だと、関数が呼ばれたときに16進数0FFFFFFD8H、符号付き10進数で40バイト分、配列arrayの領域をスタックに



(a) コンパイル結果



(b) スタックと汎用レジスタ

図1 リスト1をRL78の処理系でコンパイルした結果

配列arrayのために20バイト分のスタック領域を確保している