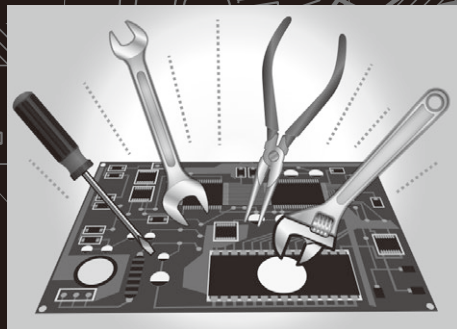
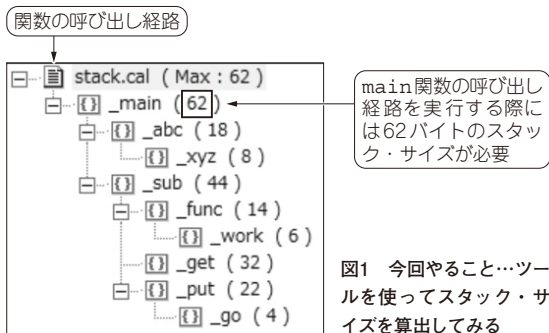


# 組み込みOS チューニング・ テクニック

最終回  
第4回 ソールによるスタック・サイズの算出



鹿取 祐二



2023年7月号特集「ゼロから作るOS」で取り上げたTry Kernelは、1500行で学習するため機能を絞り込んであります。ここで紹介するμT-Kernelは、Try Kernelの発展版と言うこともでき、多くの製品に導入されています。今回はあるマイコン向けに作り込んだOSおよび動作プログラムを、ほかのマイコンに載せ替える際に、先人がどのような工夫を施しているのかを紹介します。

今回は、組み込み向けリアルタイムOSでスタック・サイズを算出する方法を紹介します。解説にあたり、例としてトロンフォーラムと筆者が公開している組み込みシステム向けのリアルタイムOS μT-Kernel 3.0<sup>(1)</sup>のソースコードを取り上げます。(編集部)

## 無償公開されているリアルタイムOSの欠点

μT-Kernelに限らず、無償で公開されているリアルタイムOSの最大の欠点は、スタック・サイズの計算式が公開されていないことにあります。筆者は考えています。32ビットCPU向けOSのサンプルでは、タスクのスタック・サイズとして1024バイトを確保していると思います。しかしこの値はあくまでも適当なサイズです。

### ● スタック・サイズを算出することの重要性

μITRON系の場合、ほとんどはメーカーが開発し、有償で販売しています。そのため、当然のごとくス

タック・サイズの計算式が公開されています。きっとシステム・コールごとに必要となるスタック・サイズも公開されていると思います。もし、それらが公開されていなかったら、それはまがい物であると言っても過言ではありません。スタックが保証できていないシステムなど怖くて使用できませんからね。有償であれば当然のことです。

### ● 無償公開だとバージョンアップ対応の点で限界がある

しかし、無償で公開されているリアルタイムOSにそれを求めるのは酷です。使用する処理系も日々バージョンアップを繰り返すので、それらに追従するのは正直厳しいものがあり、将来的にも難しいと思います。つまり、具体的な数値としてスタック・サイズを保証できないのが無償で公開されているリアルタイムOSの欠点です。残念ながら、トロンフォーラムから公開されているμT-Kernel 3.0のソースコードも同様です。

## 解決策…スタック・サイズを自動算出するツールを使う

### ● 処理系によってはスタック見積もりツールが使える

筆者はμT-Kernel 3.0でスタック・サイズが保証できる仕組み、具体的にはスタック・サイズの計算式と、その計算を容易にしてくれるツールの提供に尽力しました。それが処理系にルネサス エレクトロニクス社のCC-RLやCC-RXを選んだ理由です。

CC-RLやCC-RXには必要なスタック・サイズを自動的に算出してくれるスタック見積もりツール(Call Walker)があります。このツールを利用すれば、コンパイラが生成した情報を基に図1のように関数呼び出し経路を表示してくれます。

呼び出し情報ビューに表示されているのが関数呼び出し経路です。必要なスタック・サイズは関数呼び出し経路によって変化するので、ソースコードから関数呼び出し経路が解析できなければなりません。また、関数名の右側に記載されている数値が、その経路を実

第1回 移植性：作ったOSの対象とするマイコンを切り替える(2023年8月号)  
 第2回 省メモリ設計への対応(2023年9月号)  
 第3回 割り込み応答速度の向上(2023年10月号)