

プロローグ マルチコア OSを理解するためには…作ってしまおう!

# ゼロから作るマルチコアOS 「Try Kernel」の概要

豊山 祐一

プロジェクトのルート・ディレクトリ		行数
application		
boot		
boot2.c	ブート第2ステージ	44
reset_hdr.c	リセット・ハンドラ	241
vector_tbl.c	例外ベクタ・テーブル	63
device	デバイス・ドライバ	
include		
apidef.h	API定義	114
config.h	コンフィギュレーション	
error.h	定義	15
gpio.h	エラー・コード定義	30
knldef.h	GPIO定義	11
sysdef.h	カーネル内部関数定義	165
syslib.h	ハードウェア定義	296
trykernel.h	システム・ライブラリ	
typedef.h	定義	70
trykernel.h	アプリケーション用	
trykernel.h	ルート定義	6
trykernel.h	データ型定義	54
kernel		
context.c	タスクの	
context.c	コンテキスト管理	32
dispatch.S	ディスパッチャ	118
eventflag.c	イベント・フラグ	147
gpio.c	GPIO制御	44
icc_core.c	CPUコア間制御	47
icc_int.c	CPUコア間割り込み	74
icc_spin.c	CPUコア間	
icc_spin.c	スピンロック	35
inittsk.c	初期タスク	55
interrupt.c	割り込み管理	116
messagebuf.c	メッセージ・バッファ	190
scheduler.c	スケジューラ	69
semaphore.c	セマフォ	110
syslib.c	システム・ライブラリ	30
systemer.c	システム・タイマ	39
task_manage.c	タスク管理	83
task_queue.c	タスクの待ち行列操作	51
task_sync.c	タスク付属同期	86
linker		
pico_memmap.ld	リンカ・スクリプト	70

図1 今回作成するOSのファイルと行数の一例(マルチコア対応 Try Kernel-S)

## マルチコア OSを作る環境が整ってきた

マルチコアの技術は、今やPCやサーバでは当たり前となりました。スマートフォンやタブレットでもマルチコアのプロセッサが普通に使われています。そして、マイコンの世界でもマルチコアが広まってきています。

本特集で取り上げるラズベリー・パイ Pico はとても安価なマイコン・ボードですが、搭載されているのは Arm Cortex-M0+ コアを2個搭載したマルチコア・マイコンです。また、2024年8月に発表されたばかりのラズベリー・パイ Pico 2 は Arm Cortex-M33 コアを2個搭載し、さらに RISC-V コアも搭載しています。このように気軽にマルチコア・マイコンを使える環境が整ってきました。

本誌2023年7月号の特集「ゼロから作るOS」では、ソースコード1500行の小さなOS「Try Kernel」を作り、OSの仕組みやマイコンの低レイヤの制御を解説しました。Try KernelはシングルコアのOSでしたが、今回はこれを拡張しマルチコア対応のOSを作成します。

## マイコン OSの自作の勧め…作って理解しよう

OSの仕組みを理解するには実際に自分でOSを作ってみるのが確実です。またOSのような低レイヤ・プログラミングはコンピュータの仕組みを理解するうえでも役に立ちます。マルチコアのマイコンはどのように制御するのか、そしてどのように使うのかなどは、OSを自作することにより、理解が深まるでしょう。

これがWindowsやLinuxのようなPCのOSとなると、ソフトウェアの規模がとて大きく、作成は大変な作業になります。しかし、マイコンのOSは非常にコンパクトであり、自作は十分に可能です。