

# 実践的 PowerPC 活用テクニック



## 第12回 デバッグ用に独自 OS を実装する

坂井 弘亮

CQ 出版の組み込みシステム開発評価キット (BLANCA) のオプション CPU カードとして、PowerPC のオプション・カードが販売されています。本連載では数回にわたって、その使い方について説明する予定です。

今回は PowerPC そのものの解説から少し離れ、次回の GDB 対応の準備として、簡単な OS を動作させてみます。(筆者)

### 1. 今回解説する独自 OS —— KOZOS

GDB でデバッグできるようにするには、GDB スタブというプログラムを移植する必要があります。GDB スタブはターゲット・マシンで動作し、パソコン側の GDB からコマンドを受け取ってターゲット・マシンを制御する簡単なプログラムです。これは割り込みの延長で動作するため、必ずしも OS が必要というわけではありません。

スタブを移植する際は、スタブ側が割り込みハンドラを自前で用意し、動作時に差し替える処理が必要です。しかし OS を動作させ、その割り込みハンドラからスタブを呼び出してやると自前のハンドラは不要になり、アセンブリ・コードもほとんど不要になるためスタブの移植の敷居はずっと低くなります<sup>注1</sup>。

またデバッガの動作を見るためには、やはり OS くらいは動かないといけないという思いもあります。今回は GDB 対応への準備段階として、筆者の作成している独自組み込み向け OS「KOZOS」を移植してみます。KOZOS については下記の URL を参照してください。

<http://www.saturn.dti.ne.jp/~hsakai/kozos/index.html>

KOZOS はまだまだ開発途中ですが、さまざまな組み込み機器上で独自 OS 作成用の簡単なサンプルとなるものを目指しています。このため内部構造を必要最低限の機能に絞って極力シンプルに作っており、今回のように「ちょっと OS を動かしたい」という用途には最適です。

注1：ただしいくつかの制限や注意すべき点がある。これらについては次回に説明する。

### 2. OS の移植とソース・コードの解説

今回は誌面掲載用として、必要最低限の部分のみを抽出して移植を行いました。OS の機能としてスレッド管理やメモリ管理、割り込み処理を持っています。

まず、KOZOS のソース・コードについて解説します。KOZOS のソース・コードとして表1に示すファイルを追加しました。

それぞれのファイルについて説明します。

#### ● configure.h, kozos.h の追加

では、スタック関連や OS 内部で利用するシグナル番号を定義している configure.h、外部向けに各種関数などのプロトタイプ宣言を行っている kozos.h (リスト1) を追加しましょう。KOZOS を利用する場合は、kozos.h をインクルードするとシステム・コールの呼び出しが可能になります。

#### ● thread.h, thread.c の追加

スレッドの処理用に thread.h と thread.c を追加します (リスト2, リスト3)。これが OS の中核部分になります。

thread.h では、kz\_msgbuf, kz\_context, kz\_thread の3種類の構造体を定義しています。これらはそ

表1 KOZOS ソース・コード一覧

ファイル名	役割
configure.h, kozos.h	各種定義ファイル
thread.h, thread.c	スレッド操作関連 (OS コア)
memory.h, memory.c	メモリ管理
syscall.h, syscall.c	システム・コール処理
command.c, extintr.c, idle.c	各種スレッド