

TOPPERS/ASP for ColdFire の実装

付属 ColdFire マイコン基板への μ ITRON 4.0 準拠リアルタイム OS TOPPERS/JSP の移植については、すでに本誌 10月号で解説した。JSP の発展型である ASP を付属 ColdFire マイコン基板で動作させるためには、ASP で定義された TOPPERS 標準割り込み処理モデルへの準拠が必要になる。ここでは、TOPPERS/ASP の実装について、付属マイコン基板で使われた ColdFire MCF5223x を題材に解説を行う。

(編集部)

横田 敬久

本誌 2008 年 10月号では、9月号に付属した ColdFire マイコン基板へ TOPPERS/JSP を移植しました。このマイコン基板は ColdFire MCF52233 を搭載しています。

TOPPERS プロジェクトでは、TOPPERS/JSP の発展形として、2008 年度の ESEC に合わせて TOPPERS 新世代カーネル ASP の公開しました。そこで本章では、ColdFire MCF5223x 系の CPU で TOPPERS/ASP を使う方法について解説します(写真1)。

1. μ ITRON から脱却した新世代カーネル ASP

● TOPPERS/JSP から ASP への変更点

TOPPERS プロジェクトが新世代カーネルとして採用し

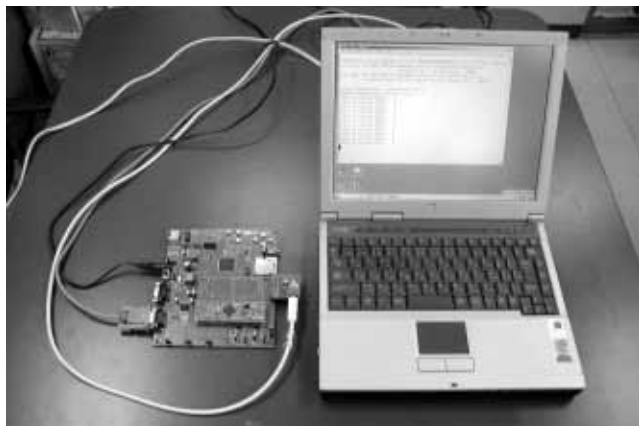


写真1 付属 ColdFire マイコン基板 + 拡張ベースボードで TOPPERS/ASP が動作している様子

BDM デバッガとして本誌 p.144 で紹介されている KEL_TBLCF を使用している。

た ASP には、JSP と比較してさまざまな改良が加えられています。

改良点は以下のとおりです。

- TOPPERS 標準割り込み処理モデルの採用
- イベント・フラグの複数タスク待ち
- データ型に C99 の型名を採用
- アラーム・ハンドラの採用
- 割り込みサービス・ルーチンの採用
- 優先度データ・キューの採用
- 割り込み要求ラインの属性の設定
- オブジェクト状態参照機能
- コンフィグレーション・パスの見直し
- そのほか、サービス・コールの改良、見直し

このため、ASP カーネルは μ ITRON のサービス・コール名を意識した構成になっていますが、互換性があるとはいえない構成になっています。あくまで JSP カーネルの成果をもとに、拡張/改良を加えた独自の Operating System (OS) というべきでしょう。

TOPPERS/JSP は、 μ ITRON 4.0 の標準(スタンダード)プロファイルを元に実装したオープン・ソースのリアルタイム OS です。それに対して TOPPERS/ASP は、 μ ITRON の仕様に改良を加えています。特に割り込み制御に対して、「TOPPERS 標準割り込みモデル」を採用し、従来のゆるい標準化から脱却し大幅に強化を図っていることが最大の特徴です。

追加されたサービス・コール(表1)を見ると、割り込み周りについては μ ITRON と比較しても充実していることがうかがえます。

表1 追加されたサービス・コール

内容	サービス・コール
割り込み要求ラインの属性の設定	CFG_INT(INTNO intno, { ATR intatr, PRI intpri })
割り込みサービス・ルーチンの追加	ATT_ISR({ ATR isratr, intptr_t exinf, INTNO intno, ISR isr, PRI isrpri })
割り込みの禁止	ER ercd = dis_int(INTNO intno)
割り込みの許可	ER ercd = ena_int(INTNO intno)
割り込み要求ラインの参照	ER ercd = ref_int(INTNO intiId, T_RINT *pk_rint)
割り込み優先度マスクの変更	ER ercd = chg_ipm(PRI intpri)
割り込み優先度マスクの参照	ER ercd = get_ipm(PRI *p_intpri)

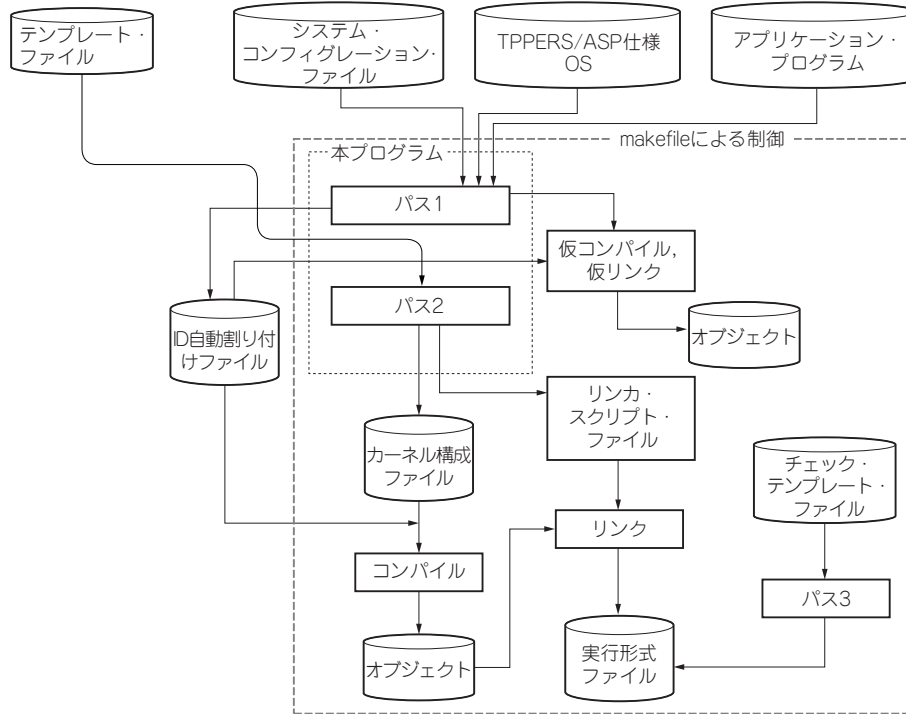


図1
TOPPERS/ASPの
コンフィギュレーション
手順

● TOPPERS/ASPの構造とコンフィグレータ

ASPカーネルでは、独自の高性能なコンフィグレータとこのコンフィグレータにより、カーネル・オブジェクトや割り込みベクタ・テーブル、割り込み処理エントリなどを自動生成できます。アプリケーションのmake時に3段階のパスを経て実行ファイルを生成します(図1)。

2. ASPカーネルのディレクトリ構造とコンフィグレータ構成

● ASPのディレクトリ構造

ASPカーネルのディレクトリ構造は、JSPカーネルと同じようにCPU依存部と非依存部に分かれています。さらに依存部の再利用性を考慮して、プロセッサ、チップ、開

発環境に依存する部分を切り分けるポリシーになっています。プロセッサ、チップ、開発環境依存部はarchディレクトリ配下に置かれます。命名規則は、

(プロセッサ名, チップ名)_(開発環境名)

とします。例えば、ColdFireのV2コアでGNUを環境とした場合は、arch/cfv2_gccとなります。ターゲット依存部ディレクトリはtarget配下に置かれ、

(ターゲット名)_(開発環境名)

と表せるので、M52235EVBボードを使用した環境ではtarget/m52235evb_gccとなります。

このようなディレクトリ構造により、ターゲット部を経由して各種依存部を参照する形になります。そのため、ターゲット依存部以外へのパスの参照は、トップ・ディレクトリを経由したパス名になります。