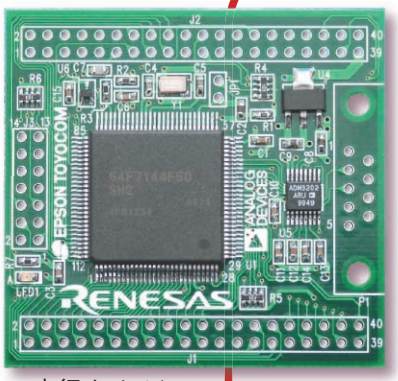
第2章

サンプル・プログラムの中身はこうなっている!

μITRON4.0仕様書を片手にsample1を読む

邑中 雅樹



第1章では，TOPPERS/JSPで動作するサンプル・プログラムsample1をSH-2付録基板上で実行させるところまでを行った．本章では，このsample1がどのようなプログラムになっているのかを解説する．
sample1は短いプログラムながら，μITRON4.0仕様のエッセンスを濃縮した内容の濃いプログラムとなっている．本章を読みこなせば，μITRONの要点を効率的に習得できるだろう．（編集部）

第1章では，はじめの一歩ということで，サンプル・プログラムのビルドとフラッシュ・メモリへの書き込み，実行までを行いました．

本章では，ソース・コードを追いながら，TOPPERS/JSPがどのようにサンプル・プログラムを処理/実行しているのかについて説明します．

μITRON4.0仕様書を片手に…

TOPPERS/JSPは，仕様名でいうと「μITRON4.0仕様スタンダード・プロファイル」を忠実に実装したものです．TOPPERS/JSP自身にも添付の解説文書が存在しますが，カーネルの作成者からの視点で書かれている点は正直否めません．ある程度理解が進むと使える資料なのですが，最初の文書としてはやや障壁が高いかもしれません．

そこで，読者の方々には，トロン協会が発行している「μITRON4.0仕様書」を参考書として手元に置いて学習されることを勧めます．μITRON4.0仕様書は，最新のVer4.02のPDF形式で，

http://www.ertl.jp/ITRON/SPEC/mitron4-j.html

から無償でダウンロード可能です．μITRON4.0仕様書も図版が少なく，入門書としては難解です．しかし，もっとも信頼できる，TOPPERS/JSP添付の文書よりはわかりやすいものとなっています．

標準サンプル・プログラムsample1

● カーネルといっしょに配布されているsample1

第1章でビルド/ダウンロード/実行したsample1は，TOPPERS/JSPカーネルのアーカイブにも収録されているサンプル・プログラムです．標準的に配布されていることから，TOPPERS/JSPを使ったアプリケーション開発者がかならず最初に実行するといっても過言ではないでしょう．

sample1は，基本的にターゲット非依存な構成になっています．同時に，TOPPERS/JSPカーネルを各種ターゲット・ボードに移植する際の簡単なテスト・プログラムも兼ねており，TOPPERS/JSPカーネルでアプリケーションを作成するうえで肝となる部分が適度な長さの中に凝縮されています．

● どうやってsample1を理解するか

sample1は短いソース・コードの中にいろいろな要素が含まれています．そこで，「タスク」，「周期ハンドラ」，「割り込み」の三つの観点からsample1を解説していきます．

タスクからsample1を追う

● sample1も複数のタスクから構成されている

まずは，タスクという切り口から入ってみましょう．オペレーティング・システム（OS）の話題でよく使われる「マルチタスク」ということばのとおり，TOPPERS/JSPカーネルを用いたアプリケーションの多くは，複数のタスクで構成され，実行できます．

● main文はどこにあるのか

どのようなプログラムでも，処理の開始から順に追って行くのが定石です．処理の頭を探すことにしましょう．

その前に，C言語でのプログラミングの基礎の基礎である，Hello Worldを思い出してみましょう．典型的な例は，以下のようなものです．

```
int
main(int argc, char **argv)
{
    printf("Hello World.");
}
```

mainも関数ですが，printfなどほかの関数とは違い，だれかが呼んでくれることになっていました．C言語で書かれた

KEYWORD — μITRON4.0仕様，TOPPERS/JSP，タスク，周期ハンドラ，割り込み

プログラムを読むには，関数 main を探すのが手っ取り早そうです．

さて，sample1.c に戻って，改めて見てみましょう．sample1.c にどのような関数が含まれているのかは，PizzaFactory3 を使うと簡単に一覧できます．sample1.c の横にある［＋］記号をクリックして，開いてみてください．**図1**のように，ファイルに含まれるインクルード・ファイル/グローバル変数/関数を得ることができます．緑色の丸印がついているのが関数です．sample1.c に存在するのは，**表1**に示した五つ（そのうち，PizzaFactory3 に表示されるのは，cpuexc_handler を除いた四つ）だけです．

main_task という，なんとなく main に近い名前の関数はあります．「並列実行されるタスク」という関数コメントがついた task という関数もなんとなく関数 main のような雰囲気はあります．しかし，そのものズバリの関数 main はありません．これはいったいどういうことでしょうか．

実は，C 言語においては関数 main は必須ではなく，好きな関数をプログラムの開始点（エントリ・ポイント）として設定できます[注1]．μITRON4.0 仕様 OS には，関数 main はありません[注2]．

TOPPERS/JSP は OS の一種ですが，現実的にはフリー・スタンディング環境の一種とみなせます．

● TOPPERS アプリケーションのエントリ・ポイント

それでは，エントリ・ポイントとなる関数は，どこで設定するのでしょうか．その答えは，sample1.cfg というファイルにあります．sample1.cfg のようなファイルをコンフィギュレーション・ファイルと呼びます．さっそく sample.cfg を覗いてみましょう．**リスト1**に sample1.cfg の一部を抜粋しました．

先ほど話題に出た，task や main_task が引き数として入っています．さっそくμITRON4.0 仕様書を開いて CRE_TSK を探してみましょう．確かに「タスクの生成」という説明で，CRE_TSK という項目があります（**図2**）．静的 API という耳慣

表1　sample1.c に存在する関数

関数名	関数コメントの内容
task	並行実行されるタスク
tex_routine	並行して実行されるタスク用のタスク例外処理ルーチン
cpuexc_handler	CPU 例外ハンドラ（マクロ CPUEXC1 が定されている時のみ有効）
main_task	メイン・タスク
cyclic_handler	周期ハンドラ

コラム1　TOPPERS API 仕様書

本章では，μITRON4.0 仕様書を API のユーザ・マニュアルのように利用しています．最近まで，TOPPERS 系の ITRON 仕様 OS は，μITRON4.0 仕様に忠実な仕様書であろうとしていたため，独自の API 仕様書をもっていませんでした．しかし，μITRON4.0 仕様書は最大公約数的な記述しかなく，TOPPERS/JSP や TOPPERS/FI4 カーネルではどのような実装になっているのかという情報が決定的に不足しています．また，割り込み標準モデルや ASP/HRP カーネルの登場など TOPPERS プロジェクト独自に機能定義を行うようになってきてから，μITRON4.0 仕様との差分量が大きくなり，このままではユーザの利便性が落ちる懸念が出てきました．

そこで，NPO 法人 TOPPERS プロジェクトでは，今年からある程度の予算を割いて，TOPPERS プロジェクト独自の仕様書を作成する計画を立てています．この API 仕様書が完成すれば，μITRON4.0 仕様書を参照しなくても TOPPERS/JSP カーネルのアプリケーションを作成できるようになると期待されています．

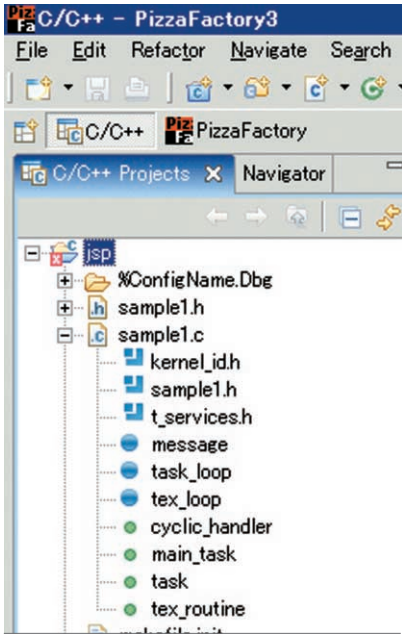


図1
関数の一覧表示

注1：より厳密には，C 言語で作成したアプリケーションの実行環境にはホスト環境（OS 上でアプリケーションが動作する環境）とフリー・スタンディング環境（OS なしで直接アプリケーションが動作する環境）という二つの環境があり，フリー・スタンディング環境では関数 main がエントリ・ポイントである必要がない，ということになる．

注2：TOPPERS 系 ITRON 仕様 OS では，関数 main はまったく存在しないが，μITRON4.0 仕様 OS の実装によっては，カーネルそのものの開始を関数 main で始めているものもある．