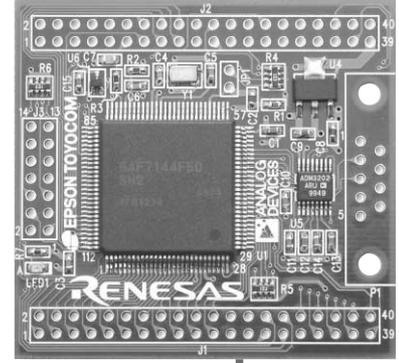


車載向けOS OSEKを動作させる

SH-2付録基板でのTOPPERS/OSEKカーネルの実行

森川 聡久/片岡 歩/鵜飼 敬幸/杉本 明加/橋本 昌伸/大西 秀一



本章では、車載向け用途OSであるTOPPERS/OSEKカーネルについて解説する。TOPPERS/OSEKカーネルはOSEK/VDX OS仕様に準拠したOSで、カーネル本体以外にも、コンフィグレーション・ファイルOILと、それを解釈するシステム・コンフィグレータも重要になる。そこで本章では、カーネル本体の仕様とシステム・サービスについて解説したあと、コンフィグレーション・ファイルOILとシステム・コンフィグレータについても解説する。(編集部)

本章と第5章では、TOPPERSプロジェクトより公開されているTOPPERS/OSEKカーネルをSH-2付録基板に移植します。OSEK/VDXは車載向け用途を念頭に仕様策定されました。コンパクトな仕様と高い信頼性が特徴です。そのため、今回の付録基板のような限られた環境下でも動作が可能で、学習用途にも最適です。TOPPERS/OSEKカーネルはOSEK仕様に準拠したフリーなソフトウェアです。

本章ではまずOSEKの概要について紹介したあと、すぐに試せるサンプル・プログラムの実行方法、自分でサンプル・プログラムをビルドする方法、ビルドに関連するファイルについて解説します。

TOPPERS/OSEKカーネルとは

本誌ではTOPPERS/OSEKカーネル(以下、本カーネル)を本格的に扱うのは初めてなので、本カーネルについて簡単に説明します。

本カーネルはOSEK/VDX(Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug/ Vehicle Distributed eXecutive)により仕様策定された、車両制御機器用RTOS仕様(OSEK/VDX OS仕様)^{注1}に準拠したリアルタイム・カーネルです。OSEK/VDX OS仕様はオープン仕様です

表1 OSEK/VDX仕様が提供する機能一覧

機能	概要
タスク管理	タスクの起動と終了 タスクの状態管理とタスクの切替え処理
同期機構	リソース管理・デバイスなどの共有オブジェクトの排他制御・イベント制御・タスクの同期
割り込み管理	割り込みサービス・ルーチン(ISR)
アラーム	相対/絶対アラーム
エラー処理	さまざまなエラーに対応する支援機構

が、オープンソース・ソフトウェアではありません。日本でいうところのμITRONのようにオープン仕様ではあるが、実装は各OSベンダにまかせるといった立場のOSに近いと思います。

(株)ヴィッツでは名古屋大学高田研究室との共同研究にて、OSEK/VDX OS仕様準拠RTOSの実装を行い、成果物をTOPPERSプロジェクトにコントリビュート(貢献)し、このRTOSをTOPPERS/OSEKカーネルとして公開しています。

本カーネルも車両制御用ECUに搭載されることを前提として開発されており、TOPPERS/OSEKカーネルの特徴として、高速な応答性能、省メモリ実装、高信頼性(MODISTARC適合、MISRA-C検証)を実現しています。

自動車制御機器用ECUに特化したOS OSEK/VDX

OSEK/VDX OS仕様の特徴を列挙します。表1にOSEK/VDX OS仕様で提供する機能の一覧を挙げました。マルチタスク環境を提供しながらも、自動車制御機器用ECUに必要な最低限度の機能を提供していることがわかります。とくにタスクに関連する機能に注目すると、システムの中で同期を取るための機構が非常に少ないことに気がきます。

● タスクは二つの属性と三つの状態をもつ

OSEK/VDX OS仕様では、タスクの属性として「基本タスク」と、「拡張タスク」の2種類が定義されています。

基本タスクは休止状態、実行可能状態、実行状態の3状態しかもたず、待ち状態が存在しません。拡張タスクは、基本タスクの3状態に加えて、待ち状態を加えた四つの状態をもちます(図1)。

注1:ここでいう「車両制御」は、エンジン制御やトランスミッション制御などを想定しており、高速な処理、低コスト(省メモリ)、高い信頼性が求められる。

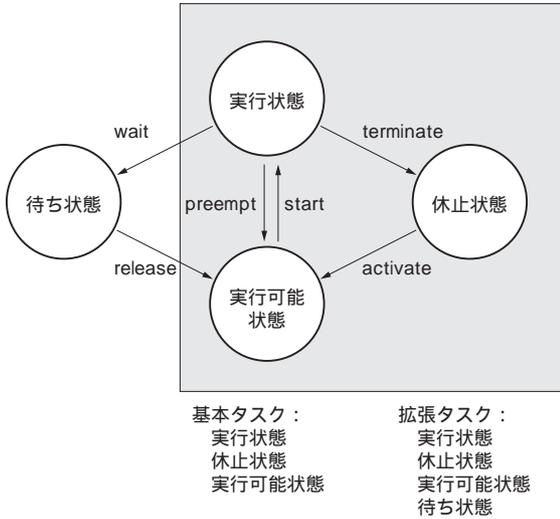


図 1 タスクの状態

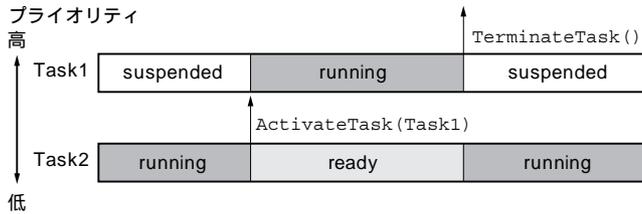


図 2 タスクの起動と終了時のタスク状態の変化

● **フック機能**——不具合の発見やシステムの性能評価を行う
エラー処理の中で特筆すべき点はフック機能です。フック機能はカーネル動作中に一時的にユーザのルーチンを実行する機構で、カーネルの開始/終了時、タスクの切り替わり前後、エラー発生時にそれぞれフック関数を用意できます。ユーザはこれらのフック関数を利用することで不具合の発見や、システムの性能評価を行うことができます。

● **カーネルが提供するシステム・サービス**
ここまで OSEK/VDX OS 仕様が提供する機能について説明しました。次にカーネルが提供するシステム・サービス(ほかの OS ではシステム・コールやサービス・コールと呼ぶもの)を表 2 にまとめました。システム・サービスの一覧を参照すれば、カーネルが提供する機能はどのシステム・サービスに対応するかがわかるといえます。

システム・サービスの詳細については、パッケージ内のカーネル・ソース・ツリーの doc ディレクトリにある TOPPERS/OSEK カーネル外部仕様書にて詳細に解説してあります。

なお、SignalCounter は本カーネル独自のシステム・サービスです。OSEK/VDX OS 仕様ではカウンタと呼ばれる、アラームの Tick 管理をしているオブジェクトへのカウント供給の方法については、実装依存としています。そのため本カーネルでは SignalCounter を独自に追加しています。

表 2 システム・サービス一覧

OSEK/VDX OS 仕様システム・サービス一覧	
システム・サービス名	説明
ActivateTask	指定タスクを起動する
TerminateTask	自タスクを終了する
ChainTask	自タスクを終了し、指定タスクを起動する
Schedule	スケジューリングを明示的に行う
GetTaskID	現在 Running 状態にあるタスク ID を取得する
GetTaskState	指定タスクの現在の状態を取得する
DisableAllInterrupts	すべての割り込みを禁止する
EnableAllInterrupts	すべての割り込みを許可する
SuspendAllInterrupts	すべての割り込みを禁止する(ネスト可能)
ResumeAllInterrupts	SuspendAllInterrupts の状態を解除する
SuspendOSInterrupts	ISR カテゴリ 2 割り込みを禁止する(ネスト可能)
ResumeOSInterrupts	SuspendOSInterrupts の状態を解除する
GetResource	リソースを確保する
ReleaseResource	リソースを開放する
SetEvent	指定タスクのイベントを設定する
ClearEvent	自タスクのイベントをクリアする
GetEvent	指定タスクのイベントマスクを取得する
WaitEvent	自タスクをイベント待ちとする
GetAlarmBase	指定アラームの基本 Tick を取得する
GetAlarm	指定アラームの現在の Tick を取得する
SetRelAlarm	指定アラームに相対アラームを設定する
SetAbsAlarm	指定アラームに絶対アラームを設定する
CancelAlarm	指定アラームの動作を禁止する
GetActiveApplicationMode	現在のアプリケーション・モードを取得する
StartOS	OS を開始する
ShutdownOS	OS を開始する終了する
ErrorHook	システム・サービス内でのエラー発生時フック
PreTaskHook	タスク切り替わり直後のフック
PostTaskHook	タスク切り替わり直前のフック
StartupHook	OS スタートアップ時のフック
ShutdownHook	OS シャットダウン時のフック
TOPPERS/OSEK カーネル拡張システム・サービス	
SignalCounter	カウンタに Tick を供給する

カーネルが提供する機能とシステム・サービスの関係

● **タスク管理とスケジューリング**

OSEK/VDX OS 仕様の処理単位の一つであるタスクは ActivateTask により起動します(図 2)。タスクの終了時には TerminateTask により自タスクを終了させます。ここで注意が必要なのが、TerminateTask は自タスクの終了しか行えないという点です。他タスクが勝手にタスクを終了させることはできないのです。この点はプログラミングにおいて注意すべきところです。ChainTask は自タスクを終了しつつ、指定したタスクを起動させるためのシステム・サービスです。シーケン