

ITRON仕様リアルタイムOSシミュレータの製作・その2

4L-9

松浦 康二 岸 信之 石川 知雄

武蔵工業大学

1 はじめに

リアルタイムOSの応用分野は従来は比較的大規模なシステムに限られてきたが、最近ではシングルチップマイコン等にも利用されるようになってきている。このような背景で基礎技術としてリアルタイムOSの利用技術を持つ必要があると考え、リアルタイムOSの教育用として学生に対してITRON-OSを個人が利用できる機会を増やし、TRONアプリケーションの開発環境の提供を目標とした[1]。そしてMS-DOS上におけるITRON-OSシミュレータ IDES (Itron application Develop Environment Simulator)を開発した[2]。今回、このシミュレータを用いITRONアプリケーションを開発し実行を行うシステム ROCESS (Realtime Operation System Education Support System)を作成したので報告する。

2 シミュレータの概要

2.1 シミュレータ機能

IDESはITRONカーネルとして μ ITRON仕様を採用している。タスクはDORMANT, RUN, READY, WAITの4つの状態で管理している。またタスクの生成はシステム起動時に静的に行う。タスク間の同期・通信にはイベントフラグ、セマフォ、メールボックスを使用する。以下にIDESのシステムコールを示す。

- タスク管理
sta_tsk, ext_tsk, tsk_sts
- タスク付属同期
slp_tsk, wai_tsk, wup_tsk
- 同期・通信
set_flg, wai_flg, flg_sts
snd_msg, wai_msg, sem_sts
snd_msg, rcv_msg, mbx_sts

Making of ITRON Real-Time OS Simulator
Kouzi MATSUURA, Nobuyuki KISHI
Tomo ISHIKAWA
Musashi Institute of Technology

- 割込み管理
def_int, ent_int, ret_int

2.2 デバッグ機能

以下にIDESのデバッグ機能について示す。

- タスク制御をサポートするコマンドを持つ
- 各資源を制御するコマンドを持つ
- システムコールヒストリー機能
- タスク状態遷移の履歴参照
- 資源獲得状況の確認
- 資源変化の履歴参照
- コマンドレベルからのシステムコールの発行
- 上記の各経過状態をリアルタイムに表示

3 ハードウェアの構成

ROCESSはPC-9801, IDES, 実験ボード及びそれらをつなぐパラレルI/Oボードより構成される。

図2-1にシステムの概略図を、図2-2に実験ボードの概略図をしめす。

I/OボードにはPPI(Programmable Peripheral Interface) 8255を使用する。I/OボードはCPUと外部機器を接続して入出力信号を中継する。プログラムはメモリーにデータを書き込んだり、メモリーのデータを読むのと同じ方法でデータの入出力が行える。用いられるI/Oボードでは8255を2個使用し16ビットでのアクセスを行う。ユーザーはIDESのユーザータスクに8255のCWR(Control Word Register)の制御をプログラムする事により実験ボードを利用する。

実験ボードは8×4個のLED, 16bitの入出力用トグルスイッチ, 割込み用プッシュスイッチより構成される。入力スイッチ列は8255の入出力ポートAに、LEDは1列16個がそれぞれ入出力ポートB, Cに接続される。入力スイッチはI/Oボードによりその状態を読み込むことができ、各スイッチ状態に応じたタスクを

作成する事により制御を行うことができる。入力用のトグルスイッチおよび割込み用のプッシュスイッチのチャタリングはチャタレス回路をハードウェアでもうけ、その影響を除去している [3]。

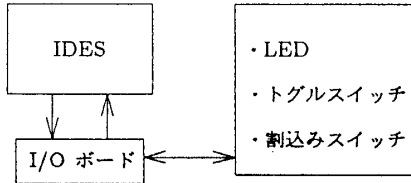


図 2-1 システム概略図

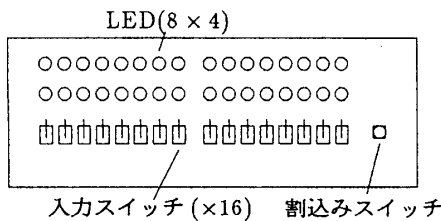


図 2-2 実験ボード概略図

4 割込み機能

割込み機能は実験ボードの割込みスイッチの入力により動作する。割込み発生時には OS の介入なしに割込み処理ハンドラが直接起動される。さらに割込みハンドラ内でディスパッチを発生させるシステムコールを発行すると、その時点ではタスクの切り替えは起こらず、遅延ディスパッチにより割込みハンドラ終了後にタスクの切り替えはおこなわれる。

タスクの切り替えは各タスクに対応した LED の動作等により確認できる。

5 アプリケーションの開発

以下にアプリケーションの開発手順を示す。

5.1 ユーザタスクの記述

目的に合わせて C 言語でユーザタスクを記述する。ユーザタスクの前処理として、ユーザタスク内にタスク初期化ルーチンを組み込む必要がある。タスク初期化ルーチンでは割込みハンドラの定義、ユーザタスクの生成を行う。ユーザタスク本体は 2 章 1 にある ITRON システムコールを用いて記述する。

5.2 実行ファイル生成

実行ファイルの生成には Turbo-C, Turbo Assembler, Turbo Linker が必要である。必要なモジュールを用意し、MAKEFILE により生成される。

5.3 シミュレータの実行

2 章 2 で示したタスク・オブジェクト操作/参照コマンドやデバッグコマンドおよび実験ボードの入力スイッチ、割込みスイッチを入力し、タスク群を走らせ、その結果を実験ボードの LED または CRT 上で確認する。

6 アプリケーション例

このシステムを用いて作成されたアプリケーション例を以下に示す。

- 割込みをトリガとし、スイッチ内容に応じたタスクを分岐させる簡単なもの
- 入力スイッチ、LED、割込みスイッチを利用したゲーム
- LED をエレベータに例え、その動作をシミュレートするエレベータ管理シミュレータ

7 おわりに

リアルタイム OS シミュレータを使用したアプリケーションの開発、実行を行える環境を作成し、実際にアプリケーション例の作成を行った。

今後はシステムコールの増加等によるさらなる高機能化、ユーザーがより簡単にアプリケーションの作成が行えるための環境への改善を目指していきたいと考えている。

参考文献

- [1] 坂村 健. ITRON 標準ガイドブック'92-'93, パーソナルメディア
- [2] 岸 信之, 石川 知雄. 第 47 回 情報処理学会全国大会論文集 6B-9, ITRON 仕様リアルタイム OS シミュレータの製作・その 1, 平成 5 年 10 月
- [3] 横山 直隆. パソコン機械制御と製作実習入門, 技術評論社