

6809 におけるリアルタイムオペレーティング システムの開発 その2

1Q-3

坪井 香奈子 林 秀房 石川 知雄

武蔵工業大学

1 はじめに

近年、マイクロコンピュータはその目覚ましい発達により、様々な機器に組み込まれ使用されるようになった。マイコン応用製品は複雑かつ高度になり、プログラムの生産性の向上のため、リアルタイム OS の利用が増加している。これに伴い、この分野に携わる技術者に基礎技術としてリアルタイム OS の利用技術の教育が必要であることは明らかである。現在、本学ではリアルタイム OS の教育として、講義が行われているが、これを実践することにより、学生の更なる理解が期待できる。

本研究では、学生実験に用いることを目的とした、6809 用 μ ITRON 仕様のリアルタイム OS ~ MITOS (Micro Industrial Operating System) ~ の開発を行った。その仕様、評価、又学生実験への応用について報告する。

2 6809 システムの構成

6809 システムの構成を図 1 に示す。MITOS の開発は NEC の PC-9801 上のクロスコンパイラ / アセンブラ (2500A.D. 社製) で行い、RS-232C ドライバを用い、6809 マイコン (テスコ社製) にダウンロードする。6809 マイコンには簡単なモニタが実装されており、コンソールとして、ポケットコンピュータが付随している。又 I/O ボード上には、ステッピングモータ、センサー、プッシュボタンスイッチ、トグルスイッチ、LED が実装されており、実機のシミュレーションに使用される。

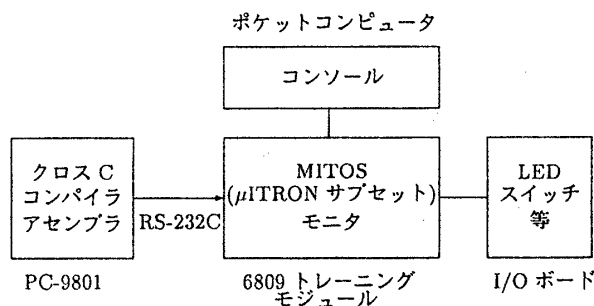


図 1 6809 マイコンシステムの構成

3 MITOS の仕様

このシステムは、 μ ITRON の機能の中から必要なものをインプリメントしたもので、 μ ITRON のサブセットとなっている。 μ ITRON

を選んだ理由は、対象が 6809 であること、機能が豊富なこと、教育を重視している点等による。プログラムは、主要部分を C 言語で、一部をアセンブリ言語で記述した。機種依存部と非依存部が分離しているため、移植性に優れている。

3.1 タスク管理機能

タスクは、最大 255 個作成することができ、RUN, READY, WAIT, DORMANT の 4 つの状態管理される。又、優先度は 1 ~ 31 の値をとることができる。

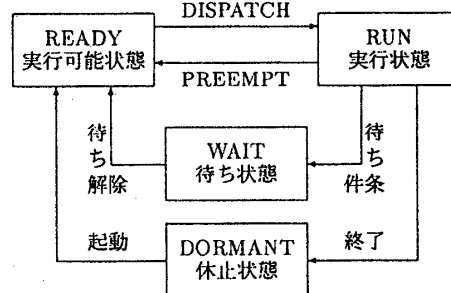


図 2 タスクの状態遷移図

3.2 同期・通信機能

同期・通信機能として、イベントフラグ、メールボックス、セマフォを使用でき、イベントフラグは 8 ビットで最大 16 個まで作成可能、メールボックス、セマフォも、16 個まで作成できる。

3.3 タイマ機能

タイマ機能として、1/300 秒周期のハードウェアタイマ割り込みを利用し、周期起動ハンドラを用意した。

尚、タスク、イベントフラグ、メールボックス、セマフォ、ハンドラはすべてシステム起動時に静的に生成され、動的に定義することはできない。

3.4 システムコール

以上の機能を実現するために表 1 に示すシステムコールをインプリメントした。

表1 MITOS システムコール一覧

タスク管理
sta_tsk ext_tsk chg_pri (ichg_pri)
タスク付属同期
slp_tsk wup_tsk (iwup_tsk) wai_tsk
同期・通信
set_flg (iset_flg) clr_flg wai_flg
snd_msg (isnd_msg) rcv_msg
sig_sem (isig_sem) wai_sem preq_sem
割込み管理
ret_int
時間管理
act_cyc ret_tmr

システムコールの処理過程を簡単に説明する。ソフトウェアのブロック図は図3のようになる。ユーザアプリケーションタスク (TASK.C) が、システムコールを発行すると、まずC言語インターフェース (SYSCALL.ASM) でシステムコールNo. のセット、パラメータパケットの作成等の処理が行われ、ソフトウェアインタラプト (SWI) でOSにジャンプする。OS内のMITOS.A.ASMでシステムコールの種類を判断し、MITOS.C.C内の各システムコール処理ルーチンがCALLされるというしくみである。

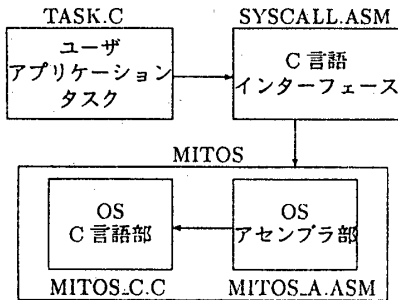


図3 ソフトウェアブロック図

4 評価

プログラムは、開発を容易にするため主要部分をC言語で記述した。プログラムのサイズは、スケジューラ部が約0.5Kバイト、システムコール処理部が約6.7Kバイト、OS全体では約9Kバイトとなった。リアルタイムOSとしてはサイズは大きめである。しかし、学生実験を行うには十分なユーザー領域が確保できるので支障はないと考えている。又、タスク切り替えにかかる時間は、優先度最高のタスクで約300ステップ、優先度最低のタスクで約750ステップであり、学生実験での制御という点では、十分許容できると思われる。

5 学生実験への応用

現在、本学では学生実験として6809トレーニングモジュールを使用したステッピングモータの駆動、プリンタ制御等が行われている。これに、MITOSを用いることにより、複雑な処理が簡単に行なえるようになる。

以下に、ステッピングモータの駆動を例に挙げ説明する。スイッチ1が押された時右回転、スイッチ2が押された時左回転するプログラムを考える。現在は、割り込み処理ルーチンの中で、押

されたスイッチにより変数FLGにRIGHT, LEFTがセットされ、メインプログラムの中でFLGの内容により回転を実行する方法を用いている。ここまでの処理であれば、この方法でもさほど難しくはない。しかし、さらにスイッチ3が押されたらストップする機能を付け加えたり、あるいはスイッチを押している間だけ回転するようにする等、機能の拡張や変更をする場合、プログラムの変更は困難である。

一方、MITOSを用いた場合は、割り込みハンドラの中で、押されたスイッチにより1番のイベントフラグにRIGHT又はLEFTをセットし、RIGHTがセットされていればタスク1が、LEFTがセットされていればタスク2が起動されるしくみになる。この場合、機能の追加は、タスクの追加や割り込みハンドラの簡単な書き換えでよく、機能を変更する場合も該当するタスクのみの書き換えで対処できる。又、周期起動ハンドラを用い、一定周期毎に自動的に回転の向きを変えることも可能である。

更に、多数のタスクを生成し、セマフォやchg_priシステムコール等を用いて複雑に絡み合わせ、ステッピングモータを回転させながらプリンタに出力し、その途中で、ステッピングモータは逆回転を始める...等の処理を行えば、ユーザーはマルチタスクの概念を目で見て理解できる。

このように、学生実験にMITOSを用いることにより実験が簡易化できるだけでなく、リアルタイムOSやマルチタスクのしくみに対する学生の理解も期待できる。

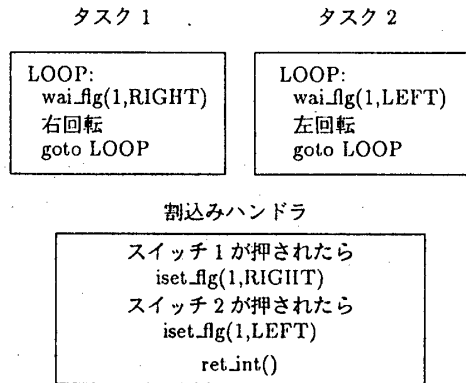


図4 MITOSを用いたステッピングモータ制御

6 まとめ

現行の学生実験の簡易化とリアルタイム・マルチタスクの理解をテーマに、6809上にMITOSをインプリメントした。今後、MITOSの特徴を活かせるように、実験項目の変更、拡張等を検討し、実際に学生実験に取り入れるための準備を進めていく予定である。

参考文献

- [1] μITRON仕様書 Ver.2.01.01.00, トロン協会
- [2] 近藤 元一, 6809マイコン制作実習(上)
- [3] 林 秀房, 浦口 正彦, 石川 知雄, 6809におけるリアルタイムオペレーティングシステムの開発, 情報処理学会第43回全国大会 2L-2 1991
- [4] 石川 知雄, 林 秀房, ITRONにおけるリアルタイムOSの教育, TRONWARE Vol.15