

SEP-3を用いたオペレーティングシステム演習の提案*

大渡 将史†

塩見 彰睦‡

静岡大学大学院情報学研究科

静岡大学情報学部情報科学科

1 はじめに

LSIの集積度の向上により、最近では小規模の組み込みシステムにおいても、ソフトウェア資源の有効利用のためにオペレーティングシステム(以下、OSと記す)が搭載される時代となった。

それに伴い、情報系の学部・学科において、システムプログラム教育の必要性が増してきている。しかし、ハードウェアに密接にかかわっているシステムプログラムの開発演習を行う教材が少ないという状況にある。

本研究では実際にOSを実装することで、OSの基本的な機能の知識を身につけることができる演習の実現を目的とする。演習の実施案の提案と演習で用いるOSの開発、演習環境の構築を行う。対象となるプロセッサは静岡大学情報学部情報科学科において用いられている教育用マイクロプロセッサSEP-3(Shizuoka Educational Processor ver.3)とする。

OSを実装を通して理解する教材の研究としては、MINIX[1]、KITE-OS[2]などが報告されている。MINIXでは、25,000行とソースコードが非常に多いため、短期間の演習として用いるには向いていない。KITE-OSについては、デバイスの排他制御を専用のフラグを用いて行っているため、タスク同士を協調させて実行させることができない。マルチタスクのOSにおいては、タスク間を協調して実行する仕組みは重要である。本研究では、学習の焦点を絞り込み、設計する機能を限定することで短期間の演習で設計できること、タスク間の制御の仕組みについても理解することを目指す。

2 提案する演習

2.1 演習の方針

提案する演習はアセンブリ言語を用いてプログラムを行う。対象者はコンピュータの基本的な仕組み、動作を理解している学部3年生とする。割り当てる時間は、3年次に行われる実験演習の5回程度(4.5時間/回)である。OSの全ての基本機能について演習時間を行うのに十分な時間とはいえない。そこで、今回はプログラムの実行に関する部分である割り込み処理、プロセス管理について理解できる演習を行うことにする。

2.2 開発したOSの概要

提案する演習を実現するに当たり、学生に設計させるためのOSが必要になる。演習において、設計するOSは一般に用いられているものの方が望ましい。そこで、一般に広く使われている μ ITRON[3][4]仕様のOSであるTOPPERS/JSPカーネル1.3[5]を参考にした。

割り込み処理、プロセス管理について機能を限定し開発を行った。 μ ITRONには、標準化により理解や技術者の教育が容易であるなどのメリットがある。学生の理解のしやすさを考慮し、OSの機能の基本要素を実装した。開発したOSが管理するリソースとOSに実装したサービスコールの一覧を以下に記す。

管理するリソース

開発したOSは、16個のタスク、2個の計数型セマフォ、1個の16ビットイベントフラグ、2個のタイマを管理する。サービスコールによりタスクの状態を遷移させ、タスクの切り替えを行う。

サービスコール一覧

搭載したサービスコールを表1に示す。演習ではこれらのサービスコールの設計を行う。時間管理機構については、SEP-3を搭載しているボード(以下、SEPボード)に依存しているため、実装させるのではなく学生に与えることとした。

表1: サービスコール一覧

機能	サービスコール
タスク管理機構	cre_tsk, act_tsk, ext_tsk
同期通信機構	wai_sem, sig_sem, wai_flg, set_flg, clr_flg
割り込み管理機構	dis_int, ena_int, def_inh
時間管理機構	sta_tim, stp_tim, set_tim, get_tim
シリアル通信	serial_write, serial_read

2.3 演習内容

提案する演習では、SEPボードのハードウェア資源や設計時間を考慮したうえで、SEP-3で動作するOSの基本機能を実装する。実装するOSの機能は、割り込み管理、マルチタスク機構、同期通信機構である。演習には、これらの機能を用いるアプリケーションの作成も含む。

アプリケーションは動作が実際に目に見え、理解しやすいという理由から、ロボットアームの制御を課題として与える。制御の対象となるロボットアームとのインタフェースにはシリアル通信を用いる。設計するOSはマルチタスクを考えているため、複数タスクからシリアルポートへの同時アクセスが起こることが考えられる。この不具合を解消するために、シリアルポートのデバイスドライバの設計を行う。

5回で行われる演習は、割り込み管理、マルチタスク機構、同期通信機構、シリアル通信制御、ロボットアームの制御に分けられる。学生が演習を円滑に行うために、それぞれの回において適切な講義を行う必要がある。それぞれの演習について説明する。

第1回 割り込み管理

この回の演習では、サービスコールの呼び出し方について理解し、割り込みの許可禁止の設定方法や各割り込み要因に対する割り込みハンドラを起動する方法について理解することを目的とする。

割り込みハンドラ登録を行うサービスコール(def_inh)と割り込みの許可/禁止の設定を行うサービスコール(ena_int, dis_int)を設計し、実際に割り込みが発生した際の動作をステップ実行により確認する。

第2回 マルチタスク機構

この回の演習では、複数のタスクを切り替えて実行する仕組みとスケジューリング技法について理解することを目的とする。

* The Exercise of the Operating System using SEP-3

† Masafumi Oowatari:Graduate School of Information, Shizuoka University

‡ Akichika Shiomi:Faculty of Information,Shizuoka University

実行状態、実行可能状態、待ち状態の3つの状態を遷移するタスクを定められた優先順位によってCPUが割り当てられるスケジューラを設計する。OSにタスクを生成、起動、終了を行うサービスコール(cre_tsk,act_tsk,ext_tsk)を作成する。作成したサービスコールを用いてSEP-3のメモリ内に複数のタスクを生成し、優先度に従って実行していることを確認する。実行状態でないタスクの状態がどのように保存されているかをタスクコントロールブロックをダンプすることによって確認する。

第3回 同期通信機構

この回の演習は、複数のタスクの同期や排他制御を行う仕組みについて理解する。同期通信機構の基本であるセマフォを理解し、待ちキューを持つセマフォを扱うサービスコール(wai_sem,sig_sem)を設計する。

指定されたデータの送受信が完了するまでポーリングし続ける、排他制御をしないシリアル通信を行うサービスコールと、デバイスの競合が起こるプログラムを学生に与える。学生はそれを実行し、確認することでデバイスの競合について理解する。シリアル通信のサービスコールに、実装したセマフォを用いて排他制御を加え、与えたプログラムを実行しても、デバイスの競合が起こらないことを確認する。

第4回 シリアル通信制御

前回の演習で改良した排他制御のみを行うシリアル通信のサービスコールでは、すべてのデータの送受信が完了するまで、ほかのタスクに移ることができない。ポーリングによるIO待ちを削減し、CPUの資源をより効率よく使用するための方法を理解することを目的とする。

送受信の割り込みが発生した際に、イベントを発生させる機構を設計する。送受信による待ちが発生した場合に、他のタスクが実行されていることをタスクコントロールブロックをダンプや、プログラムカウンタの値で確認する。

最終回 ロボットアームの制御

これまでに作成してきた機能を用いるアプリケーションを設計し、OSの動作や仕組みの理解を深めることを目的とする。

演習ではロボットアームMR-999(イーケージャパン)をそのインターフェースボードRBIO-1(共立電子産業)を介してSEP-3とシリアル接続して行う。実装した割り込みハンドラの登録や、マルチタスク機構、シリアル送信受信の機能を用いてロボットアームの制御を行う。学生は自分で実装したものが、目に見える動きとして返ってくるため、達成感を得ることができる。

ロボットアームの動作をタイマで制御し、物体を指定の場所へ動かすというものを課題として与える。課題には、動作する範囲の中に障害物があるものなど難易度の違うものを複数与える。

目標まで逐次的に関節を動かしていく方法や、複数の関節を同時に動かす方法や、ソースコードによって決まった動作をするものや与えるデータによって違った動作をするものなど、学生は様々な工夫を加えることができる。

3 予備実験・評価

提案した演習の実現可能性を測るために予備実験を行った。予備実験では、実際に演習を行った場合の時間を調べるため、SEP-3に割り込管理、マルチタスク機能、同期通信機構の設計を行った。予備実験はSEP-3の構造やアセンブリ言語を理解している大学院生が行った。予備実験の結果を表2に示す。実際に演習を行う際には、学生はそれぞれの回における講義により、実装方法などを理解しているものと考えたため、表中の時間にはOSの仕様を理解するために要した時間は含まれていない。

表 2: 予備実験の結果

モジュール名	設計時間(時間)
割り込管理	3.0
マルチタスク機構	4.0
同期通信機構	4.5
シリアル通信制御	3.5
ロボットアームの制御	3.0
合計	18.0

表中の割り込管理は、割り込みの禁止/許可と割り込みハンドラを登録するサービスコール設計し、割り込み発生時の処理の動作を確認するまでの時間である(3.0時間)。マルチタスク機構は、タスクの優先順位と状態に従ってCPUを割り当てるスケジューラとタスクの生成、起動、終了を行うサービスコールの実装を行い、正しくスケジューリングされていることを確認するまでの時間である(4.0時間)。同期通信機構は、セマフォを取得、開放するサービスコールの実装と、シリアル通信のサービスコールに排他制御を加え、動作を確認するまでの時間である(4.5時間)。シリアル通信制御は、イベントを発生させる機構としてイベントフラグを作成し、ポーリングによる待ちを削除するまでの時間である(3.5時間)。ロボットアームの制御については、逐次的に関節を動かしていくプログラムの作成を行った(3.0時間)。作成したOSのコードサイズは1.8K語、アプリケーションのコードサイズは0.5K語はである。

演習を実施する際に演習時間内に説明や資料を的確に与え、学生に理解させることにより、今回と同等またはそれ以下の時間とコードサイズで設計することは可能であると考えられる。同期通信機構については、演習の1回分の設計時間がかかっている。適切なサンプルコードを与えるなど設計時間の削減を計ることで演習時間内に設計できると考える。

4 おわりに

本研究では、OSの基本機能である割り込み管理とプロセス管理、デバイスの制御について実装を通して学習するソフトウェア演習を提案した。演習のために、SEP-3用μITRONサブセットのOSの開発を行った。仕様の説明や資料を学生に的確に与えることにより、演習時間内に提案する演習が実施可能であることを示した。

今後の課題としては、実際に演習を実施し、OSの仕様やカリキュラムを改良していくことが挙げられる。今回の演習ではOSの基本機能の一部しか確保していないので、他の機能についても学習できる環境の整備が必要である。

参考文献

- [1] A.S.Tanenbaum, “オペレーティングシステム第2版 設計及びMINIXによる実装”,1998, プレスティンホール
- [2] 末吉敏則、久我守弘、柴村英智 “KITE マイクロプロセッサによる計算機工学教育支援システム” 電子情報通信学会論文誌 D-I Vol.J84-D-I No.6 pp.917-926 2001年6月
- [3] 高田広章, “μITRON4.0仕様書”, トロン協会, 1999年6月.
- [4] 坂村健, “ITRON標準ガイドブック2” 1994, パーソナルメディア社
- [5] TOPPERS/JSPカーネルウェブサイト, <http://www.toppers.jp/>