

●教材報告

μITRONの実装と教材化に向けた取り組み

中国ポリテクカレッジ 生産情報システム技術科 杉山 誠・板坂 政昭
(中国職業能力開発大学校)

1. はじめに

ここ数年で、ユビキタス、携帯端末などに代表される従来のコンピュータシステムとは異なる形態の電子機器が加速的に普及している。従来、このような機器は独自の開発環境と、スタンドアロン形式のプログラムを内蔵して動作するものが大半を占めていた。現在では、高機能化やライフサイクルの短縮によって、従来型の開発スタイルでは対応ができなくなってきており、より効率の良い開発が求められるようになってきている。これらの機器の大きな特徴は、専用化されたマンマシンインターフェースやさまざまな周辺機器を制御する機能が大半であるため、このような部分を共通の仕様で開発する方法が望ましく、何らかのオペレーティングシステムを用いた開発スタイルが主流となりつつある。いわゆる組み込み型OSと呼ばれる範疇のOSである。一方で、不特定多数のハードウェア上で共通して動作するための組み込み型OSを実装した後、このOS上で動作するアプリケーションを開発するためには、これまでのPC上でのアプリケーション開発ではほとんど必要としなかった、高度なハードウェアの知識を兼ね備えた開発者が必要とされる。このため、これまで以上にこの分野の技術者不足が顕在化しており、このような機器をターゲットとした開発スタイルを学ぶための教材や標準的な教育コースを設定することは、非常に有意義なことと考えられる。そこで、筆者らは、近年この分野で広く採用され、またオープンソース

であるため比較的自由に利用できるリアルタイムOSであるITRONに注目し、教材として活用している。本稿では、当科のカリキュラムで実施しているリアルタイムシステムでの内容に加えて、今年度開催している企業人スクール「リアルタイムOS組み込み実践」で利用している教材の概要について報告する。

2. ITRONとは

TRON (The Real-time Operating-systems Nucleus) ^[1] は、1984年東京大学の坂村健教授によって提唱されたリアルタイムOSである。このOSは「どこでもコンピュータ」をキーワードに開発され、現在、組み込み分野での製品の約半数は、TRON系のOSを採用しているといわれており、次のような特徴を有している*1。

●弱い標準化

μITRONはトロン協会ITRON部会が中心となって策定されたリアルタイムカーネルの仕様である。特定のCPUやハードウェアを前提としておらず、実際に利用する場合には自らOS (カーネル) を制作、実装する必要がある

●オープンアーキテクチャ

TRONプロジェクトでの成果は仕様として公開され、比較的容易に入手することができる。この仕様

*1 ITRONは、工業用途に利用しやすいように設計されたTRONの種類をいう。

に基づいてTRONに準拠したOSを作ることは自由であり、独自の製品に実装して、製品の開発を行うことに何の制約も受けない。

●省資源

静的APIの導入など、小規模の組み込み環境の制約を考慮してリソースを消費しないコンパクトさを追及している。

●教育面での支援

2003年にNPO法人として設立された関係組織であるTOPPERSプロジェクトによって公開されている μ ITRONのカーネルをベースにJSP (Just Standard Profile) カーネルの開発と配布が行われている。TOPPERSプロジェクト^[2]ではこれと平行して、オープン・ソース・ソフトウェアを用いた教育コースや教材の開発を行い、これらオープン教育コンテンツとして公開している。

このようにTRONはオープンアーキテクチャである多くの恩恵を受けることができる反面、実装された実体がないため、実際に利用する際には、カーネルの実装や周辺機器のドライバなどをすべて自前で用意する必要があり、製品の開発にはある程度のスケールメリットと技術力のある企業でなければ参入が難しかった。近年では、前述のTOPPERSプロジェクトの活動などにより、多くのハードウェア上で動作が可能なJSPカーネルの提供も始まったことから、手軽にTRONに触れる機会が増えつつある。

3. 教材としての μ ITRON

ITRONを学ぶうえで最も大きな問題は、適切にカーネルの実装されたターゲットシステムと開発プラットフォームであるホストコンピュータで動作するクロス環境を用意しなければならない点にある。TOPPERS/JSPから、いくつかのターゲット上で動作するカーネルが提供されており、その多くはGNUプロジェクト^[3]によるオープンソースの開発環境を用いることができるようになってきており、有力な選択肢の1つにあげられる。もう1つの大きな問題は、

近年、組み込み OS として注目されている Linux などに比較すると、圧倒的に開発ノウハウに関する情報が少なく、フレームワーク的な開発環境は皆無である。このため、初学者には、リアルタイム、マルチタスクOS特有のプログラミングスタイルを正しく理解するための順序立てたサンプルなどが必要であると考えられる。加えて、目的とする負荷装置を思いどおりに制御するためにはターゲットとなるハードウェアの動作を正しく理解させる必要がある。そのほかに、 μ ITRONを用いて行う実習を通じて会得してほしい技術要素は以下のとおりである。

1. クロス開発環境の仕組みとその必要性
2. プログラミングでハードウェアを制御する仕組み
3. オペレーティングシステムの仕事 (OSを利用するメリット, デメリット)
4. オペレーティングシステムによるタスク管理
5. タスク間通信
6. ネットワークプログラミング
7. ハードウェアデバイスの制御

3.1 学生向け教材

比較的ハードウェアの知識の少ない本科の学生に対しては、まずTRONの動作概念を理解させた後、いくつかのサンプルを使ってハードウェアの制御方法を習得させる。その後、必要に応じてハードウェアドライバや実際に動作するアプリケーションの制作を行う方法が適切であると考えられる。そのため、手始めにTRON/OSのプログラミングスタイルを体験できる実習環境としてTOPPERS/JSPとGNU環境を利用することを前提に開発された学習キット*²を用いた実習を採り入れた。このハードウェアはルネサステクノロジ社製の32bitマイクロコントローラであるSH-2を搭載するメインボードといくつかの負荷装置を実装したオプションボードから構成されている。オプションボードにはLCD表示装置、7セグメント表示装置、マトリックスLEDのほかステッピングモータのコントローラが用意され、いくつかのサン

*2 (株)共栄エレクトロニクス社製「KED-SH100/KED-EXT101」



図1 学習キットとホストPC

プログラムも用意されており、手軽にTRONのアプリケーションの動作確認を行うことができる。また、タスクの管理方法などについての解説書やサンプルも充実している。残念なことには、これらの負荷装置を制御するためのサンプルの多くは、排他制御などマルチタスクOSを意識したドライバとして制作されたものではなく、いわゆるTRONらしさに欠けている。したがって、現状ではリアルタイム処理を意識したプログラミングが十分に体験できるレベルには至っていない。このため、ドライバの制作も含めた実習課題および資料の整備が必要であると考え。また、メインボード上にはEthernetI/Fが準備されているためJSP上で動作するプロトコルスタック (TINET) を利用が可能で、さまざまなネットワークアプリケーションの作成もできる。図1に実習環境を示す。

3.2 社会人向け教材

本年度開講した企業人スクールでは、何らかの組み込み製品の開発に携わっている受講者も多く、ITRONを使った製品の具体的な例として、ハードウェアの設計を含めて学習したいとの要望が多かった。そのため、自由にハードウェアの変更や周辺装置の追加ができ、かつ低価格にて実習環境を整備できる、秋月社製AKI-3069F/NET^[4]を利用した。本ハードウェア上でも、前述したTOPPERS/JSPとTINETが利用できる。また、RedHat社が提供しているeCosのサブセットであるRedBoot^[5]を用いると、GNUデバッガーを用いて、リモートデバッグ環境を整備す

ることができる。ただし、GNUデバッガーはタスクの動作をモニタできる機能はなく、大規模なアプリケーションの開発にはマルチタスクOSに対応したデバッガーの整備が望まれる。この環境下であれば、小規模な製品の開発までに必要な環境を手軽に入手できる。本実習で利用したツールを以下に示す。

AKI-3069F/NET	秋月電子通商
gcc-3.4.4	GNU
binutils-2.16.1	GNU
newlib-1.13.0	RedHat
insight-1.6	GNU
jsp-1.4.2	TOPPERS
tinet-1.3.2	TOPPERS
redboot-aki3068net	RedHat

これらのアプリケーションは、すべてオープンソフトウェアであり、該当するサイトよりソースコードまでを無償で入手することができる。動作環境は、Windows下のCYGWINや仮想コンピュータを実現できるVMWareをはじめ、UNIX上でも開発が可能であり、日ごろ使い慣れた開発プラットフォームを利用することができる。

3.2.1 カリキュラム

企業人スクールは次のような順序で進める。

1. TRONの概要
2. 開発環境の整備 (JSP環境を含む)
3. タスクオブジェクト (実習を含む)
4. 非タスクオブジェクト (実習を含む)
5. ハンドラ (実習を含む)
6. 割り込み処理 (実習を含む)
7. LCDドライバ、ドライバタスクの導入

これらのうち、LCDへの表示部分は、LCDのコントローラへの決められた手続きによる初期化が必要となり、実際にハードウェア上での信号の授受を正しく理解する必要がある。さらに、マルチタスクOS下では、排他制御が必須となるため、ドライバ (ドライバタスク) として用意することが好ましい。図2に制作するLCDドライバの動作概要を示す。同図

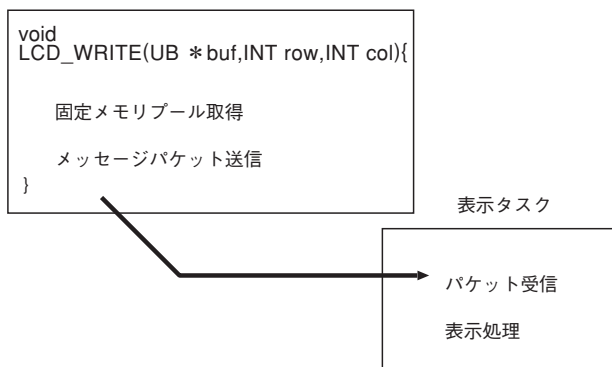


図2 LCDドライバの動作概要

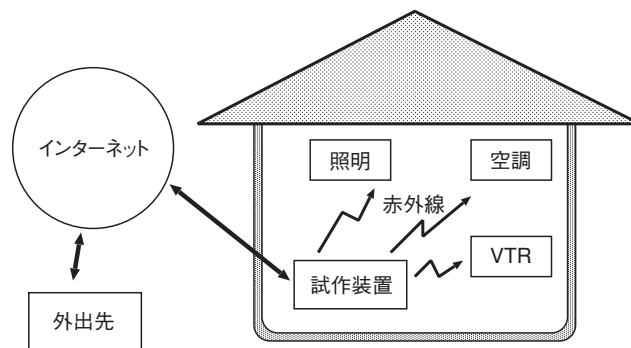


図3 利用イメージ

のように、あらかじめ実際の表示タスクは、TRONの同期オブジェクトであるメールボックスを用いて、常に処理を実行できる状態となっており、ラッピング関数であるLCD_WRITE関数からのメール送信を待っている。一方、メールボックスで利用する固定メモリの資源数を1とすることで、他のタスクからこのラッピング関数が呼び出されても、資源が取得できないため、待ち状態となり、排他制御が正しく行われることとなる。このように、ITRONで用意されている同期オブジェクトを正しく利用するためのスタイルを、実習を通して習得することができるように配慮している。本講座では、このようなドライバの開発に時間を割き、以後の開発にも再利用することができる汎用的なドライバの開発までを基本的な実習に含めている。開発したドライバは、後述する課題でも利用する。

3.2.2 製品開発課題

今回の実習では、一般家庭に手軽に設置でき、ネットワークを通じて家電製品を制御するための端末装置として利用できるネットワーク対応型の赤外線リモコン（ホームセキュリティ端末）の試作を課題として取り上げた。この装置は前述したAKI-3069F/NETボードに、表示装置として汎用的なLCDを装備し、家電製品など外部の機器を機器自体に特別な改造を必要とせず効率よく制御するためのデバイスとして、赤外線の入出力装置を付加した。そのほか、家の中の状態を知るために、温度、光センサを取り付けた負荷ボードの設計・制作を行った(図3)。ユーザが、適当なネットワーク端末装置か

ら、この装置を介して、自宅の照明機器のON/OFFや室温を確認し、必要に応じて照明やエアコンの電源を投入したりするなどの機能を持つことを想定した課題としている。課題では、赤外線通信を行うために、正確な時間制御が必要となるため、タイマ処理や割り込み処理を十分に活用することが要求される。赤外線の送信タスクも、前述したLCDのように排他制御が必要となるハードウェアデバイスとしての配慮が必要となる。この課題では、TRONのプログラムやハードウェアの知識のほかに、TCP/IPを用いた通信アプリケーションの制作法までを学ぶことができるため、さまざまな要素を含んだ適当な開発事例であると思う。図4に試作したボードの外観を示す。また、UNIX上のGTK環境で制作した、ボー

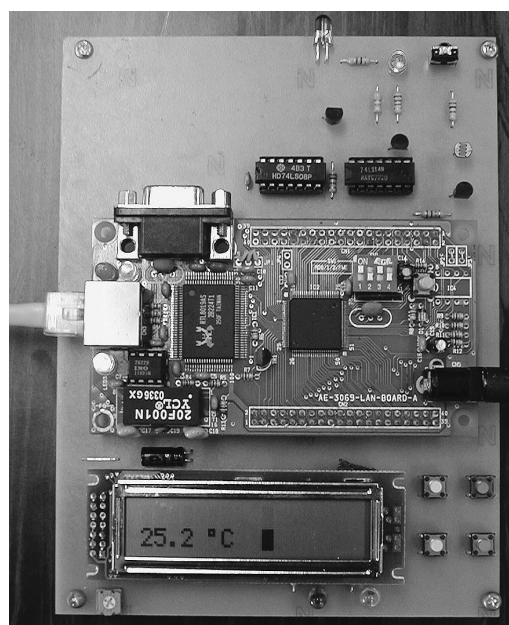


図4 試作ボードの外観



図5 制御アプリケーションの例 (Linux/GTK)

ド制御用のアプリケーションの実行画面を図5に示す。そのほかにも、制御用アプリケーションは、ネットワークプロトコルスタックが用意できる環境であれば、容易に制作することができるため、PDAや携帯電話のような端末装置からも利用が可能である。この課題では次のような要素について習得できるように配慮している。

1. 複数の機能ブロックの効率のよいタスク化
2. 赤外線通信プロトコルの概要理解
3. 赤外線入出力ハードウェアの設計・制作
4. 赤外線入出力ハードウェア用ミドルウェアの開発
5. TINETを使った通信アプリケーションの開発方法
6. ソケット通信の方法

本企業人スクールは受講者の大半はC言語などを用いて、何らかのアプリケーションの開発従事者がほとんどであり、実際の開発を例に取った講座の設定は非常に好評であった。実習では利用するハードウェアや開発環境のほかに、サンプルコードを多用した自作のテキストを中心に、60時間のコースで受講者全員が本課題を要求仕様を満たすシステムの開発を終えることができた。

4. まとめ

当科の学生に対しては、前述した教材上でTOPPERS/JSPを用いた μ ITRONのプログラミングを中心に、リアルタイムOSのプログラミングスタイルを体験させることができた。同時に、高度なOS上ではあまり意識しなかったプログラムが「どうやって動作しているか」を実感することができたように思う。これまでに紹介したような教材は、情報系の学生にとっては触れる機会の少ないハードウェアとアプリケーションやミドルウェアと融合したシステムの構築事例を、比較的身近に感じることのできる教材として非常に有効なものであると考えられる。一方、企業人スクールで利用している環境は受講者が利用する用途に応じて、自由にカスタマイズすることができる実際の製品のプラットフォームとして有効に利用できるなど、少なくとも小規模な事業所向けの開発ツールとしては有益であると考えられる。今後、ストレージデバイスなどの追加、ミドルウェアの開発行う予定である。また、テキストをはじめとした成果は、何からの形で配布できるようにしたいと考えている。

社団法人トロン協会が行っているアンケート調査^[1]の内容から、2年前の結果と比較して、有効回答数が2倍になっていることから、組み込み分野の需要はますます増大する傾向にあると考えられる。その一方で、同アンケートでは技術者の不足、開発環境やツールの不足などの問題が半数を占めている。このような現状から、さまざまな用途に利用できるリアルタイムOSをミドルウェアを含めて実装したハードウェアと、動作するアプリケーションを効率よく開発の行える環境の整備は非常に重要な要素の1つであり、この分野に秀でた技術者の育成が今後ますます重要になってくると考えられる。

<参考文献>

- [1] TRONプロジェクト <http://www.assoc.tron.org/>
- [2] TOPPERSプロジェクト <http://www.toppers.jp/>
- [3] GNUプロジェクト <http://www.gnu.org/>
- [4] 株式会社 秋月電子通商 <http://akizukidenshi.com/>
- [5] Red Hat K.K. <http://www.redhat.com/>