

OS 教育支援における可視化環境の実現

西野洋介† 大角圭吾‡ 早川栄一‡

† 拓殖大学大学院 工学研究科 ‡ 拓殖大学 工学部 情報工学科

1. はじめに

情報工学を学習する学生にとって、システムソフトウェアについて、その動作を学習し、理解することは重要である。情報工学のカリキュラムにおいても、コアカリキュラムとして必須項目となっている[1]。この分野は計算機の根幹であり、その技術者の育成は重要な課題である。現在、こういった問題を解決するために、様々な学習環境の研究・開発が進められている[2][3]。

これらの背景の中で、我々はシステムソフトウェア教育支援環境の開発プロジェクト「港」[4][5][6][7]を進めている。「港」はシステムソフトウェアの統合的な教育支援環境であり、OS、リアルタイムシステム、プロセッサの三つの部分から構成されている。これまで、我々は可視化を中心とした「港」における各部分の開発を進めている。

この開発の中で、各部分の関連付けに問題があることが明らかになった。現在の支援環境の構成では、システムソフトウェアにおいて重要となる相互関係が見えにくく、視点を変えた可視化が行いにくい。また、可視化部分を再利用することが難しく、教育支援環境として教授者が教材を作成し、多様な可視化環境を学習者に提供することが困難である。

本研究の目的は、「港」における OS 部分の可視化環境の開発である。上述した各部品間での相関性および、部品の再利用性の問題点を解決する。これにより、教授者の演習環境構築の手間を減らすことができる。

2. 問題分析

OS の学習、教育に対して、従来型学習方法の問題点を次に挙げる。

- (1) 演習環境の構築に手間がかかる
- (2) 学習項目における視点が固定である
- (3) OS 学習の初学者への教育支援環境が少ない
- (4) 視覚的な教育支援を行う環境がない

3. 要求仕様

前述の問題点を解決するために、教育支援環境の要求仕様を次のように定めた。

- (1) 学習項目に応じた演習環境が容易に構築できること
- (2) ある学習項目に対する視点の粒度を変更できること
- (3) 概念に重点をおいた教育支援ができること
- (4) 視覚的な教育支援ができること
- (5) 動的な状態遷移を追えること

Implementation of visualization environment in OS education support

Yosuke Nishino Keigo Osumi Eiichi Hayakawa

Faculty of Engineering, Takushoku university

4. コースウェア

次に本環境を使用した講義計画の一例を図1に示す。概念の講義からソースコードに触る実践学習まで一連した講義を支援することができる。

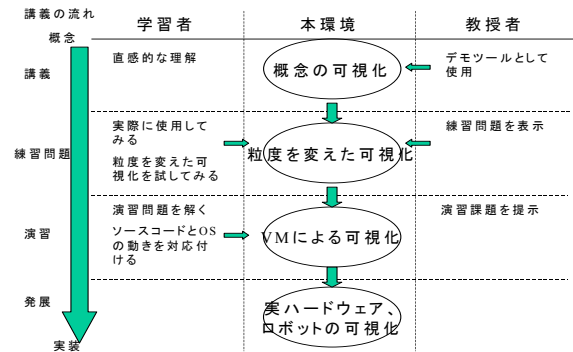


図1. コースウェア

5. 全体構成

3. 要求仕様で述べた設計方針を基に、全体設計を行った。次に、本教育支援環境の全体構成を図2に示し、各部の詳細な設計について次に述べる。

本教育支援環境では、要求仕様で述べた学習項目に応じた可視化環境を実現するために MVC(Model-View-Control)モデルを基とした 3 部分から構成されている。この全体構成においては、可視化の対象となるモデルをプラグブルにしている。その際にはコントローラおよび可視化部には変更を加えずに差し替え可能となっている。各部の詳細設計を次に示す。

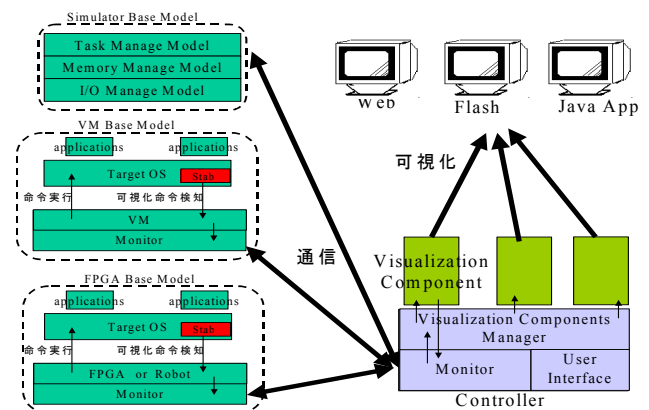


図2. 全体構成図

5. 1 モデル設計

可視化対象となるモデルは 3 部で構成されている。このように 3 部形態をとることで、教授者は用途や学習者の理解に応じた演習環境、デモンストレーションツールを容易に準備することができ、手間を減らすことができる。

- シミュレータベースモデル

シミュレータベースの可視化対象モデルは OS の機能を概念の学習に適した抽象化を行い、その動きを可視化されるモジュールである。

- 仮想マシンベースモデル

仮想マシンをベースとした可視化対象モデルは、実際のカーネル（学習対象 OS）を仮想マシン上で動作させ、その動作の様子を可視化するものである。

- FPGA ベースモデル

FPGA をベースとした可視化対象モデルは VM ベースと同様に、実際のカーネル（学習対象 OS）を FPGA によって実装されたオリジナルプロセッサ上で動作させ、その動作の様子を可視化するものである。すでに先行研究として実装されているプロセッサ[3]を利用する。

5. 2 可視化コンポーネント設計

5. 2. 1 可視化の方針

可視化を行う際の方針として、配色に工夫をつけた。OS の各状態や状態遷移を赤、緑、黄色の信号色を使って表現することで、学習者の直感的な理解を得るものである。具体的には、タスク管理において、実行中のタスクは緑、実行待ちのタスクは黄色、停止中のタスクは赤で表すといったものである。

5. 2. 2 可視化コンポーネント

可視化を行う可視化コンポーネントはシミュレータベースモデルおよび仮想マシンベースモデル各モデルの状態遷移のデータを受け取り、可視化データに変換するものである。図 3. 全体構成で述べたように、可視化コンポーネントにおいて可視化された情報は、Web、Flash、Java と多様な出力形態を持つ。これは、環境にとらわれず、様々な出力において可視化できるように構成されたものである。これにより、教授者はその学習環境に応じて出力するアプリケーションを選択すればよいことになる。

可視化コンポーネントは OS のハードウェア資源、ソフトウェア資源両者の組み合わせによって、構成されるものとする。各資源については同様に、粒度を設け、表示、非表示を選択することができる。これは、すべての情報を一度に可視化してしまうと、学習者はその情報の多さに注目すべき点を見落とし、余計な混乱を招く恐れがあるからである。

次にコンポーネントの一覧を示す。

表 1. 可視化コンポーネントの一覧

	粒度大	粒度小
可視化コンポーネントの種類	タスク、アドレス空間、ファイル、タスクスケジューリング、タスクの状態、仮想メモリ、ファイルシステム、割り込み	プロセッサ、メモリ、キーボード、ディスプレイコンテキストスイッチ、排他制御

5. 3 実現

前述した設計から、試作を行った。試作の目的は OS 教育に

おける可視化環境の有用性および、学習項目に応じた可変的な構造をもつ可視化環境の有用性の確認である。以下の例では、タスクスケジューリングの可視化について以前に作成した物と、コンポーネントを利用して作ったものの比較である。次の表 2 にまとめる。

表 2. コンポーネント使用前と使用後の比較

	コンポーネント不使用	コンポーネント利用
工程日数	約 3 か月	約 1 週間
コード数	約 1000 行	約 100 行
言語	C++	JAVA

これらの結果から、大卒での手間の削減につながっていることが言える。試作画面を次の図 3 に示す。

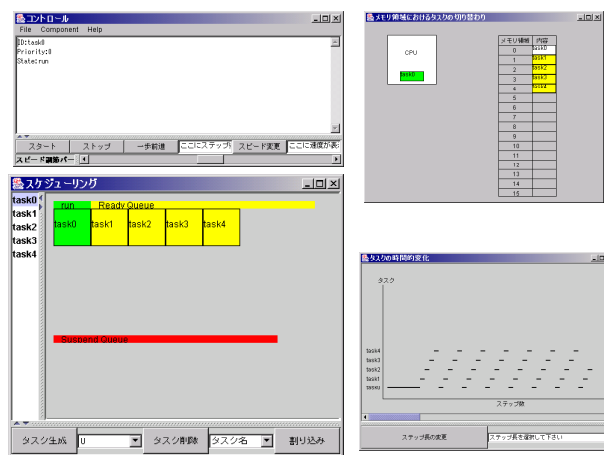


図 3. 実現画面

6. おわりに

本報告では、システムソフトウェア教育支援環境「港」の概要と、「港」の一環である、OS 教育支援環境の可視化の設計について述べた。本研究を通して OS の教育を視覚的に支援する環境を実現し、これにより学習者の理解の促進および、教授者の講義、演習における教材開発の手間を削減することができるようになった。今後は本環境のシステムとしての評価および、可視化を用いた OS 教育支援の教育効果の評価を行う。

参考文献

- [1] 社団法人情報処理学会:大学の理工系学部情報系学科のための コンピュータサイエンス教育カリキュラム J97
- [2] 末吉、久我、柴村:“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-1 VolJ84-D-1 No.6 2001
- [3] 美馬、田中、佐藤、有田:“計算機教育向けシステム KERNEL1 の設計”第 64 回情報処理学会全国大会 IZB-02 2002
- [4] 吉田、早川:“ロボットプログラミング学習支援環境の開発”FIT2003,N-026 2003
- [5] 横山、早川:“自走ロボットを利用したリアルタイムシステムの学習支援環境の開発”電子情報通信学会 教育工学研究会 2002
- [6] 下川、西野、早川:“システムソフトウェア教育支援環境「港」における FPGA を利用した演習環境の開発”電子情報通信学会教育工学研究会 2003
- [7] 米田、早川:“体験学習モデルによる OS の学習システム”情報処理学会コンピュータと教育研究会,2002-CE-67 2003