

## 学習支援のための OS の可視化環境の設計と実現

4 L - 4

伊藤 能康†, 早川 栄一†, 並木 美太郎†, 高橋 延匡†  
†東京農工大学 ‡拓殖大学

## 1. はじめに

オペレーティングシステム(OS)は、コンピュータサイエンスを学ぶ学生が理解していなければならない重要なソフトウェアの一つである。しかし、OS は非同期な割込み動作のため、動作の概要を理解することは容易ではない。また、CPU などのアーキテクチャに強く依存しているため、さらに理解への敷居を高めている。OS の教育環境として Nachos[2]などがあるが、可視化環境ではなく、予備知識の少ない学生を支援しにくい。そこで我々は、OS の基本概念として割込みや仮想化、その動作や管理機構などの容易に理解をできるようにするために、OS を可視化し、OS の学習を支援する環境を構築することにした。本環境の目的は次のとおりである。

- (A) OS の基本的概念を直感的に理解させる
- (B) OS の実装を知ってもらう

## 2. 可視化環境の全体設計

可視化環境の利用形態は学習効果を考え、ただ可視化された表示を見せるだけではなく、実際に学習する OS のソースの変更しながら、OS の概念や動作、機構などを理解させることにした。

ところで、OS の可視化環境に必要なとされる機能を直ちに決めることはできないため、我々は可視化ツール初版[1]を設計し、実現を行い、OS 演習内で利用、評価した。評価の結果、次のようなことがわかった。

- (1) アニメーションを用いた OS の動作の可視化は好評で、効果が高い
- (2) 講義だけでは、OS の概念を正確に理解していなかった
- (3) ソースコードの解読に手間取っていた  
また学生からの要望として、次のようなものがあった。
- (4) OS の動作をソース上のどこが処理しているのかを表示してほしい
- (5) コンテキストスイッチの具体的なやり方を見たい

この評価結果から可視化環境の全体設計を行った。まず、本環境の利用対象者は、OS の基本的な概念や動作原理、基本的な機能などを学習途上にある学生とした。具体的には、東京農工大学工学部電子情報工学科コンピュータサイエンスコース 3 年次の学生である。

## 3. 可視化環境の設計

## 3.1 可視化環境の設計方針

初版の評価結果を考慮して本可視化環境の設計方針

針を次のように設定した。

- (1) OS の学習初期段階からの利用を考慮する
- (2) 基本概念である仮想化の理解を支援する
- (3) OS の基本機能だけを可視化する
- (4) 実行環境は学生の利用しやすいものを用意する
- (5) ソースコードと動作との対応を強化する  
また可視化環境として、OS によって可視化内容が異なるような場合や利用環境などの、教育者側への配慮を考慮した。このための方針は次のようになる。
- (6) 可視化環境の変更を容易にし、汎用性を重視する

## 3.2 可視化の設計

可視化の表示は初版を継承し、OS の動作をアニメーションやアイコンを用いて表示を行う。これはタスク管理、割込み、SVC などの OS の基本機能やそれらとハードウェアとの関連などを可視化する。これらは OS の管理機構についての可視化だけではなく、CPU を仮想化したものがタスクである、といった概念の理解を支援するための内容を含める。具体的には図 1 に示すように、学生が理解しやすいような表示を行う。

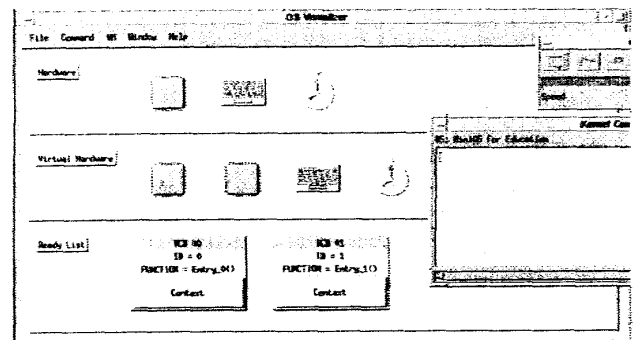


図 1. 全体画面

次に個々の可視化表示の設計について述べる。

- (1) ハードウェアをアイコン表示する
- (2) 割込みなどをアニメーション表示する (図 2)
- (3) タスク管理は、リストの内容やその動作をアニメーション表示する (図 3)

タスク管理リストの接続の様子はボックスや矢印などを用いて示し、その変更の様子を、動作の確認が容易になるように残像を残したアニメーション表示によって示す。タスク管理情報は、ボックスを選択することで詳細なコンテキストを表示する。

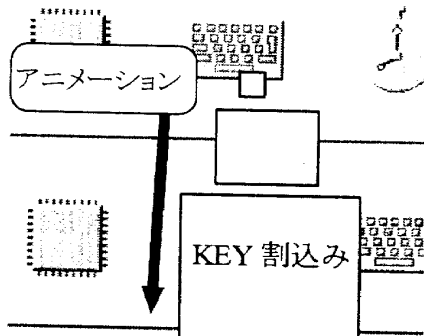


図 2. 割込みの表示

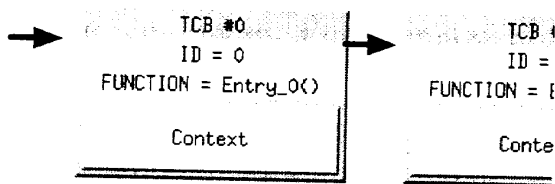


図 3. タスクリストの表示

- (4) コンテキスト操作をアニメーション表示する  
コンテキストスイッチ動作は、CPU アイコンから保存対象のタスク管理リスト内のレジスタ保存領域へのコピーアニメーションなどを表示する。
- (5) 学生から要望のあった、ソースレベル・アセンブルレベルの動作追跡を表示する

### 3.3 利用環境の設計

本可視化環境は、より多くの学生に利用してもらうことを考え、実行環境を汎用性の高い X Window 上に移行した。また、可視化対象 OS の動作環境は同様に汎用性を考慮して、仮想機械から CPU エミュレータ上で動作させるようにした。ハードウェアはハードウェアシミュレータを用意する。

ここで、可視化対象 OS を動作させるアーキテクチャとして Alpha アーキテクチャを選択した。Alpha アーキテクチャでは Privileged Architecture Library (PAL) コードによってアーキテクチャを隠蔽可能であり、また複雑な CPU アーキテクチャを持たないため、学生が理解しやすいからである。

### 4. 可視化環境の実現

可視化環境は、多くの環境で利用可能にするために、実現は Linux 上でいき、GUI は Motif を用いて構築する。CPU エミュレータは実装を容易にするために、浮動小数点演算は省き、実際に OS やタスクの実行に必要なインストラクションに限定して実現した。ハードウェアシミュレータが提供する割込みなどではできるだけ実際のハードウェアに近い動作するように、独自に割込みトリガや優先度などを設定した。

可視化環境の全体構成を図 4 に示す。

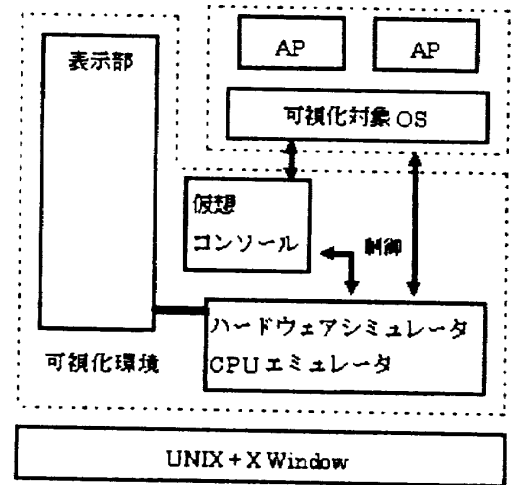


図 4. 可視化環境の全体構成

なお可視化環境の実現規模は、OS 非依存部として CPU エミュレータが言語 C のソースコードで約 3000 行、可視化部が約 4000 行で、OS 依存表示部が 2000 行程度である。可視化対象 OS は機能を限定し、1000 行程度のミニ OS を用意した。

### 5. おわりに

本報告では、コンピュータサイエンスを学ぶ学生が OS の動作や管理機構を理解するための可視化環境の設計と実現について述べた。本可視化環境によって、学生が OS の学習を行うことが容易になった。

今後は、実際に学生に利用してもらい評価する予定である。

### 謝辞

本研究は、文部省科学研究費基盤研究 (A)09358004 により行われた。

### 参考文献

- [1] 伊藤能康他: 学習支援のための OS の可視化ツールの設計と実現, システムソフトウェアとオペレーティング・システム 74-30, マルチメディア通信と分散環境 81-30, 1997
- [2] Wayne Anderson: The Nachos Instructional Operating System, The College of Engineering, University of California at Berkeley, <ftp://ftp.cs.berkeley.edu/ucb/nachos/>
- [3] 日本デジタルイクイップメント株式会社: Alpha AXP アーキテクチャ概要, 共立出版株式会社
- [4] Digital Equipment Corporation: Alpha Architecture Handbook