

可視化によるOS教育支援環境の実現と評価

西野洋介, 早川栄一, 高橋延匡

拓殖大学大学院工学研究科、拓殖大学工学部情報工学科

計算機科学を学ぶ学生にとってオペレーティングシステム(以下 OS)の理解は必須である。しかし従来のテキスト、講義による学習では、OSの動作が見えず、非同期で実行が行われるので、学習者はOSの動作と構造の対応がイメージしにくい。このような問題に対して、我々は可視化を中心としたOS学習、教育を支援する環境を提案した。この研究はブラックボックス化しているOSを可視化という手段によって、OSの機能、構造を表示し、教育、学習支援を行う環境を提供することで、学習者の理解を促進させることが目的である。本教育支援環境では、特に重要なOSの概念から教育支援を行い、また可視化部分を可視化モジュールとして独立させることで可視化環境の構造に柔軟さを与えている。これにより学習者のレベルに応じた学習を行うことを可能にする。我々はプロセス管理、メモリ管理について実現し、マルチプログラミングの概念、プロセスとメモリとの協調の概念などの学習が可能となった。また、本学情報工学科におけるOSの講義においてデモンストレーションを行い、評価を行った。

Realization and evaluation of OS educational support environment by visualization

Yosuke Nishino, Eiichi Hayakawa and Nobumasa Takahashi
Takushoku University

The understanding of operating system (OS) is indispensable for the student who learns computer science. But, the execution of the OS and the correspondence of the structure are hard for a learner to image because the movement of the OS can't be visualized and a usual textbook and learning by the lecture are executed by the event that occurs asynchronously. An environment to support the OS learning that utilized visualization is designed and implemented. The function of the OS is indicated, and a purpose is to make it promote learner's understanding by providing the environment where it is brought up and learning is supported by the means of the visualization. Education is supported from the operating system concepts, and a flexible visualization environment is provided in this education support environment by making a visualization part become independent as a visualization module. The matter that the learning that depended on a learner's level by this is done is made possible. It is animated about the process and the virtual memory, and learning such as a concept of multi-programming and the cooperation with the process and the memory management became possible. Moreover, evaluation was tested at the lecture on the OS in our compute science course, and the validity of this system was confirmed.

1. はじめに

オペレーティングシステム（以下 OS）はハードウェア、ソフトウェアを管理し、仮想化する最も重要なシステムである。ゆえに計算機科学を学ぶ学生にとって OS の学習、理解は最重要課題の一つである。ゆえに大学を含め多くの教育機関においてシステム分野の教育がなされている。

しかし、計算機科学を学ぶ学生が OS の概念、動作、機構などを理解するのは難しい。その大きな理由の一つとして OS がブラックボックス化してしまい、動作が実体として見えないことが考えられる。

このような問題に対して、我々は可視化を中心とした OS 学習、教育を支援する環境を提案した[1]。この研究はブラックボックス化している OS を可視化という手段によって、OS の機能、構造を表示し、教育、学習支援を行う環境を提供することで、学習者の理解を促進させるものである。本稿ではまず計算機科学を学ぶ学生を対象として、OS 学習における教育環境と本研究の目的を述べ、OS 教育の支援環境の実現および評価について述べる。

2. 研究の目的

本研究の目的は OS を学習、教育するための支援環境の開発である。具体的には可視化という手段により OS の動作を具体化することで、従来のテキストによる OS 学習に比べ、OS の動作、機構が簡単に理解を促す事ができる教育支援環境を学生に提供することである。

なお、本研究は計算機科学、特にシステムプログラミングに関する教育支援環境の開発の一環として位置付けられている。

3. 問題分析

3.1 現在の OS の教育環境についての問題点

現在、一般的な科目での教育環境では、講義による説明とそれと並行してテキストを用いて学習させている。しかしテキストをベースとした学習環境においては学習者は OS の動作が把握しにくい。なぜならば OS はその性質上からユーザからは直接見えない、いわゆるブラックボックスとなっているからであ

る。このため、学習者は OS の概念や動作の様子をイメージすることができないという問題点がある。

その他の一般的なプログラミング言語の講義やソフトウェア開発などの演習では、計算機上でアプリケーションを作成し実行する学習環境が与えられる。これは、講義による説明とテキストによる学習だけでは習得するのが難しく、自らものを作り体験することにより学習内容を習得し、学習者自身の力になるからである。OS を学習する際にも講義による説明とテキストを用いた学習の後に実験演習できる環境があれば、構造の理解や動作のイメージをするのに役立ち、学習した内容をより確実に習得できると考えられる。

3.2 既存の OS 教育支援環境における問題点

従来のテキストによる学習や、講義による学習では、概念や実装コード例を示すことはできるが、実際に OS を操作して実験しながら学習することは難しい。

これに対して、実装が簡易な OS を用いてユーザに実装のコードを読ませたり、変更させたりするアプローチがあるが、OS は非同期的な割込みによって動作するので、テキストベースによる学習では、ユーザはその動作概念がイメージできず、結果として OS の理解を妨げる原因になってしまう。また OS のソースコードと OS の実行との動作対応とれず、OS を概念的には理解したが実装レベルとしての理解が得られにくい。そこで実際に OS を動かしながら、動作と OS のコードとの対応づけがなされた学習環境のニーズが生まれてくる。

3.3 既存の OS 教育支援教材について

ここでは既存の OS 教育支援教材についていくつかの例を挙げ、それらの特徴と問題点を挙げる。

(1)MINIX[2]

MINIX は、学習に適した OS と目的とするためソースコードに教科書とも言えるコメントが大に書かれている。しかし学習者にとっ

ては、サイズが大きい。

(2) Nachos[3]

Nachos 小型であり、ソースコードにもコメントが丁寧に書いてあるため OS 学習に向いていると言える。しかしソースコードを変更する以前と変更した後の変化の様子は目に見えるものではなく、果たして改良がっているのかどうか確認する方法はない。これは MINIX にも共通する問題である。

(3) アプレットによる可視化[4]

インターネット上には OS の可視化に関する様々なアプレットが存在している。しかしながらこれらはそのアプレット単体での動作しかすることができない。また、一つのアプレットで様々なアルゴリズムを試してみたり、様々な動作が協調して動作する様子をあらわすことができなかつたりとシステムとしての柔軟さに欠けるといった問題点がある。

4. 対象者

本教育支援システムでは教育支援システムでは OS の講義を行う教育者、特に OS 学習における初期段階の講義を持つ講師を対象とし、教育者への講義支援ツールとしての使い方を想定している。OS について説明を行う側にとっては、可視化環境をテキスト、資料とともに利用することでテキストだけの解説に比べより具体的な講義にすることができる。また、OS の実験演習における課題作りや実験を簡単にシミュレートすることができるようになる。

また、情報工学科もしくはそれに相当する学科に在籍する、OS 学習における初期段階の学生も対象とする。こちらは座学ではない演習、実験において学習者自身が学習ツールとして使用されることを想定している。

5. 教育支援の方針

(1) 概念に重点をおいた教育支援を行う

先にも述べたように OS においてはそれぞれの機能が何らかの形で協調して動作する場が多い。一般的に計算機科学を学ぶ学生にとって重要なことは、基本的な概念の理解と

深度の深い実践的な学習が必要である。これらは相反することのように思えるが、概念を理解していなければ実践的な学習を行っても全体のイメージが付かずシステム全体を見渡す力に欠けてしまうことが多い。

これらの理由から本教育支援システムでは OS の機構、動作の概念に特に重点をおいて教育支援を行うものとする。

(2) 学習者の理解度に応じた段階的な教育支援を行う

学習においては学習者の理解度に応じて段階的な学習が望ましい。そこで本教育支援システムでは学習する項目に対して、可視化の粒度を変化させることで、学習者が段階的に学習できるようにする。具体的にはまず OS の基本機能を定義し、これらの理解、学習が終わった段階で拡張機能としてさらに OS の機構を実装段階まで追っていく。

(3) 本教育支援環境における OS の教育項目

ここで本教育支援環境における教育のねらいについて述べる。

前述したように本教育支援環境では OS の基本概念である、資源の仮想化、管理、運用に重点をおいて教育支援を行う。なぜならば、OS 学習を含むさまざまな分野の学習において、その概念の学習、概念を教えることが非常に重要であるからである。たとえ実践レベルの深い知識を持っていても、その概念を理解していなければ、全体のイメージがつかめない。特にシステムプログラミングの分野においては、ハードウェア、ソフトウェアのあらゆる知識を持っていなければならない。それらを総括して効率のよいシステムを構築しなければならない。これらの知識を全体のイメージとして捉えるにはシステムプログラミングの概念を理解していなければならない。つまり OS に置き換えると、OS の概念こそがこれに相当すると考える。

ここで学習者に教育支援を行う際の、OS の概念、基本機能の定義について述べる。本教育支援環境においては次に示す要素を OS の基本機能として本教育支援環境の学習要素と定める。

表1 . OS 機能の概念の定義

学習レベル ベース コンポーネント	基本概念セット	拡張概念セット
プロセス管理	プロセスの定義、多重プログラミング、プロセススケジューリング、同期、排他制御	コンテキストスイッチ、プロセス制御ブロック、セマフォ、デッドロック
メモリ管理	スワッピング、仮想記憶、ページング、ページ置き換えアルゴリズム、セグメンテーション	断片化、ワーキングセット
ファイルシステム	ファイルの概念、ディレクトリ、ファイル構造	ディスクスケジューリング、ディスク構造
入出力管理	入出力ハードウェア、割り込み	ネットワーク
OS 構成法	OS アーキテクチャ、モノシリック、マイクロカーネル、ポリシとメカニズムの分離	分散システム、リアルタイム OS

- (1) プロセス管理
- (2) メモリ管理
- (3) ファイルシステム
- (4) 入出力管理
- (5) OS 構成法

これらを本教育支援環境における OS の基本機能と定める。特にここでは基本概念セットと呼ぶ。さらに OS 学習における基本概念セット以外に拡張概念セットとしてさらに詳細な機能を可視化する。

学習項目を基本概念セットと拡張概念セットに分けた理由は、学習者に対して一度に多くの学習項目を表示することで、学習の要点が隠れてしまい、学習者の混乱を招く可能性を排除することにある。基本概念セットおよび拡張概念セットの詳細は表1の通りである。

これらは J97 [5]や代表的な OS 参考書 [6][7][8]などを参考に定義した。

6 . 設計

6 . 1 設計方針

本教育支援システムでは次の設計方針を定める。

(1) 可視化部分を可視化モジュールとしてそれぞれ独立させる

OS には様々な機能がある。OS の可視化を行う際に、膨大にある OS の機能を一度にすべて行っていたのでは可視化システム全体が複雑になり、新しい機能、可視化要素の追加や修正が困難になってしまう。そこで、OS の各機能をそれぞれ独立させて可視化を行うこととする。具体的にはプロセス管理、メモリ管理、ファイルシステムといった大きな機能ごとを可視化のセット、つまり可視化モジュールとしてそれぞれを独立させる。さらに、それぞれの可視化モジュールにおいて可視化インタフェースを定義しておく。これにより新たな可視化要素の追加や改良ができるようになり、システム全体の自由度を向上させる。

(2) 可視化インタフェースを定義する

前項でそれぞれの可視化モジュールを独立させることで新たな可視化要素の追加が可能になると述べた。しかし実際にはそれだけでは自由度を向上させたことにはならない。そこで可視化インタフェースを定義することでデータの流れを明確にし、新たな可視化コンポーネントの追加や変更を可能にする。具体的にはプロセス管理であれば PCB (Process Control Block) を用いることや、メモリ管理

であればメモリテーブルの定義などである。

6.2 全体構成

ここまでで述べたように本教育支援環境では概念に重点をおいた、段階的な教育支援を行うことを方針とする。次に全体の構成を示す(図1)。本教育支援環境は可視化モジュールとそれらをコントロールするユーザインタフェースから構築されている。さらに、可視化モジュールはベースコンポーネントと可視化コンポーネントから構成されている。

6.3 可視化モジュールの設計

可視化モジュールは母体となるベースコンポーネントとベースコンポーネント上に実装される基本セットの可視化コンポーネントからなる。ベースコンポーネントは可視化に共通するインタフェースと提供し、可視化コンポーネントはそのインタフェースをもとに可視化を行い、出力する。可視化コンポーネントによって状態が変更されたときは、ベースコンポーネントの共通インタフェースを修正する。

(1) ベースコンポーネントの設計

ベースコンポーネントは OS 機能における大項目、つまりプロセス管理、メモリ管理、ファイルシステムなどの大きな基本機能をコンポーネントとして抽象化したものであり、OS における大きな機能の可視化を行う際の母体となるコンポーネントである。例えばプロセス管理の場合であれば、プロセス管理における様々な機能、スケジューリングやディスパッチ、同期、排他制御などの可視化を行う際の共通できる情報を提供する。

(2) 基本セットの可視化コンポーネントの設計

ここで述べる基本セットの可視化コンポーネントとは表1における基本セットの各機能である。つまり、ベースコンポーネントにあたる各大項目における各機能である小項目である。

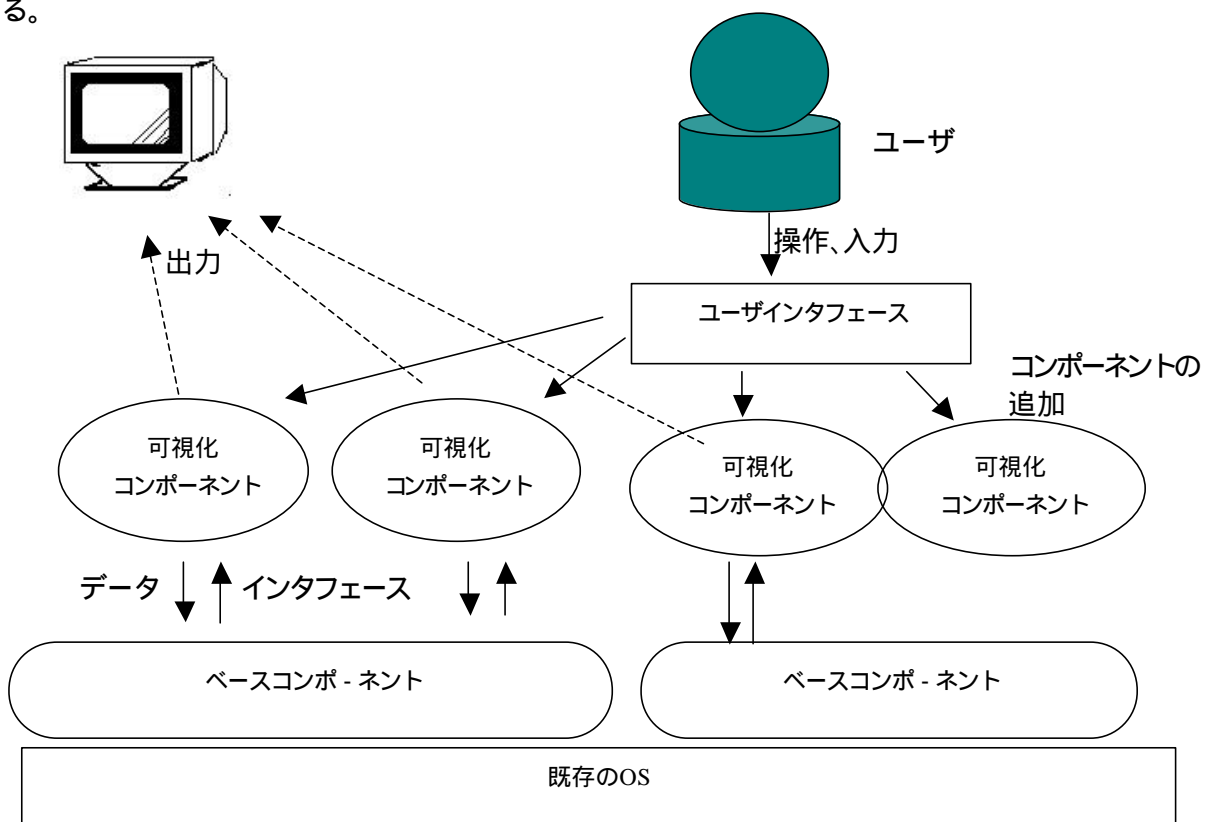


図1. 全体構成図

これらはそれぞれベースコンポーネント上に実装され、ベースコンポーネントの情報をもとにOSの各機能の可視化を行う。

可視化コンポーネントにおいては実際にユーザに可視化を出力する重要な部分である。次に本教育支援環境における可視化を行う際の可視化の特徴を示す。

) アニメーションを用いた表示

動作をアニメーションによって表現することで状態の遷移を明確にすることができる

) 色表現の工夫

可視化する際の配色を工夫することによって一目でその状態、意味を理解することができる。プロセス管理にたとえると、実行しているプロセスを緑、実行待ちしているプロセスを黄色、停止しているプロセスを赤というように信号色を使うことで各プロセスの実行状態を一目で知ることができるようになる。

) 動作スピードの変更

学習者は個人によって理解度が違うので、ゆっくり進めたい、速く飛ばしたいという要望がある。そこでアニメーションにおける動作のスピードを調節できるようにする。

6.4 ユーザインタフェースの設計

ここでは、ユーザインタフェースの設計について述べる。ユーザインタフェースではユーザが可視化モジュールの操作や入力するインタフェースとしての機能を提供する。主な機能は次の通りである。

- ・ 可視化コンポーネントの表示、選択
- ・ 可視化のスピード調整
- ・ 各パラメータの入力

7. 実現

本教育支援環境ではすでにプロセス管理におけるプロセススケジューリング、同期、排他制御、ディスパッチ、メモリ管理におけるページングの例が可視化されている。(図2、3、4、5、6)

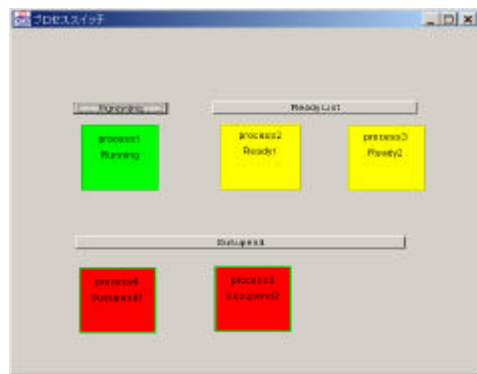


図2. プロセススケジューリングの可視化

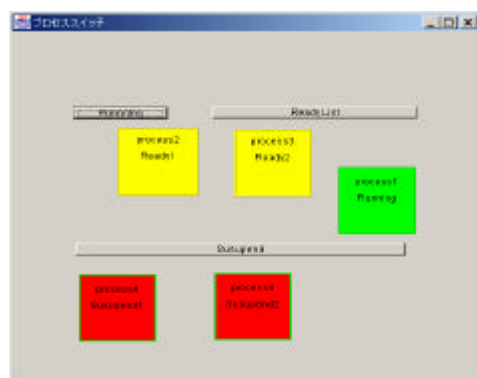


図3. プロセススケジューリングの可視化

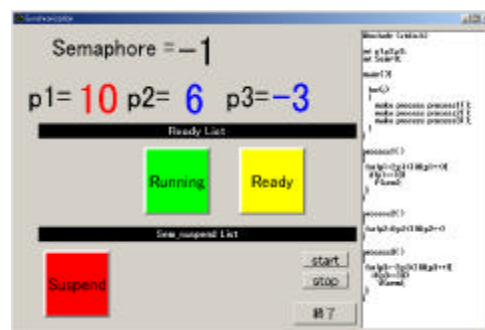


図4. 同期の可視化

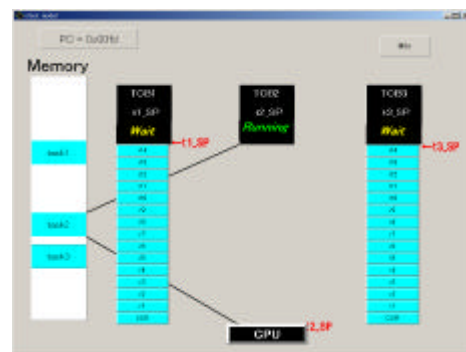


図5. ディスパッチの可視化

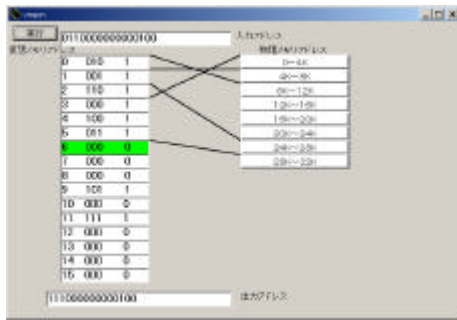


図 6 . ページングの可視化

図 2、3 はプロセススイッチの様子を可視化したのものである。ここではラウンドロビンアルゴリズムによってプロセスがスケジューリングされている様子をあらわしている。色のついた箱はそれぞれプロセスを表している。これによりマルチプログラミングの概念や、CPU の仮想化の概念を学習することができる。

8 . 評価

8 . 1 アンケート結果

実現した可視化の例を本学情報工学科における 3 年生の OS の講義においてプロセススケジューリング、同期、排他制御、ページングの例についてデモンストレーションを行い、アンケートをとった。対象となった学生は 3 年次に OS についての概念を知った学生であり、各要素についての講義を行った後に復習する意味でデモンストレーションを行ったものである。

質問内容は次の通りである。

- (1) 可視化の様子はわかりやすかったか (5 段階評価、5 が最良)
- (2) 可視化は見やすかったか (5 段階評価、5 が最良)
- (3) 良かった点はどこか
- (4) 悪かった点、見にくかった点はどこか
- (5) わかりにくかった点について、どのように改良すればよいと思ったか

約 50 人の学生にこれらの質問内容への回答が得られた。次に (1) (2) の質問の回答を表 2 に、(3) (4) (5) の回答のまとめを示す。

表 2 . アンケート結果

評価	← 高い → 低い				
	5	4	3	2	1
わかりやすさ	14 人	20 人	11 人	3 人	0 人
見やすさ	5 人	20 人	15 人	7 人	1 人

(3) 良かった点はどこか

- ・ 実際に目で見えるところ
- ・ 文章だけではなく実際に動きがあるところ
- ・ それぞれの状態によって色分けされているところ
- ・ どのように動いているのかゆっくりと実行される場所
- ・ セマフォがある場合とない場合の動きの差が明確になっていたところ

(4) 悪かった点、見にくかった点はどこか

- ・ 時間の経過がわかりにくいところ
- ・ 解説がないと何が起きているのかわかりにくいところ
- ・ 字が多いところ

(5) わかりにくかった点について、どのように改良すればよいと思ったか

- ・ ひとつ前の状態に戻るアンドゥ機能をつければ良いと思う
- ・ 同期や排他についてもっと違う例で示せば良いと思う
- ・ 説明しなくてもぱっと見ただけでわかるようにして欲しいと思った

これらの結果が得られた。

8 . 2 考察

前述したアンケート結果からこれらの可視化についての考察を行う。

(1) わかりやすさ、見やすさについての考察

わかりやすさ、見やすさについてはほぼ満足できる結果となった。しかしながらわかりやすさに比べ見やすさの評価が低かった点に着目すると、配色の配慮や字の大きさなどに変更の余地があると考えられる。

(2) 良かった点についての考察

良かった点についてはねらいどおりの結果を得ることができた。アンケート回答を見てもわかるように講義においてテキストや図の説明だけではなく、実際に動くものを併用することで理解の促進を得ることができると考えられる。

(3) 悪かった点、見にくかった点についての考察

本可視化環境ではアニメーションによる可視化を行っているアニメーションという手法は状態の遷移を明確にできるという大きな利点があるが、状態が遷移する直前までの情報は表すことができない。これらの問題については逆再生ができる、もしくは残像効果の方法で解決できるのではないかと考える。

また、解説がないと何が起きているのかわからないという回答に対しては、本教育支援環境はまず、講師が講義のサポートツールとしての使用を想定しているので問題はないと考える。ただし、学習者が独習する場合の教材としての使用には問題がある。この問題点については、可視化における OS 機能の抽象化の仕方をもう少し深く考える必要がある。

(4) どのように改良すればよいかという回答の考察

どのように改良すればよいかという回答では様々な意見を得ることができた。特にアンドゥ機能をつければよいという回答は非常に良い提案であると考えられる。

9. おわりに

本報告では計算機科学、特に OS の基本機能を学ぶ学生が、OS の概念や動作、管理機構の学習時において、可視化によって学習を支援する環境の実現および評価について述べた。この教育支援環境によって、OS の基本的な機

能を容易に学習でき、より深く理解することが可能となった。

今後は考察における問題点の解決および、可視化コンポーネントの追加を行っていく。

参考文献

- [1]西野洋介、早川栄一、高橋延匡：可視化による OS 基本機能の学習支援システムの開発、情報処理学会研究報告、2000-OS-84、pp.173-180、2000 .
- [2]A.S.Tanenbaum：MINIX オペレーティングシステム、1989年、アスキー出版局
- [3]W.Christopher, et al：The Nachos Instructional Operating System, <http://http.cs.berkeley.edu/~tea/nachos/nachos.ps>
- [4]Dining Philosophers http://www.cs.bham.ac.uk/teaching/examples/simjava/dining-philosophers/dining_philosophers.html
- [5] 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** <http://www.ipsj.or.jp/katsudou/chosa/J97-v1.1.pdf>
- [6] A.S.タネンバウム:OSの基礎と応用株式会社トッパン:1998年
- [7] 前川守:オペレーティングシステム 岩波書店 1988年
- [8] J.L.ピーターソン、A.シルバーシャッツ:オペレーティングシステムの概念培風館 1988年