

学習支援環境「港」における Nachos を用いた OS 可視化ツールの開発

本橋 大樹[†] 西野 洋介[‡] 早川 栄一[†]

拓殖大学工学部情報工学科[†] 東京都立府中工業高等学校[‡]

1. 研究の背景と目的

現在、拓殖大学工学部情報工学科早川研究室で行われている研究としてオペレーティングシステム（以下 OS）の学習支援環境「港」プロジェクトというものが行われている。このプロジェクトはスケジューリングといった OS の基本的な動作の学習に用いるために開発されているもので、対象としている OS は C 言語などで記述されたものである。

「港」プロジェクトの OS 教材は C 言語のものではなくオブジェクト指向ベースの OS 教材やそれに対応した可視化環境も必要性がでてきた。

本研究ではオブジェクト指向学習用 OS として実際に学習用 OS 教材として多く用いられている Nachos[1]を採用し、これを用いて「港」プロジェクトにあるような可視化環境と連動して実際の OS の動作を学習できるシステムの開発を行う。

2. 問題分析

(1) 学習用 OS の問題

学習用 OS への要求としてコードの読みやすさがある。学習用 OS である Nachos はオブジェクト指向で記述されており、コードの見通しがよくなっている。

さらに、Nachos は多くの大学などで学習用教材として用いられている。しかし、「港」プロジェクトの可視化環境は Nachos のような OS には対応していないため有効利用することができない。Nachos に対応することで、この可視化環境を有効利用できる機会が増えると考えられる。

(2) オブジェクト指向 OS における「港」プロジェクトへの要求

Nachos はオブジェクト指向言語である C++言語で記述されていて、可読性が上がっている。Nachos のような OS の可視化を行うには、オブジェクトのふるまいを可視化することが重要である。

従来の「港」プロジェクトの可視化環境ではこれらに対するサポートがない。そこで「港」プロジェクトに求められるものとしては、オブジェ

クト間のアクセスの様子を可視化するというものである。これによって学習者は OS の動作を理解することを容易にすることができる。

3. 特徴

本研究で行う可視化の特徴は次のとおりである。

- (1) オブジェクトという単位で可視化を行うことで、オブジェクトのアクセスや依存関係などといったことを知ることができる。
- (2) 可視化ツールで、動作ログを読み込むことによりオブジェクトのアクセスの関係やタスクの状態変化の可視化を行うこと。
- (3) 一般的に学習用 OS として用いられている Nachos の可視化を行うことで、学習者が理解しやすくなる。

4. 設計

4.1 全体構成

図 1 に、全体の構成図を示す。Nachos 中の nachos/codes/threads にあるマルチスレッドシステムについて可視化を行った。OS を実行した際にスレッド制御やスケジューラといったオブジェクトへのアクセスの様子や、タスクの状態変化、コンソール画面に出力された実行結果を動作ログとして出力させる。その動作ログを可視化ツールで読み込み、ツール内の可視化部分呼び出すことで OS の動作の可視化を行う。

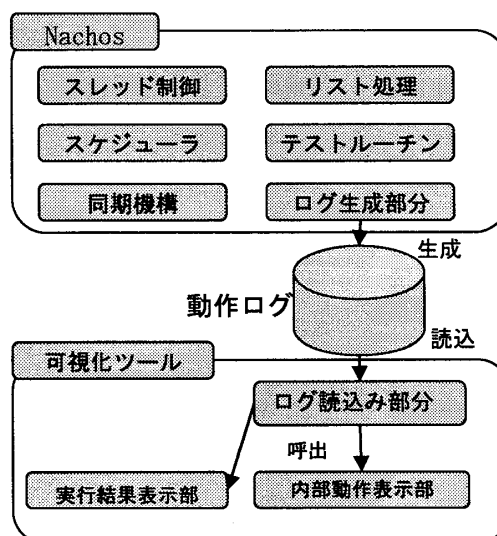


図 1 全体構成図

Development of on Nachos based Operating System Visualization Tool on the “Minato”

[†] Daiki Motohashi, Eiichi Hayakawa · Faculty of Engineering, Takushoku University

[‡] Yousuke Nishino · Tokyo Metropolitan Fuchuu technical high school

4.2 OSの可視化

OS部分で実装するものは次のとおりである。

- タスクの状態変化やオブジェクト間のアクセスの様子、コンソール画面でのOSの実行結果などを動作ログとして出力させる部分を実装する。
- 「港」プロジェクトの可視化環境にNachosに対応させるために、それに合わせた動作ログを出力する部分を実装する。
- 動作ログをテキストファイル形式にすることで可視化ツールのような異なる実行環境でも利用できる。

これらのことにより生成された動作ログを、可視化ツールで読み込むことで可視化を行うことができる。

生成したログによって行える可視化を図2に示す。これは、「港」プロジェクトの可視化環境でNachosの可視化を行った画面である。図2では、タスクの状態遷移を時間軸上で表したもので、タスクが実行されている間は緑色で表している。黄色で表されている部分は、タスクが実行されていないことを示している。

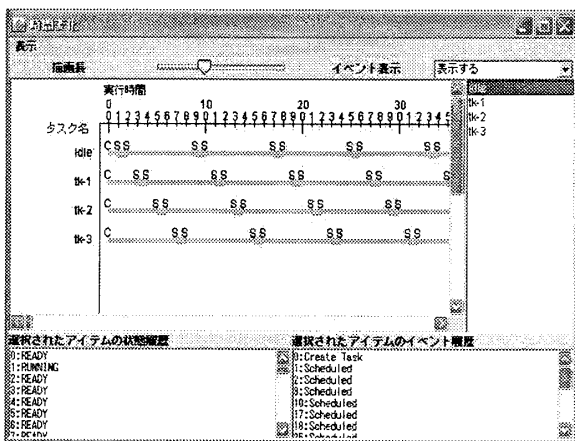


図2 時間変化によるタスクの状態遷移の可視化

4.3 可視化ツールの設計

「港」プロジェクトの可視化環境を元に、Nachosを実行させたときに出力される動作ログを読み込み、読み込んだデータを元に可視化を行う。

読み込むログファイルには、タスクの状態変化、オブジェクト間のアクセスの様子、コンソール画面に出力される動作結果といった情報がデータとして格納されている。

動作ログを読み込む部分では、読み込んだデータから実行結果の出力かOS内部の動作なのかを判断し、それぞれの可視化部分呼び出して可視化を行う。

可視化を行う部分は、OS内部の動作の可視化を行う部分と、OSを実行させたときの実行結果を

表示させる部分の二つに分けられる。内部の動作の可視化として、オブジェクトへのアクセスの様子やタスクの状態変化の可視化を行う。OSの実行結果とは、OSを実行させた際に、コンソール画面に出力されるものである。

この実行結果とOS内部の動作の可視化を同時に表示させることにより実行結果の出力に対して、OS内部の動作をオブジェクトのアクセスの様子を可視化することで表すことができる。

図3に、可視化ツールの実行画面のイメージを示す。上の部分には、コンソール画面に表示されるOSの実行結果、下の部分でOS内部の動作としてオブジェクトのアクセスの様子を可視化している。アクセスされているオブジェクトを矢印で結び、呼出し関係を表している。

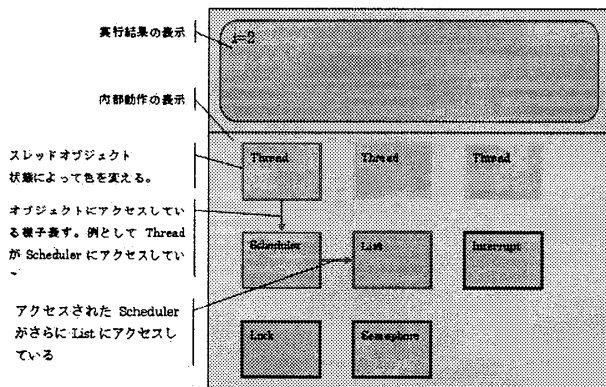


図3 可視化ツールの表示イメージ

5. おわりに

本研究では、学習用OSとして広く用いられているNachosを用いて、OSの動作を可視化する可視化ツールの開発を行った。また、「港」プロジェクトの可視化環境をNachosに対応させるために動作ログを生成し、Nachosにおける動作の可視化を行えるようにした。これにより、Nachosにおいても「港」プロジェクトの環境で学習することが可能となった。

今後の課題としては、現時点ではオブジェクトという単位でだけ可視化を行ったが、今後はオブジェクト内のメソッドまで範囲を広げて呼び出し関係の可視化を行えるツールの開発を行っていく。

参考文献

- [1] Nachos, <http://www.cs.washington.edu/homes/tom/nachos/>
- [2] Qing Li, Caroline Yao, リアルタイム組込みOS, 翔泳社, 2005
- [3] 永井正武, 実用組込みOS構築技法, 共立出版, 2001