

|              |                                                                                 |
|--------------|---------------------------------------------------------------------------------|
| Title        | オペレーティングシステムの最適化に関する研究                                                          |
| Author(s)    | 寺田, 徹                                                                           |
| Citation     |                                                                                 |
| Issue Date   | 1997-03                                                                         |
| Type         | Thesis or Dissertation                                                          |
| Text version | author                                                                          |
| URL          | <a href="http://hdl.handle.net/10119/1068">http://hdl.handle.net/10119/1068</a> |
| Rights       |                                                                                 |
| Description  | Supervisor:中島 達夫, 情報科学研究科, 修士                                                   |

# オペレーティングシステムの最適化に関する研究

寺田 徹

北陸先端科学技術大学院大学 情報科学研究科

1997年2月14日

**キーワード：** 拡張可能オペレーティングシステム、動的コード生成、モジュール.

オペレーティングシステムの研究あるいは実用の歴史は常にその機能を追加し続け、オペレーティングシステムは大きく複雑になった。しかしいかに複雑になろうと、あらゆるアプリケーションにとって十分な機能と十分なパフォーマンスを提供できるオペレーティングシステムは存在しない。それは個々のアプリケーションによって要求が大きく異なるためであり、あらゆるアプリケーションの要求を満たすオペレーティングシステムは今後とも現れないであろう。

そこで近年、オペレーティングシステムの研究は単なる機能追加とは異なる方向を目指すようになった。それは、アプリケーションによって拡張することができるオペレーティングシステムであり、その中でマイクロカーネルが登場した。マイクロカーネルはそれ以前のモノリシックなオペレーティングシステムから一部の機能を分割して、ユーザレベルのサーバとして実現した。このことによって、システムのリブートを伴わない動的な拡張、異なるサーバを使うことで異なるサービスが受けられる柔軟性、サーバの障害がそのサーバを利用しないアプリケーションには影響を与えないという安全性が実現した。

しかしマイクロカーネルをベースとするシステムは、カーネル/ユーザ境界を越えるデータや制御の移動の頻度が従来のモノリシックカーネルと比較すると大きく、パフォーマンスに問題がある。これを改善するための研究が行われている中で、アプリケーションのコードの一部を動的にカーネルへダウンロードすることによってカーネル/ユーザ境界の移動を削減するアプローチが有力である。カーネルへダウンロードする拡張コードは他のアプリケーションに影響を与える危険のあるエラーを含んではならない。しかし、拡張コードのエラーを完全にチェックするメカニズムは現在まで存在しないので、カーネルへのダウンロードコードを含むアプリケーション設計者は、意味的なエラーを含まないように注意しなければならない。

本研究は、拡張コードにエラーを含まないためには、簡略で汎用的なプログラムを組み合わせるモジュール化設計による設計が良いと考える。しかし、モジュール化設計による

プログラムは、単一の目的に特化して設計したものと比較すると効率の面で劣る。そこで本研究では、モジュール化設計の効率に関する欠点を改善する最適化を提案する。

モジュール化設計のプログラムの実行効率が悪い原因の一つは、それぞれ独立して設計した複数のモジュールが、同一のデータ領域に対しての処理を含むことである。データのレジスタへのロード、何らかのデータ処理、メモリへのストアという通常の処理の流れの中で、同じデータを複数のモジュールでそれぞれロード/ストアすることが冗長である。そこで、複数モジュールにおける処理を単一のロード/ストア間にまとめる機構とインタフェースを提供する。

また別の原因は、モジュール集合がカーネル内イベントなどのモジュール外部の要因によって実行を行うときなどに存在する。モジュール集合の実行中、その実行状況によってはアプリケーション全体として意味のある状況ではないと判定して実行を終了する場合があるが、このときいくつかのモジュールの実行が終了するまでこの判定できないことがあり、その場合その実行は不要な処理を多く含む。またはその実行がアプリケーション全体として意味のある場合であっても、複数のモジュールで同じ判定を含むことは冗長である。そこで、ここにパケットフィルタのようなフィルタリングの機構を導入して、実行状況の判定を集約する。

これらの最適化の実現には動的コード生成 (Dynamic Code Generation) を利用する。動的コード生成は、まずモジュール集合をひとつの関数にまとめることによってプロシージャコールのコストを削減する。また処理の順序を動的に決定するので、データ処理を統合してメモリ/レジスタ間のロード/ストアを削減することを可能にする。そしてフィルタ構造をマシンコードへ変換することによって、実行時の処理を削減する。

動的コード生成には MIT で開発された VCODE システムを使い、提案する最適化を Real-Time Mach マイクロカーネルに実現する。最適化の有効性をアプリケーションがカーネルへ拡張コードをダウンロードすることの有効性とあわせて検証する。