3L-5

# マルチコアCPUにおけるOSの資源管理方式の研究

磯部 泰徳† 佐藤 未来子‡ 並木 美太郎§
東京農工大学大学院工学府情報工学専攻 †/ 東京農工大学大学院 ‡/ 東京農工大学大学院共生科学技術研究院 §

## 1 はじめに

近年，マルチコアプロセッサはゲームなどの様々な機器に搭載されている [1]．マルチコアプロセッサの性能を引き出すためには，システムソフトフェアの支援が必要不可欠である．

本研究では，共有メモリ型マルチコアプロセッサを対象として，その性能を活かすためのプログラム実行基盤の実現を目指している．ホモジニアスなマルチコアプロセッサに対しては，専用 OS を利用したコアの仮想化を行い，汎用 OS に対してマルチスレッド実行環境として提供する．また，動的再構成可能プロセッサ対しては，汎用 OS 用のドライバを用いて実行管理を行うことで，汎用 OS 上で実行するプログラム上から動的再構成可能プロセッサを活用することを可能とする．

## 2 本研究の目標

本研究では，汎用 OS とマルチコアに特化した専用 OS を連携動作させる機構と汎用 OS から動的再構成可能プロセッサを制御する機構を提案する．

前者はマルチコアプロセッサ上の複数のコアを専用 OS で仮想化し，それらを効率的に利用できるマルチスレッドプログラムの実行環境を汎用 OS に対して提供する．また，専用 OS 単体では，汎用 OS におけるファイルシステムや I/O などの機能は期待することが難しいため，双方の OS を連携させることにより，専用 OS に対して汎用 OS の管理する資源を利用可能とする．

これにより，ユーザは利便性の高い汎用 OS から高効率な専用 OS のプログラム実行環境を利用できるようになるとともに，専用 OS 用プログラムから汎用 OS の管理する資源が利用できるようになり，より自由度の高いプログラミングを行うことができる．

後者は汎用 OS に対して動的再構成可能プロセッサを適切に仮想化し，汎用 OS 上のプログラムから動的再構成可能プロセッサを利用するためのインターフェースを提供する．

---

A Study on Resource Management System on Operating System for Multi-core Processor

† Hironori Isobe
Computer and Information Sciences, The Graduate School at Tokyo University of Agriculture and Technology
‡ Mikiko Sato
Graduate School of Engineering, Tokyo University of Agriculture and Technology
§ Mitaro Namiki
Institute of Symbiotic Science and Technology, The Graduate School at Tokyo University of Agriculture and Technology

これによって，従来はレジスタなどを直接プログラムから変更して利用されることが多かった動的再構成可能プロセッサを汎用 OS のインターフェースで扱うことができるようになり，より容易に動的再構成可能プロセッサを活用したプログラムを開発できるようになる．

## 3 システム構成

本システムは，OS 連携機構と動的再構成可能プロセッサ制御機構の 2 つからなる．図 1 にシステムの全体の概要を示す．

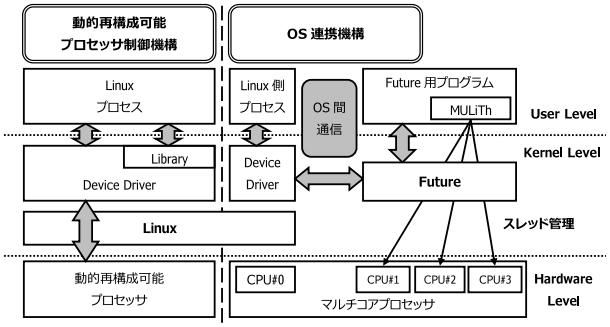


図 1: システムの概要

OS 連携機構ではマルチコアの異なるコア上で汎用 OS と専用 OS を同時に動作し，連携することで一つのシステムとして動作する．汎用 OS である Linux がファイルシステムや I/O，ネットワークなどの機能を提供し，専用 OS である Future[2, 3] は Linux に対し，マルチスレッドプログラム実行環境を提供する．

動的再構成可能プロセッサ制御機構では，Linux のデバイスドライバを用いて動的再構成可能プロセッサを仮想化する．プロセッサ構成情報の変更や，ローカルメモリへの入出力といった操作に対し，適切なシステムコール I/F を提供することで，動的再構成可能プロセッサを用いた SIMD 演算環境を Linux のユーザープログラムから利用可能にする．

## 4 連携機構

Linux と Future は連携して一つのシステムとして動作する．双方の OS は CPU コア間の割込による OS 間通信で連携し，データの受け渡しは共有メモリを用いることでメモリコピーのオーバーヘッドの発生を避ける．OS 間連携機構では，次の機能を提供する．

**Linux からの Future の起動**

あらかじめ起動した Linux から Future のカーネルをロードし，他の CPU コア上でブートする．

**Future 用プログラムのロード及び実行**

Linux のファイルシステム上で管理している Future 用プログラムを Linux からロードし，Future 上で実行させる．

**Future 用プログラムのシステムコール代行処理**

Future 用プログラム中で発行されたシステムコールを Linux で代行することで，Linux 側で管理している I/O などの資源を利用可能にする．

## 5 動的再構成可能プロセッサ制御

動的再構成可能プロセッサは設定したコンフィグレーションによってその処理内容が変わる．演算処理を行う際は，処理内容に合わせたコンフィグレーションを予め指定し，処理の対象となるデータをプロセッサのローカルメモリに転送してから演算処理を開始しなければならない．動的再構成可能プロセッサ利用の手順を図 2 に示す．

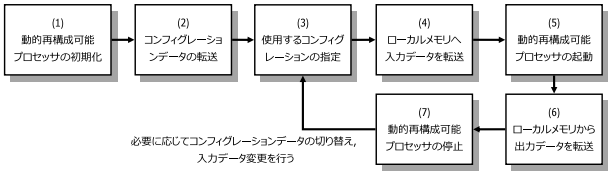


図 2: 動的再構成可能プロセッサ利用の手順

この演算処理を行うための手順を従来は I/O レジスタを直接操作することによって行うことが多かった．これらの操作をデバイスドライバのシステムコールとして仮想化することによって OS からの管理を容易にする．

しかし，動的再構成可能プロセッサはその構成の自由度の高さから数多くのレジスタを備えている場合が多く，その全てをデバイスドライバのシステムコールとしてサポートするのは現実的であると言えない．

このため，デバイスドライバでは，ローカルメモリへの入出力と I/O レジスタの入出力のみをサポートし，実際の制御はライブラリによって提供することで処理内容に応じた制御に柔軟に対応することができる．

図 3 に動的再構成可能プロセッサ制御の概要，表 1 にデバイスドライバでサポートするシステムコール，表 2 に制御ライブラリ関数の例を示す．

表 1: 対応するシステムコール

| ハンドラ名 | 動作内容 |
|---|---|
| read | ローカルメモリから読込 |
| write | ローカルメモリへ書込 |
| ioctl(PUT_REG) | 指定した I/O レジスタの書込 |
| ioctl(GET_REG) | 指定した I/O レジスタの読出 |

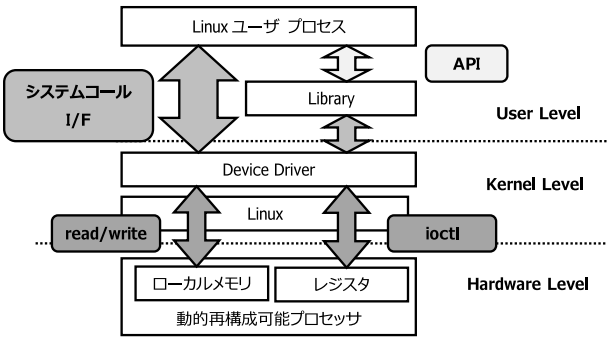


図 3: 動的再構成可能プロセッサ制御の概要

表 2: 制御ライブラリ関数の例

| 関数名 | 動作内容 |
|---|---|
| set_config() | コンフィグレーションの指定 |
| exec_process() | 処理の実行 |
| stop_process() | 処理の中断 |
| reset_processor() | プロセッサの初期化 |

これらのシステムコールと API を用いることで Linux プロセス上から動的再構成可能プロセッサによる演算処理を利用することが可能となる．

## 6 おわりに

本稿では，マルチコア CPU における OS の資源管理方式について述べた．今後はこれらの機能を実装し，評価をしていく必要がある．

## 参考文献

[1] S. Torii et al., "A 600MIPS 120 mW 70 $\mu$A Leakage Triple-CPU Mobile Application Processor Chip," Proc. 2005 IEEE Int'l Solid-State Circuits Conf., Digest of Technical Papers (ISSCC 05), IEEE Press, 2005, pp. 136-137.

[2] 佐藤未来子，磯部泰徳，十山圭介，野尻徹，入江直彦，内山邦夫，並木美太郎．汎用ホモジニアスマルチコアプロセッサにおける OS とスレッドライブラリ．情報処理学会第 20 回コンピュータシステムシンポジウム, ポスターセッション,(2008)

[3] 佐藤未来子，磯部泰徳，並木美太郎．マルチコアプロセッサにおける OS による MMU を用いたスクラッチパッドメモリの管理方式．コンピュータシステム・シンポジウム (2009)

[4] 下澤拓，藤田肇，石川裕．マルチコア SH における複数カーネルの実行機構の設計と実装．情報処理学会研究報告,2008-OS-109,pp.25-32(2008)

[5] 遠藤幸典，菅井尚人，山口義一，近藤弘郁．シングルチップマルチプロセッサ上のハイブリッド OS 環境の実現 - システムアーキテクチャ -．情報処理学会第 66 回全国大会 (2004)