

Title	タスクの静的優先度に従ったバイナリの構築法に関する研究
Author(s)	笹山, 高志
Citation	
Issue Date	2009-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8131
Rights	
Description	Supervisor: 田中清史, 情報科学研究科, 修士

タスクの静的優先度に従ったバイナリの構築法に関する研究

笹山 高志 (0610040)

北陸先端科学技術大学院大学 情報科学研究科

2009 年 2 月 5 日

キーワード: ARM プロセッサ, μ ITRON4.0 仕様 OS, 静的優先度, コンフィギュレータ.

1 はじめに

近年の組み込みシステムでは, 厳しいコストの制約や, ハードウェアの物理的な大きさの制限により, 搭載できるメモリ容量が限られている. そのため, コードサイズが小さく, 使用メモリ容量を節約できるシステムが望まれている. 組み込み用途向けのプロセッサにはコードサイズを削減することを目的に異なる命令長を備えているものがある. その代表的なプロセッサに ARM[1] がある. ARM プロセッサは近年, 組み込みシステムなどで広く利用されている 32 ビット RISC プロセッサであり, 通常の 32 ビット ARM モードとコード圧縮を優先した 16 ビット Thumb モードの 2 つを備える.

本研究では, ARM プロセッサ上で動作する μ ITRON4.0 仕様 OS[2] を実装し, OS の上で動作するタスクの静的優先度に従って, 自動でタスクと, タスクで使用されるサービスコールを ARM モード, Thumb モードと切替える. そしてリアルタイム性[3], コードサイズに最適化された実行バイナリを, 開発者の負担を減らしつつ構築することを目的とする.

2 ARM と μ ITRON4.0 仕様 OS

組み込みシステムなどで広く利用されている 32 ビット RISC プロセッサの ARM は, 32 ビット ARM モードと, 16 ビット Thumb モードの 2 つを備える. Thumb モードは ARM モードに比べて, 実行命令数増加により実行速度が遅いというデメリットがあるが, 総コードサイズが ARM モードに比べて圧縮できるというメリットがある. この 2 つのモードは 1 つの実行バイナリの中で混在することが可能である.

また, 近年の組み込みシステムの開発において, 組み込み用 OS が使用される機会が多くなっている. これは OS の導入によりリアルタイム性の向上や, アプリケーション開発の負担が軽減される等の利点があるためである. しかし, 組み込み用 OS では汎用 OS に比べ

て、よりリアルタイム性が求められる傾向があり、デッドラインミス、平均応答時間などの情報が重要になる。本研究では、組み込み用 OS の 1 つである、 μ ITRON4.0 仕様 OS を用いる。 μ ITRON4.0 仕様 OS では、タスクに設定されている静的優先度に従いスケジューリングを行う方式がとられている。

3 提案手法

本研究では、ARM プロセッサ上で動作する、 μ ITRON4.0 仕様アプリケーションをターゲットとした、実行バイナリの最適化システムを提案し、実行バイナリのリアルタイム性を評価可能なシミュレータを実装する。実行バイナリの最適化システムは、OS 上で動作するタスクの静的優先度をもとに、自動でタスクと、タスクで使用される OS のサービスコールを ARM モード、Thumb モードと切替え、プログラマが手動で行う実行バイナリの最適化処理を自動化し、開発の負担を軽減しつつ最適化された実行バイナリを得ることを目的としている。実行バイナリの最適化システムのアルゴリズムは、ある一定の閾値をプログラマが設定し、その閾値と静的優先度を比較して、優先度が高い場合は ARM モード、低い場合は Thumb モードと切替える。シミュレータは、タスクの平均応答時間、デッドラインミス情報などを計測可能とし、実行バイナリのリアルタイム性を考慮しながらアプリケーション開発を支援することを目的としている。なおシミュレータは、ARM プロセッサ (ARM7TDMI) 搭載機器をターゲットとしたものである。

4 評価

ARM プロセッサ上で動作する μ ITRON4.0 仕様アプリケーションをターゲットとした、実行バイナリの最適化システムを実装した。そして、その最適化システムを使用して、自動でタスクの静的優先度に従った実行バイナリを構築した。実行バイナリのリアルタイム性を評価するために、ARM シミュレータを実装した。評価の結果、提案手法により実行バイナリのサイズを削減できることが確認できた。そしてシミュレータによりタスクの平均応答時間、デッドラインミス数などを計測することができた。

5 まとめ

本研究では、ARM プロセッサ上で動作する、 μ ITRON4.0 仕様 OS を実装した。そして、タスクの静的優先度に従って実行バイナリをコンフィギュレーションするツールを開発し、それをを用いることで、実行バイナリのサイズを削減できることを示した。また、実行バイナリのリアルタイム性を評価可能なシミュレータを実装した。

参考文献

- [1] ARM Limited.“ ARM Architecture Reference Manual ”,Pearson Education, 1996-2000.
- [2] トロン協会.“ μ ITRON4.0 仕様 Ver.4.03.00 ”.
- [3] G . C . Buttazzo .“ Hard Real-Time Computing Systems : Predictable Scheduling Algorithms And Applications ”Springer , 2004.