

Title	特定アプリケーションのためのRTOSの最適化に関する研究
Author(s)	LI, JIN
Citation	
Issue Date	2016-06
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/13654
Rights	
Description	Supervisor:田中 清史, 情報科学研究科, 修士

特定アプリケーションのための RTOS の最適化に関する研究

LI JIN(1310206)

北陸先端科学技術大学院大学 情報科学研究科

2016 年 5 月 19 日

キーワード: RTOS, システムコール, 最適化, 組込みシステム, エラーチェック, ロック.

● はじめに

現在、組込み開発分野における RTOS が広く利用される。RTOS を利用することにより、リアルタイム処理や資源管理の効率化が容易に実現できる。しかし、RTOS の機能は固定されており、開発対象のアプリケーションが使用しない機能を多く含むことになる。このことが、システムの実行オーバーヘッドや実行バイナリのサイズに影響を与える可能性がある。RTOS はあらゆる組込みシステムで使用されることを想定している。アプリケーションの実行時に起こり得る様々なエラーをチェックするコードを含む。例えば、不正な ID 番号の使用, 対象オブジェクトが不在, 不正なパラメータ値の使用などのエラーチェックを行う。これらのエラーチェックは、システム稼働時に行われるため、実行オーバーヘッドが発生することになる。一般的に組込みシステムでは、アプリケーションは限定される。従って、特定のアプリケーションでは、RTOS が提供する全てのエラーチェックが必要となるわけではない。このことから、アプリケーションに従って不必要なエラーチェックコードを削除することにより、実行オーバーヘッドを低減することと、バイナリコードのサイズを削減することが可能である。また、システムコールは複数のタスクから同時に呼び出されることがある。重要なカーネルデータの一貫性を保障するために、システムコール内でロック機構が多用される。しかし、特定のアプリケーションでは、カーネルデータの排他アクセスが必要ない場合がある。前述のエラーチェックと同様に、ロックが不要となる箇所についてはロックのためのコードを削除することが効果的である。本研究では、アプリケーションコードの解析、およびシステムコール内の不必要なコードの除去を自動的に行う手法を提案し、提案する方式を利用することにより、システムの開発者は容易に最適化された RTOS を利用できることが特色である。また、評価において提案方式の有効性を示すことにより、今後の組込みシステム開発分野における RTOS の最適化の重要性を示すことになる。

- **提案手法**

本研究では、アプリケーションのソースコードを解析することにより、利用するシステムコール内の不必要なエラーコードを検出する方法を検討し、具体的な検出方式を提案した。次に各エラーチェックの要・不要を判断した結果を define.h というファイルに書き込む。ヘッダーファイルに書き込まれた値により、システムコール内のエラーチェックコードを除去することができる。続いて、アプリケーションのソースコードを解析することにより、利用するシステムコール内の不必要なロックコードを検出する方法を検討し、具体的な検出方式を提案した。次に各ロックコードの要・不要を判断した結果を define2.h というファイルに書き込む。ヘッダーファイルに書き込まれた値により、システムコール内のロックコードを除去することができる。

- **評価**

シミュレーションにより、提案した方式を実装する。実装したシステムに作成したタスクセットを適用し、アプリケーションの実行性能 (バイナリコードのサイズおよび実行時間と実行命令数) を評価する。実験によって対象システムコール内の各エラーチェックコードの削除によってバイナリサイズは平均 16.88% 削除した。実行時間は平均 10.46% 削除した。実行命令数は平均 9.46% 削除した。対象システムコール内のロックコードの削除によってバイナリコードのサイズは平均 6.84% 削除した。実行時間は平均 28.51% 削除した。実行命令数は平均 35.21% 削除した。

- **まとめ**

本研究では、アプリケーションコードの解析、およびシステムコール内の不必要なコードの除去を自動的に行う手法を提案した。本手法を利用することにより、実行オーバヘッドとバイナリサイズが削除されることを評価により示した。アプリケーションタスク群のソースコードを解析することにより、利用するシステムコール内の不必要なエラーコードとロックコードを検出する方法を検討し、具体的な検出方式を提案した。エラーチェックは主にシステムコールのオペランドと記号定数の解析により必要性を決定する。また、各 RTOS 機能の利用がタスク間で競合するかどうかを解析し、ロック機能の必要性を決定する。実装するシステムはアプリケーションのソースコードを入力し、不必要なコードを除去したシステムコール関数を出力するものである。実装したシステムに作成したタスクセットを適用し、アプリケーションの実行性能を評価した。(本研究で使用するタスクセットのプログラムを C 言語で作成する。各タスクは μ ITRON4.0 のシステムコールによって RTOS の機能を利用するものとする。

参考文献

- [1] μ ITRON4.0仕様, (社) トロン協会 ITRON仕様検討グループ
- [2] The SPARC Architecture Manual Version 8.SPARC International Inc., Prentice Hall
,1992