複数レジスタセットを用いたタスク切り替えの高速化〜OSの開発〜

小山 俊之† 宮内 新† 荒井 秀一†
武蔵工業大学 ‡

1. はじめに

近年, 組み込みシステムの適応分野は急速に広がっていることにより, リアルタイム処理の性能に対する要求は高まっている. このような要求を満たすために, 組み込み分野においては, リアルタイムオペレーティングシステム (RTOS) が用いられる.

RTOS において, 重要な機能の一つとして高速な応答があり, 実現するためには, タスクの切り替えを高速化する必要がある. タスクの切り替えを高速化するためには, その中で行われるコンテキストの切り替えを高速化しなければならない. これは, 通常 RISC プロセッサなどはレジスタを多く保持するため, レジスタの退避, 復帰などに, より多くの時間がかかるためである. また, 割り込みに関してもコンテキスト切り替えは必要となる.

本研究室では, このようなタスクの切り替えを高速化するため, タスク, 割り込みハンドラごとにレジスタセットを保持し, タスク, 割り込みをハードウェアによって管理することで性能向上を図るプロセッサの開発が行われている. 図 1 にプロセッサのブロック図を示す.

タスク管理ユニットでは, 実行タスクを管理する優先度付き待ちキュー（レディキュー）のハードウェア化を行い, 次に実行するタスクの探索時間を削減する. レジスタ管理ユニットでは, 複数のレジスタセットをどのタスクに割り当てるかを決定する. また, ハードウェア化の対象 OS は ITRON 仕様 [1] の OS を採用する.

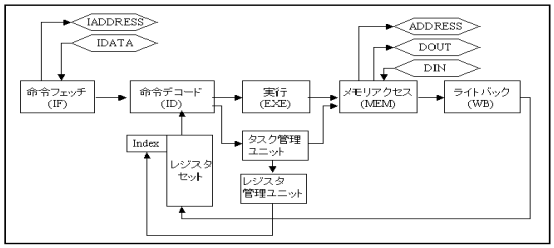

図 1: タスク切り替えに特化したプロセッサのブロック図

本研究目的は, タスク切り替え高速化プロセッサ上で動作する OS の開発を行い, ソフトウェアの視点からハードウェアの評価を行う.

2. レジスタ管理ユニットにおける割り当てアルゴリズム

レジスタ管理ユニットでは, 複数存在するレジスタセットの中からタスクを割り当てるレジスタセットを決定する. タスク数がレジスタセット数以下であれば, 全てのタスクがレジスタセットに割り当たるので, コンテキスト切り替えは, レジスタセットを切り替えるだけでよい. しかし, タスク数がレジスタセット数より大きい場合, レジスタセットに割り当てることができないタスクが存在する. このタスクの実行には, あるレジスタセットを選択し, そこに割り当てられているタスクのコンテキストをスタックに退避し, 次実行タスクのコンテキストを復帰する.

この割り当てアルゴリズムによって, 動作時の処理時間が異なる. よって, 処理時間やを考慮し, 最適なアルゴリズムを考慮する必要がある.

2.1 リアルタイムシステムへの適応

一般に周期の短いタスクは, 高い優先度を割り当てられる. そして, 高優先度のタスクの応答時間が短くなれば全体の処理時間は短縮される. つまり, 高優先度のタスクがレジスタセットに割り当たっていれば, 高速化につながる.

割り当てアルゴリズムに FIFO を用いた際は, レジスタセットに存在しているタスクが将来使われる可能性が低い場合には有効である. LRU を用いた際には, レジスタセットに存在しているタスクが後続でも頻繁に実行されると予想される場合には有効である.

アプリケーションによって異なるが, 短い周期で実行されるタスクが存在することは考えられる. その場合は,FIFO よりも LRU を用いた方がよいと考えられるが, 短い周期で実行されるタスクが高優先度であると想定すると, 優先度に基づく割り当てアルゴリズムの適用が考えられる.

2.2 最適な割り当てアルゴリズム

FIFO,LRU,PRI のアルゴリズムで割り当てを行い, ディスパッチの際, 次実行タスクがレジスタセットに割り当たっている割合を求めるシミュレーションを行う.

タスクは, 図 2 に示す割合で一周期に実行されるとし, 優先度はタスク 1 が最も高い優先度でタスク 16 が最も低い優先度とする.

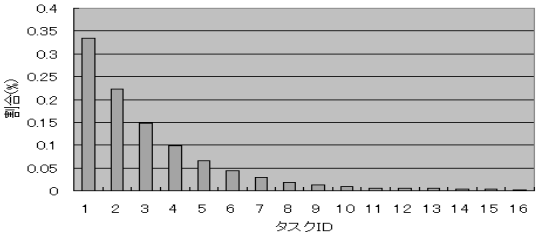

図 2: 一周期に出現するタスクの割合

その際, 次実行タスクがレジスタセットに存在する割合を図 3 に示す.

Improvement of task switching with multiple register set -Development operating system-
†Toshiyuki Koyama, Arata Miyaruchi, Syuichi Arai
‡Musashi Institute of Technology

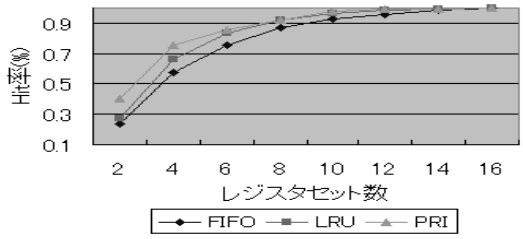

図 3: 実行タスクがレジスタセットに存在する割合

図 3 より, 高優先度のタスクが短い周期で出現する場合は, 優先度に基づいたアルゴリズムがもっとも高速に実行できる.

また, リアルタイムシステムにおいては, 実行される頻度は低いが, 高速に応答したいタスクが存在する. 例えば, 緊急時に起動したいタスクがそれに当たる. この場合, 割り当てアルゴリズムとして FIFO や LRU を用いると, 高速に応答したいタスクのコンテキストが, レジスタセットに割り当たっていない状況が生じる. しかし, 優先度に基づくアルゴリズムを選択し, 高速に応答したいタスクに最も高い優先度を割り当てることで, 頻度が低いが高速に応答することが必要な場合にも対処できる.

3. OS の開発

高速化を行っていないベースプロセッサ (36.1Mz) とタスク切り替え高速化プロセッサ (34.6Mz) について比較を行う.

レディキューの操作は, レディキューのハードウェア化が行われているので, 専用命令を用いて行う. レディキュー操作命令は,1CLK で実行する.

コンテキスト切り替えは, 通常多くの時間を費やす. しかし, 高速化プロセッサでは, コンテキストの切り替えは, レジスタセットの切り替えで行えるので,1CLK で実行することができる. したがって, ディスパッチが高速に実行できる.

3.1 実装結果

表 1 にレディキューに追加, 削除する際の各プロセッサによる実行時間の比較を示す. また, 表 2 にディスパッチ時の比較を示す. ディスパッチは, ベースプロセッサでは, 次実行タスクの優先度により実行時間が異なるため, ここでは, 優先度 1 のタスクを取得する時間とする.

表 1,2 からわかるように, 追加は約 30 倍, 削除は約 20 倍高速化されている. ベースプロセッサのディスパッチに比べて高速化プロセッサでソフトウェアでのコンテキスト切り替えが発生した時の方が短い時間で実行される. これは, レディーキューから次実行タスクの取得を専用命令を用いて 1CLK で実行できるためである.

表 1: レディキュー操作に費やす時間 単位:$nsec$

	追加	削除
ベースプロセッサ	886	692
高速化プロセッサ	2.89	2.89

表 2: ディスパッチに費やす時間 単位:μsec

ベースプロセッサ	4.15	
高速化プロセッサ	Hit	Miss
	0.116	3.21

次に, システムコールに費やす時間の比較を行う. ディスパッチが発生しない場合を (I), 発生する場合を (II) とし, 表 3,4,5,6 にその結果を示す. ディスパッチは, 優先度 1 のタスクを取得する時間とする. act_tsk はタスクを起動,slp_tsk は起床待ち状態へ移行,wup_tsk は起床待ち状態を解除,chg_pri は優先度を変更するシステムコールである.

表を見ると, 次実行タスクがレジスタセットに割り当てられていない場合は, あまり効果は得られないが, それ以外は大きく高速化されている. 特にレディキューの操作とハードウェアでコンテキスト切り替えの実行がある時は, 大きく時間が削減される. よってタスク切り替え高速プロセッサは, レディキューの操作やディスパッチが多く実行されるアプリケーションに有効である.

表 3: act_tsk に費やす時間 単位:μsec

ベースプロセッサ	I	II	
	6.31	7.59	
高速化プロセッサ	2.14	HIT	MISS
		2.25	5.34

表 4: slp_tsk に費やす時間 単位:μsec

ベースプロセッサ	I	II	
	4.52	5.79	
高速化プロセッサ	0.780	HIT	MISS
		0.896	3.99

表 5: wup_tsk に費やす時間 単位:μsec

ベースプロセッサ	I	II	
	4.99	6.26	
高速化プロセッサ	0.867	HIT	MISS
		0.983	4.08

表 6: chg_pri に費やす時間 単位:μsec

ベースプロセッサ	I	II	
	5.76	7.40	
高速化プロセッサ	1.01	HIT	MISS
		1.07	4.16
レディキューに繋がれていないタスクを対象にした場合は供に 47CLK			

4. むすび

レジスタ管理ユニットにおける高速な応答の実現とリアルタイムシステムへの適応に最適な割り当てアルゴリズムを考察した. また, タスク切り替え高速化プロセッサ上で動作する OS を開発し, 同プロセッサの性能を評価した. その結果, 大きく高速化されたことがわかった.

しかし,OS は完全に完成しておらず, 設計においての改良の余地も多く残っている. 今後の課題として,OS の完成とアプリケーションによる評価があげられる.

本研究の一部は, 文部科学省科学研究費補助金課題番号 13680425 によって行われたものである.

参考文献

[1] 坂村 健:" μ ITRON4.0 標準ガイドブック", パーソナルメディア,2001