

## マイクロプロセッサ rj406 の OS に対する評価

1L-3

五味 智 大林 雄次 井谷 真 中川 圭介

電気通信大学

32ビット RISC マイクロプロセッサ rj406[1][2][3] についてオペレーティングシステムを実行したときの振舞いを調べるため、MINIX オペレーティングシステム [4] のシステムコール、特に fork についてシミュレーションによる動作解析を行ったので報告する。

## 1 命令アーキテクチャの概要

rj406 は 32bit のレジスタを 32 個持ち、ロードストアアーキテクチャを採用している。命令は演算命令、ロード・ストア命令、分岐命令、システム制御命令から構成されている。

演算命令は 3 アドレス指定で、レジスタ間演算とレジスタ-イミディエイト間演算がある。命令の種類には、加減算、論理演算、シフト、セット命令がある。乗除算や浮動小数点演算はコプロセッサで行なうものとし、rj406 自体には実装していない。

分岐命令はレジスタの値を直接参照して分岐する。また、ロード命令と分岐命令は遅延実行方式になっている。

割り込みはトラップ命令かハードウェアからの割り込み要求によって起こる。割り込みベクタは採用しておらず、常に定まった番地 (8 番地) から処理を開始する。

## 2 システムコール

動作の解析のためのシミュレーションは、MINIX オペレーティングシステムのプロセス制御のための

システムコールに対して行うことにした。以下に利用者プログラムから fork を実行したときの結果について考察する。fork は、

1. 子プロセスのメモリ割当
2. 親プロセスのイメージを子にコピー
3. メモリマネージャ(MM)のプロセステーブル変更
4. カーネルのプロセステーブル変更
5. カーネルに子プロセスのメモリマップ報告
6. 親及び子プロセスの起動

という手順で行われるが、実際はそれぞれの処理の間でメッセージの転送 (sendrec など) が行われている。表 1 は各々のステップについて、実行命令数、各命令の出現割合、遅延スロットに無効命令 (nop) が現れる割合、キャッシュミスの割合を示したものである。なお、プログラムの実行はシミュレータ上で行っているが、キャッシュ記憶は命令キャッシュとデータキャッシュに分かれ、ともに 16KB の容量を持たせてある。またブロック長は 1 語、連想度は 1 となっている。

表 1 から load と store 命令の出現頻度が高いことがわかる。これはメッセージ交換を行っている部分とそれ以外の部分に共通している。現在、rj406 ではチップへの実装の便利さから load 命令と store 命令の実行に 2 クロックが必要になっているが、できれば 1 クロックで実行できるように改良するのが望ましい。キャッシュに関してはミス率は高い。ブロックを 2 語、4 語と増やすと、命令のミス率は 5.2%、2.8% に、データは 49%、40% と改善される。

分岐命令の遅延スロットが有効な命令で埋まらない割合は 3% 以下であり、分岐成功の割合が 67% であることを考えると、遅延分岐は有効であると考えられる。

Performance of Microprocessor rj406 for Operating System  
Satoshi Gomi, Yuuji Ohbayashi, Makoto Itani,  
Keisuke Nakagawa  
University of Electro Communications  
1-5-1 Chofugaoka, Chofu-shi, Tokyo, Japan

### 3 メッセージ転送

メッセージ交換の処理は割り込みに関係しており、最も処理速度が要求される部分であると考えられる。この部分をさらに細分して調べたものが表2である。これは表1の2行目に対応するもので、メッセージは9語で構成されている。表2では2行目と最後の行にあるレジスタ退避と復帰がアセンブラ言語でかかっている。2行目の処理の中で8番地からの割り込み原因の検査を行う短いプログラムが実行される。また、このプログラムではすべてのレジスタを退避しているが、退避するレジスタの数を減らして、高速化することも考えられる。

このようにした結果でも、割り込み処理の占める部分は小さく、rj406の単純化された割り込み機能で十分であると考えられる。

### 参考文献

- [1] 清藤ほか, "マイクロプロセッサ rj406", 情報処理学会第46回全国大会講演論文集, 6-9, 1993.
- [2] 青柳ほか, "マイクロプロセッサ rj406 のアーキテクチャと評価", 情報処理学会第47回全国大会講演論文集, 6-5, 1993.
- [3] 菅原ほか, "マイクロプロセッサ rj406 の浮動小数点演算", 情報処理学会第48回全国大会講演論文集, 6-134, 1994.
- [4] A. S. Tanenbaum, "Operating Systems : Design and Implementation", Prentice-Hall, Inc., 1987.

表 1: fork の動作

処理内容	実行 命令数	load (%)	store (%)	ALU (%)	bra (%)	j (%)	loadi (%)	set (%)	nop			キャッシュミス	
									load (%)	bra (%)	j (%)	Inst (%)	Data (%)
fork 呼び出し	12	0	41.7	8.3	0	8.3	8.3	0	0	0	8.3	100.0	75.0
sendrec	422	20.4	19.0	20.4	8.3	5.5	11.1	0.7	4.3	2.4	1.9	44.0	43.5
メモリ割当	109	16.5	20.2	21.1	7.3	4.6	17.4	0.9	1.8	0.9	0.9	76.4	43.6
sendrec	365	21.6	18.4	19.5	8.8	6.0	8.2	0.5	5.2	2.5	2.2	5.7	42.9
イメージコピー	1573	33.2	32.7	24.9	8.4	0.3	0	0	0	0.1	0.1	3.2	96.5
send & rec	525	22.9	19.6	17.7	7.6	5.5	8.6	0.4	5.1	2.5	1.9	5.7	46.4
MM テーブル	1458	16.8	9.7	32.8	16.2	0.2	5.7	0.1	4.8	4.5	0.1	7.9	31.5
sendrec	412	20.4	19.9	19.9	7.8	6.1	9.7	0.5	4.6	2.2	1.9	4.4	44.0
カーネルテーブル	339	31.9	32.7	22.1	10.3	0	0.6	0.6	0.6	0.3	0	15.3	86.2
send & rec	547	22.9	18.8	17.9	7.7	5.9	8.6	0.4	5.1	2.4	2.4	4.9	37.6
	3	66.7	0	0	0	0	0	0	0	0	0	100.0	0
sendrec	411	20.2	19.7	20.0	7.8	6.1	9.7	0.5	4.6	2.2	1.9	4.1	41.2
メモリマップ	185	26.5	22.2	28.1	10.3	3.2	2.2	1.1	0.5	1.6	1.1	36.6	78.7
send & rec	547	22.9	18.8	17.9	7.7	5.9	8.6	0.4	5.1	2.4	2.4	1.6	37.6
子起動	4	25.0	0	0	0	0	0	0	0	0	0	100.0	0
send	307	22.5	20.2	20.8	7.8	4.9	8.5	0.3	3.9	2.0	2.0	10.1	44.3
親起動	54	25.9	0	25.9	7.4	3.7	25.9	0	3.7	0	3.7	50.0	16.7
send & rec	545	22.0	14.7	25.3	7.9	5.5	9.3	0.3	4.4	1.7	2.3	4.8	32.1
合計	7818	23.9	20.7	23.3	9.8	3.3	6.2	0.3	3.3	2.1	1.2	9.8	60.5

表 2: メッセージ交換の動作

処理内容	実行 命令数	load	store	ALU	bra	j	loadi	set	nop		
									load	bra	j
前処理	83	4	18	15	0	10	15	0	0	0	2
レジスタ退避	66	4	33	12	2	0	6	0	4	2	0
送受信判定	31	7	5	7	5	0	1	1	0	2	0
送信(成功)	116	23	19	29	15	7	9	1	4	2	2
受信(ブロック)	78	11	8	13	10	8	10	0	7	3	3
レジスタ復帰	48	33	1	4	0	1	2	0	3	0	1
合計	422	82	84	80	32	26	43	2	18	9	8