2H-07

プロセッサの違いに着目した走行モード変更機構の評価

福居 誠二† 佐藤 将也‡ 山内 利宏‡ 谷口 秀夫‡
†岡山大学工学部 ‡岡山大学大学院自然科学研究科

1 はじめに

オペレーティングシステム (以降，OS と略す) において，システムコール発行は，プロセスの走行モード変更を伴うためオーバヘッドが大きい．このオーバヘッドを削減する手法として，プロセスを任意の時点でスーパバイザモードに変更可能にする走行モード変更機構を提案した [1]．また，OS 空間を保護するため，仮想空間切り替え方式とセグメント切り替え方式を提案した [2]．ここで，走行モード変更に伴うオーバヘッドはプロセッサの性能差による影響が大きい．本稿では，プロセッサの性能の異なる 2 つの計算機として Pentium 4 または Core i7 を搭載した計算機を用い，プロセッサの違いに着目した走行モード変更機構の評価結果を報告する．

2 走行モード変更機構

2.1 基本機構

走行モード変更機構の概要を図 1 に示す．以降では，ユーザ空間において，ユーザモードで実行するプロセスを Umode プロセス，スーパバイザモードで実行するプロセスを Smode プロセスと呼ぶ．

Umode プロセスについて，AP を実行している場合は，ユーザ空間のみ利用可能であり，OS のシステムコール処理を直接に呼び出すことはできない．AP から OS への処理依頼は，AP からシステムコールを発行することで実現される．この方式では，システムコール発行の際にユーザモードとスーパバイザモードの走行モードの変更処理を伴うため，オーバヘッドが大きい．

一方，Smode プロセスは，AP が OS のシステムコール処理を直接呼び出す．この方式では，走行モードの変更が不要なため，オーバヘッドを削減できる．

また，プロセスは，モード変更システムコールにより，任意の時点で走行モードを変更できる．つまり，プロセスは AP の実行に関して，ユーザモードとスーパバイザモードの両方の走行モードを使い分けることができる．

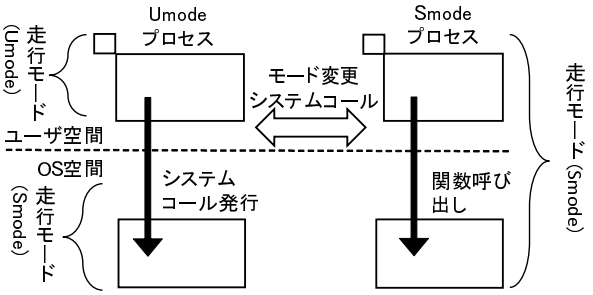

図 1 走行モード変更機構の概要

2.2 課題と対処

文献 [1] で提案した走行モード変更機構について，Smode プロセスはユーザ空間をスーパバイザモードで走行するため，OS 空間を保護できない欠点がある．この対処として文献 [2] において，仮想空間を切り替える方式 (以降，仮想空間切り替え方式と略す)，および Pentium 4 などに実装されているセグメント機構を用いた方式 (以降，セグメント切り替え方式と略す) を提案した．これらの方式により，OS 空間の保護が可能になる一方で，OS 空間保護のためのオーバヘッドが新たに発生する．なお，以降では，OS 空間を保護する Smode プロセスを Smode-P プロセスと呼ぶ．

3 評価

3.1 環境と実測

走行モード変更機構の評価を行うために用いた CPU アーキテクチャを表 1 に示す．Pentium 4 と Core i7 では，マイクロアーキテクチャの違いにより，パイプライン段数やキャッシュサイズなどが異なる．また，OS には ***AnT*** オペレーティングシステムを用いた．

走行モード変更を伴う処理において発生するオーバヘッドを以下のように定義し，表 1 に示した CPU を搭載した 2 台の計算機を用いて各オーバヘッドを測定した．

- O_s : SYSENTER 命令と復帰のオーバヘッド
- O_{vm} : 仮想空間切り替えのオーバヘッド
- O_{seg} : セグメント切り替えのオーバヘッド
- O_{sys} : I/O 命令発行処理におけるシステムコール処理のオーバヘッド

実測した処理移行時のオーバヘッドを表 2 に示す．

3.2 各走行モードの有効域

表 2 を用い，文献 [2] により，Umode プロセスと Smode-P プロセスの有効域を求める．ここで，走行モー

Evaluation of Dynamic Running Mode Switch Mechanism Focusing on Difference in Processors
Seiji Fukui†, Masaya Sato‡, Toshihiro Yamauchi‡, Hideo Taniguchi‡
†Faculty of Engineering, Okayama University
‡Graduate School of Natural Science and Technology, Okayama University

表 1 各 CPU のアーキテクチャ

CPU 名	クロック数	マイクロアーキテクチャ	コア数	スレッド数	ビット数	パイプライン段数	キャッシュサイズ		
							L1	L2	L3
Pentium 4	2.40 GHz	NetBurst	1	1	32bit	20 段	8KB	512KB	-
Core i7-2600	3.40 GHz	SandyBridge	4	6	64bit	16 段	64KB/コア	256KB/コア	8MB

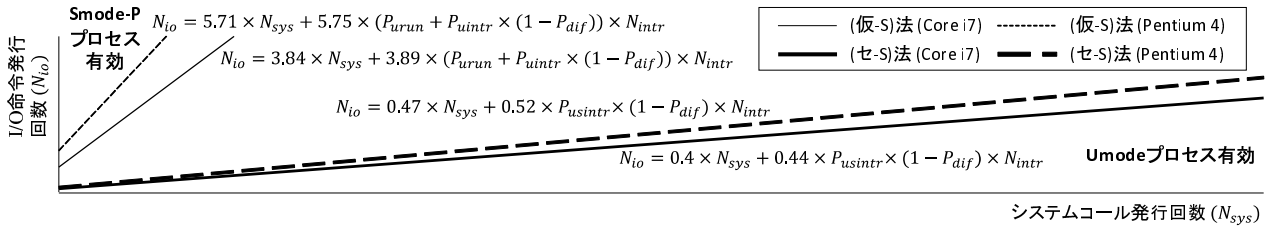

図 2 Umode プロセスと Smode-P プロセスの有効域

ド変更機構の有効域は，OS 空間保護方式とシステムコール発行方式の組み合わせによって異なる．システムコール発行方式には，処理のオーバヘッドの小さい SYSENTER 命令を用いる．以降では，仮想空間切り替え方式と SYSENTER 命令の組み合わせを (仮-S) 法，セグメント切り替え方式と SYSENTER 命令の組み合わせを (セ-S) 法と略す．また，変数を以下に定義する．

- P_{dif} : 実行プログラムと割り込み処理を実行するプロセスが異なる確率
- P_{urun} : 実行プログラムをユーザ空間で実行する確率
- P_{uintr} : 割り込み処理をユーザ空間で実行する確率
- P_{usintr} : 割り込み処理をユーザ空間 (Smode-P) で実行する確率
- N_{sys} : システムコールの発行回数
- N_{io} : I/O 命令の発行回数
- N_{intr} : 割り込みの回数

ここで，Umode プロセスと Smode-P プロセスのオーバヘッドが等しい場合の式は，

$$N_{io} = a \times N_{sys} + b \times N_{intr} \quad (1)$$

と表現できる (a，b は変数)．これは，各方式における優劣の分岐線であり，分岐線の上側の場合は Smode-P プロセスで走行させることが有効である．また，各方式における変数 a，b の定義を表 3 に示す．

表 2 と表 3 より，Core i7 における各オーバヘッドは，Pentium 4 に比べて全て減少しており，各方式の変数 a，b の分母と分子は小さくなる．また，O_{vm}，O_{seg} といった OS 空間保護のオーバヘッドの減少率は，他のオーバヘッドと比べて大きい．以上より，変数 a，b の分母と分子を比べると，分子の方が減少率が大きく，Core i7 における変数 a，b の値は，共に Pentium 4 より小さくなる．

ここで，表 2 の Pentium 4 と Core i7 における各オーバヘッドを考慮した場合の (仮-S) 法と (セ-S) 法における式 (1) を図 2 に示す．図 2 より，Pentium 4 と比べて

表 2 処理移行時のオーバヘッド

変数	処理時間 (μ 秒)		減少率
	Pentium 4	Core i7	
O_s	0.0079	0.0035	55%
O_{vm}	0.5175	0.1514	71%
O_{seg}	0.0467	0.0174	63%
O_{sys}	0.1721	0.0744	57%

表 3 各方式における変数 a，b の定義

方式	a	b
(仮-S) 法	$\frac{O_{vm} \times 2 - O_s}{O_s + O_{sys}}$	$\frac{O_{vm} \times 2}{O_s + O_{sys}} \times (P_{urun} + P_{uintr} \times (1 - P_{dif}))$
(セ-S) 法	$\frac{O_{seg} \times 2 - O_s}{O_s + O_{sys}}$	$\frac{O_{seg} \times 2}{O_s + O_{sys}} \times P_{usintr} \times (1 - P_{dif})$

Core i7 の方が N_{io} 切片は下にさがり，N_{sys} の傾きは小さくなっており，Smode-P プロセスの有効域は拡大している．つまり，同一のシステムコール発行回数に対して，Core i7 は，Pentium 4 と比べて少ない I/O 命令発行回数においても Smode-P プロセスが有効となる．

4 おわりに

プロセッサの違いに着目した走行モード変更機構の評価を行った．実測した結果より，Core i7 における仮想空間切り替えやセグメント切り替えといった OS 空間保護のオーバヘッドは，Pentium 4 と比べて大きく減少した．このように，プロセッサの性能向上により，走行モード変更機構における OS 空間保護のオーバヘッドが減少することで，走行モード変更機構の有効域が大きくなることを示した．

参考文献

[1] 横山 和俊, 乃村 能成, 谷口 秀夫, 丸山 勝巳 : 応用プログラムの走行モード変更を可能にするプロセス制御機構, 電子情報通信学会論文誌 (D), Vol.J91-D, No.3, pp.696–708 (2008.03).

[2] 公文 宏樹, 谷口 秀夫, 横山 和俊 : セグメント機構を利用した走行モード変更機構の実現, 情報処理学会研究報告, Vol.2010-OS-115, No.6, 電子媒体 (2010.08).