

プロセスの軽量化に関する検討

4K-9

谷口 秀夫 箱守 聡 横山 和俊

NTT データ通信 (株) 開発本部

1. はじめに

現在の多くのプロセッサは、割込みを契機に実行する処理の内容を変えることができる。そのため、複数の処理を行なうマルチプロセッシングシステムでは、プロセッサに割り当てる処理の単位を「プロセス」と呼び、制御プログラムがプロセスを切り換えている。

筆者らは、プロセスのプロセッサスケジュールを効率的に行なうためのプロセス走行環境と制御方式について検討している。本稿では、プロセスを構成する要素、構成要素をプロセス間で共有する時の得失、プロセス制御での留意点およびプロセス形態について、報告する。

2. プロセスの構成要素とその共有

2. 1 プロセスの構成要素

プロセスを構成する要素は、以下の5つに分類できる。

(1) テキスト部やデータ部およびスタック部からなるプロセスイメージ

(2) プロセスが利用するレジスタ

(3) プロセスの状態などを管理するプロセス管理テーブル

(4) 仮想空間を実現し、メモリへのアクセスを管理するメモリ管理テーブル

(5) ファイル等の資源へのアクセスを管理する資源アクセス管理テーブル

これらの中で、各プロセスが個別に保有する必要があるものは、(1)のスタック部と(2)のレジスタおよび(3)のプロセス管理テーブルである。他の構成要素は、プロセス間で共有することができる。

2. 2 プロセス間で構成要素を共有する得失

プロセス間で共有が可能なプロセスの構成要素を、各プロセスで共有する時の利点と欠点を表1に示す。表1からわかるように、構成要素の共有により処理性能が向上する。しかし、プロセス間の保護を実現しようとする、処理が複雑化する。

3. プロセスの制御

3. 1 制御の観点

プロセスを制御する際は、次に走行させるプロセスを選択する処理とプロセスを切り換える処理に留意する必要がある。

次に走行させるプロセスを選択する処理は、優先度とFIFOによる制御が多い。固定優先度の場合、プロセス数に関係なく一定の選択速度である。しかし、可変優先度の場合、優先度の変更処理時間がプロセス数に依存する。つまり、選択速度が変わる。また、マルチプロセッサの場合、次に走行するプロセスの選択方式が重要になる。例えば、プロセス切り換えによりプロセスの走行プロセッサが移動すると、プロセスのデータを複写すること等が必要になる。この時には、メモリキャッシュのヒット率が低下する。

プロセスを切り換える処理は、プロセス間でメモリ管理テーブルを共有して論理空間を共有するか否かにより、大きく異なる。

論理空間を共有していない場合、プロセッサの高機能化により論理空間の切り換えは1命令でできるため、この処理のオーバーヘッドはあまり問題にならない。また、プロセス間でのメモリアクセス保護を実現しているテーブルも自動的に切り換えられる。むしろ、切り換え後に、

表1 プロセスの構成要素をプロセス間で共有した時の利点と欠点

共有する構成要素	利 点	欠 点
メモリ管理テーブルの共有により論理空間を共有	・プロセスを切り換えてもメモリキャッシュやTLBのヒット率が低下しない。	・プロセス固有な領域への不当アクセスを防ぐには処理が複雑化する。
さらにテキスト部を共有	上記に加え、 ・メモリ効率が向上する。	上記に加え、 ・共有するためにメモリ管理テーブルの制御が複雑化する。
さらにテキスト部とデータ部を共有	上記に加え、 ・データのアドレス渡しによる高速なプロセス間通信ができる。	上記に加え、 ・1つのデータを1つのプロセスで占有するには処理が複雑化する。
資源アクセス管理テーブルを共有	・同じ資源を利用したプロセス間の協調処理を効率的に行なえる。	・1つの資源を1つのプロセスで占有するには処理が複雑化する。

メモリキャッシュやTLBのヒット率が低下することが問題になる。

これに対し、論理空間を共有している場合、上記の問題はない。しかし、プロセス間でメモリアクセス保護を実現する時には管理テーブルを書き換えることが必要になり、このオーバーヘッドは非常に大きい。

3.2 応用プログラムへのインタフェース

プロセスの構成要素をプロセス間で共有すると、表1に示したように、プロセス間の通信が速くなる等の利点がある。一方、共有する資源を複数のプロセスでアクセスするため、応用プログラムは、新たに排他などによる同期の制御を行なう必要がある。したがって、制御プログラムは、同期の制御のための機能を高速な形で応用プログラムに提供する必要がある。

4. プロセスの形態

プロセスの形態は、プロセスの構成要素をプロセス間で共有する範囲により分類できる。同じプログラムファイルを利用して生成したプロセスにおける形態の分類を表2に示す。

表2において、項番0は通常のプロセスである。項番2はスレッドと呼ばれている軽量化されたプロセスである。

表2 プロセスの形態

項番	共有の範囲	空間	資源アクセス	アクセス保護
0	テキスト	別	非共有	有
1	テキスト	別	共有	有
2	テキスト・データ	同	共有	無
3	テキスト・データ	同	非共有	無
4	テキスト・データ	同	共有	有
5	テキスト・データ	同	非共有	有
6	テキスト	同	共有	無
7	テキスト	同	非共有	無
8	テキスト	同	共有	有
9	テキスト	同	非共有	有

5. おわりに

プロセスを構成する要素は、大きく5つに分類できる。これらを各プロセス間で共有すると、処理性能は向上する可能性が高い。しかし、プロセス間の保護が問題になる。プロセスの形態として種々のものが考えられ、応用プログラムの処理内容に合わせた形態が必要である。

今後は、各形態のプロセスを実現し、性能評価により、それらの適用領域を明確にする予定である。