

マルチPSI要素プロセッサPSI-IIの メモリ管理とプロセス管理

吉田 裕之 池田 守宏 中島 浩 横田 実 中島 克人
(沖電気工業(株)) ((株)三菱電機) ((株)日本電気) ((財)ICOT)

1 はじめに

第五世代コンピュータシステム開発プロジェクトの一環として、現在ICOTにおいてマルチPSIが開発されつつある。マルチPSIの要素プロセッサPSI-IIは、プロジェクト前期において開発され現在運用中の、逐次型推論マシン(PSI)のCPUを小型化すると共に、実行速度と機能に関しても改良を加えたものである。

本稿では、先ずPSI-IIで採用した多重論理空間について述べ、次にプロセス管理方式とメモリ管理方式について報告する。

2 多重論理空間

2.1 多重論理空間の必要性

複数のプロセスがイベントの発生によって起動されるマルチ・プロセス機能は、オペレーティング・システムを記述するうえで重要な機能である。

このマルチ・プロセス機能を単一論理空間で実現させるためには、一枚の論理空間を例えば複数のエリアに分割し、各プロセスに個々のエリアのいくつかを割付ける方式が考えられる。しかしこの場合は、生成プロセス数がエリアの総数により制限されるという欠点がある。

これに対し多重論理空間では、プロセス毎に独立な論理空間を割当てるので、プロセス数に対する論理的な制限はない。従ってPSI-IIではマルチ・プロセス機能を柔軟にサポートするために、多重論理空間を実現した。

2.2 多重論理空間におけるアドレス変換

各プロセスの参照空間は、プロセス固有のアドレス変換テーブルにより論理アドレスから物理アドレスにマッピングされる。つまり同一の論理アドレスでも、プロセスにより参照される物理アドレスは異なる。以下にその具体的な実現法について述べる。

論理アドレス32ビットは、上位3ビットのエリア番号、ビット28～10(19ビット)のページ番号、下位10ビットのオフセットに分割される。主記憶は1Kワードのページ(物理ページ)単位に管理され、物理アドレスは、エリア番号に対応するベース・アドレスとページ番号の和により、アドレス変換テーブルをひいた結果(物理ページ)とオフセットによって決定される(図1参照)。つまり各エリアには、アドレス変換テ

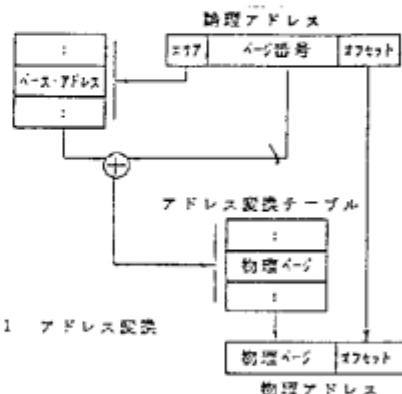


図1 アドレス変換

ル上のベース・アドレスが示す位置から連続した物理ページが割当てられる。プロセス毎に固有のベース・アドレスが割当てられるので、各プロセスはプロセス固有にアドレス変換テーブルを持つことができる。PSI-IIではこれらのアドレス変換をハードウェアで実現している。またベース・アドレスのセットはファームウェアにより行われる。

3 プロセス管理

PSI-IIでは各プロセスが動作するために必要な環境(プロセスのステータス、ファームウェアの作業環境、プロセスで使用しているエリアの管理情報等)を、メモリ上の特別なエリア(システム・エリア)に保存している。プロセスの動作環境はコントロール・ブロック(プロセスの数だけ存在する)にひとまとめにして保存され、コントロール・ブロックはシステム・エリア上でスタック的に管理される。システム・エリアはスタックの伸長に応じてメモリが割付けられるので、プロセスはメモリ容量の範囲内で何個でも生成できる。

プロセスはシステム内に複数個存在し、生成されたり消滅したり、また実行が別のプロセスに切り替わったりする。プロセスを生成するには、新しくコントロール・ブロックをシステム・エリア上のスタックに割当てると共に、プロセスで使用するエリアに応じたアドレス変換テーブルのベース・アドレスや必要な物理ページの確保等の初期値をコントロール・ブロック内にセットする。これらの新しいブロックの生成はファームウェアで行わ

Memory and Process Management for Multi-PSI Element Processor (PSI-II)

H. YOSHIDA¹ M. IKEDA² H. NAKASHIMA² M. YOKOTA³ K. NAKAJIMA⁴

1. Oki Electric Industry 2. Mitsubishi Electric 3. NEC 4. ICOT

れる。

プロセスを消滅させるときは、消滅させるプロセスのコントロール・ブロックをシステム・エリア上のスタック中から探し出し、コントロール・ブロックの情報を基にこのプロセスに割付けられている物理ページの解放等の処理を行う。このときコントロール・ブロック自体は消滅せず、後にソフトウェアにより再利用される。

プロセスを切り替えるときは、現在のファームウェアの作業環境や、使用しているエリアの情報（物理ページ数、アドレス変換テーブルのベース・アドレス等）をそのプロセスのコントロール・ブロックに返還させた後、切り替えるプロセスのコントロール・ブロックの内容をロードする。PS I-IIではこれらの処理をファームウェアでサポートしている。

4 メモリ管理

4.1 効率のよいメモリ管理の必要性

論理型言語の実行は、変数領域や実行順序の制御情報を、動的に生成することにより実現される。PS I-IIではプログラムの実行時に実行環境を保持するための3本のスタック・エリア（各プロセス固有）と、またside effectを必要とするデータやプログラム・コードを保持するためのヒープ・エリア（全プロセス共通）を必要とする。

以上述べたように、1つのプロセス内でのプログラムの実行には最低4エリアあれば足りるが、実行に必要な各エリアのメモリ量をあらかじめ予測できないこと、さらにメモリを有効に利用したいことから、PS I-IIのメモリ割当ては、前もって固定サイズを割当てる方式ではなく、プログラムの実行時にon-demandで行う方式とした。

4.2 グレイ・ページによるメモリ管理

メモリ・アクセス回数が多い論理型言語をインプリメントする上で、メモリ管理のオーバーヘッドはプログラムの実行速度を低下させる原因の一つである。PS I-IIでは、メモリ管理のオーバーヘッドを軽減するためにグレイ・ページという概念を導入した。

グレイ・ページとは各エリアに割付けられた最後の物理ページのことであり、グレイ・ページか否かはアドレス変換テーブル上のグレイ・ビットによって示される。グレイ・ページへアクセスすると、アドレス変換テーブル参照時にハードウェア的に割り込みが発生し、残りページがないこと（新しいページを割付ける必要があること）が事前に通知される。

これによりスタックの伸長のたびにファームウェア等でページ境界を越えるか否かを常に監視している必要がなく、全体的な実行速度の向上につながる。

4.3 グレイ・ページの割付け方法

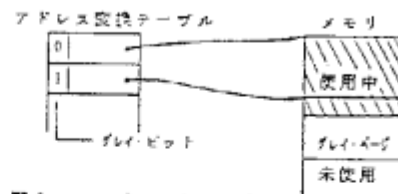


図2 ヒープ・エリアのグレイ・ページ割当て法

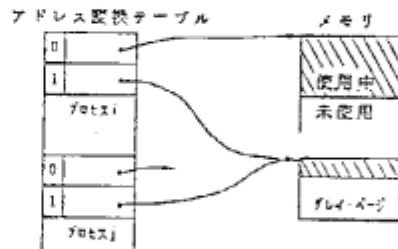


図3 スタック・エリアのグレイ・ページ割当て法

前述のように、PS I-IIではプログラムの実行に、全プロセス共通のヒープ・エリアと、プロセスに固有な3本のスタック・エリアの計4つのエリアを必要とする。仮にグレイ・ページを各エリアの最後に1ページずつ割付けるとすると、全体で生成プロセス数の3倍+1ものグレイ・ページが存在することになる。これでは物理ページの利用効率が悪い。

そこでPS I-IIでは、グレイ・ページ割込みの割込みレベルを最も高く（最優先）し、マシン・レベルにおいては各プロセスが同時には実行されないことを利用して、3本のスタック・エリアについてはスタックごとに全プロセス共通のグレイ・ページを割付けた。

ヒープ・エリアでグレイ・ページ割り込みが発生した場合は、グレイ・ページをそのまま新しいページとして与え、未使用の物理ページを次のグレイ・ページとなるように設定する。（図2参照）

スタック・エリアでグレイ・ページ割込みが発生した場合、ヒープ・エリアと同様な処理を行うと同時に、他の全プロセスのグレイ・ページ・ポインタを書き換える必要があり、プロセス数に比例してオーバーヘッドが大きくなる。従ってPS I-IIでは、グレイ・ページの内容を新しいページに移し、グレイ・ページの設定をアドレス変換テーブルの次のエントリへ移す方式を採用した。（図3参照）

5 おわりに

本研究にあたり多くの貴重な助言をいただいたICOT、及び関連メーカーの方々に深く感謝する。

参考文献

西川 宏、横田 実、山本 明、滝 和男、内田俊一：
逐次型パーソナル推論マシンの設計思想とそのアーキ
テクチャ、Proc. of The Logic Programming Conference
'83 pp7. 2-1~12