

汎用OSにおけるマルチプロセスサポート方式の メモリ継承機能

5H-2

細内昌明 池ヶ谷直子 新井利明 木下俊之
日立製作所システム開発研究所

1. 動機と目的

近年の小型計算機やネットワークの技術進歩と普及に伴い、メインフレームにおいても、他機種との接続連携を考慮したオープン化対応機能の必要性が高まっている。

このため、メインフレーム用の汎用OSに対するPOSIXインタフェース¹⁾のサポートを提案した²⁾。オープン環境の標準インタフェースであるPOSIXを汎用OSに搭載することによって、オープンプロトコルを用いたプログラムのメインフレームへのポーティング支援を狙う。その結果として、相互運用性を向上させ、メインフレーム資産の有効活用をはかる。

本報告では、POSIXインタフェースのうち、汎用OSとの親和性が低いプロセス生成インタフェースの実現をアシストするメモリ継承機能の汎用OSへの適用方法を述べる。

2. POSIX-汎用OS間のメモリ管理方式の比較

POSIXでは、生成要求空間の仮想記憶等をコピーすることによって新プロセスを生成する。しかし、空間生成時に全仮想記憶をコピーすると、空間生成オーバーヘッドが大きいため、通常メモリ管理にメモリ継承機能が適用される³⁾。メモリ継承機能とは、生成要求空間と被生成空間との間で、テキスト等の読込専用領域を共有し、データやスタック等の書込可能領域を参照時または書込時にページ単位に複写する機能である。汎用OSでは、新

規空間では、常に新たなプログラムが実行される。このため、メモリ継承機能は存在せず、仮想記憶の共有は、全空間共通領域のみ可能である。

(1) セグメンテーション方式(POSIX)

図1に、POSIXの空間生成方法をアシストするセグメント方式のメモリ管理構造を示す。セグメント方式では、用途別にページテーブルが分離されるため、読込専用領域の共有は、ページテーブルの共有により容易に実現される。ページスチールの基準となるUIC (Unreference Interval Counter)を仮想記憶上に設け、仮想アドレス順にUICを更新・比較することで、共有領域のスチール対象ページを決定できる。

(2) セグメンテーションページング方式(汎用OS)

汎用OSの多くで採用されているセグメントページング方式のメモリ管理構造を図2に示す。本方式では、空間全体を、複数のページから構成される固定長のセグメントに分割する。

各領域を非連続に割り当てることが可能なため、仮想アドレス順にページスチールを行うと、オーバーヘッドが高くなる。このため、UICは、実記憶管理テーブル上に配置され、実記憶割当て参照順に更新される。ページスチールでは、ページテーブルを無効化するため、実アドレスを仮想アドレスに変換する。

(3) 共有容易性の比較

表1に、特定空間間のメモリ共有の容易性からみた前記2つのメモリ管理方式間の違いを示す。セグメントページング方式は、メモリ割当の自由度は高いが、メモリ共有容易性の点では以下の点で不利である。

- (a) UIC更新が仮想アドレス順でないため、共有ページテーブルだけを参照してスチールすることができない。
- (b) ページテーブルが論理的に分離されていないため、読込専用領域のページテーブルだけを共有することが簡単に行えない。

表1 メモリ管理方式間の共有容易性の比較

メモリ管理方式名称	セグメンテーション方式	セグメンテーションページング方式
UIC更新方法	仮想アドレス順	実記憶割当て順
共有単位	論理単位 (領域)	物理単位 (セグメント/ページ)

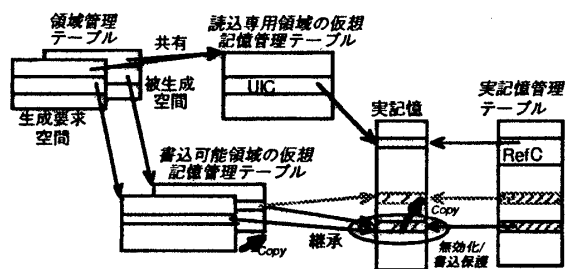


図1 セグメンテーション方式

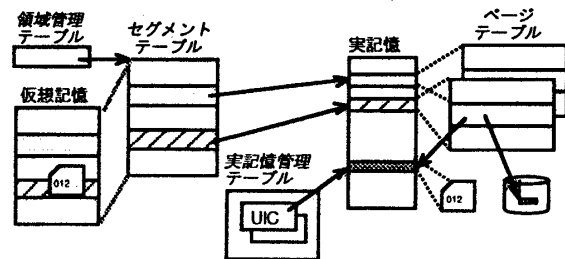


図2 セグメンテーションページング方式

3. セグメンテーションページング方式に適応したメモリ継承機能の実現方法

セグメンテーションページング方式において、特定空間間における、読込専用領域の仮想記憶共有と、書込可能領域の書込/参照時複写と、を実現する方法 (図3参照)を示す。

(1) セグメント単位の共有

共有単位を空間内の最大分割単位であるセグメントとし、ページテーブルを共有することで、仮想記憶共有を実現する。

(2) 読込専用領域と書込可能領域のセグメントの分離

領域割当て時に指定する領域グループを利用する。異なる領域グループの領域を、同一セグメント内に配置しないことで、共有するセグメント内への書込可能領域の混入を防止する。

(3) セグメント単位の参照時複写

書込可能領域のセグメントを無効化し、セグメント変換例外解決処理にてページテーブルを複写することにより、参照時複写を行う。

ページより大きなセグメントを単位とすることで、空間生成時のページテーブル作成オーバーヘッドを削減する。参照時複写の際にページを無効化すれば、ページ単位の参照時複写も可能である。

(4) 共有ページテーブルの一意識別

同一アドレスに配置される複数の共有ページテーブルを一意識別が必要があるため、ページテーブルの実アドレスを識別子とする。

(5) 共通領域を介したメモリ管理情報の受け渡し

ページテーブルの実アドレスやアドレスの割当範囲の情報等を共通領域を介して受け渡し、メモリ管理テーブルを再構築することで、仮想記憶継承を実現する。継承は、空間生成時以外でも可能である。

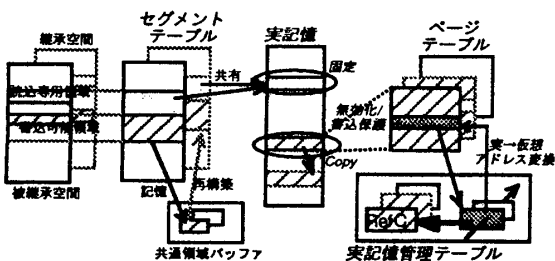


図3 セグメンテーションページング方式におけるメモリ継承の適用方法

4. 仮想アドレス連続保証機能

(1) 従来の仮想記憶割当解放方式の比較

セグメンテーション方式では、既割当領域に隣接した領域を割り当てる。従って、二重割当を防止するには、割り当てた領域の最終アドレスを記憶するのみでよい。

一方、セグメンテーションページング方式では、領域を割り当てる度に割り当てた領域の先頭アドレスや領域長情報を記録しなければならない。このため、必要な領域長が不明でオンデマンドに割り当てる場合は、割当オーバーヘッドが少なく、複数回割り当てた領域を一括管理可能なセグメンテーション方式が優れている。

しかし、解放を考慮すると、不要領域をどのアドレスでも即時解放できるセグメンテーションページング方式の方がメモリオーバーヘッド上優れる。

(2) 割当方法を選択可能とした仮想記憶管理方法

上記のように、オンデマンドの割当ではセグメンテーション方式が、解放ではセグメンテーションページング方式が、それぞれ優れる。そこで、両方式を両立させ、ユーザが選択可能とする方式を提案する (図4参照)。

領域割当要求時に特定の領域グループを指定すると、ユーザが使用宣言したアドレス範囲外に、その領域グループ内の既割当領域に隣接したアドレスに領域を割り当てる。

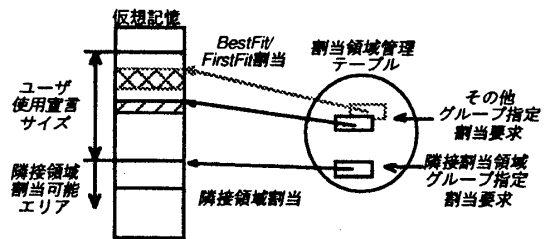


図4 仮想アドレス連続保証機能

5. まとめ

セグメンテーション方式とセグメンテーションページング方式を比較し、セグメンテーションページング方式に適応したメモリ継承機能の実現方法を示した。本方式により、ページ及びページテーブルの複写オーバーヘッドを低減し、従来の空間生成処理とほぼ同等のオーバーヘッドにて仮想記憶を継承した空間を生成できる。

参考文献

- 1) IEEE:POSIX1003.1a"POSIX part1:System Application Program Interface(API)",IEEE(1990)
- 2)池ヶ谷他：汎用OSにおけるマルチプロセスサポート方式の提案,本予稿集(1995)
- 3) Bach:The Design Of The UNIX Operating System, Printace-Hall InterNational (1986)

注) POSIXは、Electrical and Electronics Engineers(IEEE)で制定された標準仕様です。