

オペレーティングシステムOMNIの
メモリ管理

2N-7

矢野 達朗 長谷部 忍 飯間 豊
沖電気工業(株) コンピュータシステム開発本部

[1] はじめに

われわれは、密結合マルチプロセッサシステムORION⁽¹⁾⁽²⁾の上で、オペレーティングシステムOMNIを試作している。このOSは、ハードウェアを隠蔽し上位層に仮想マシンを提供するカーネルと、カーネルの上に構築されユーザインタフェースを提供するサービスの2つの層からなる⁽³⁾。

本稿ではOMNIカーネルのメモリ管理が、どのようにしてORIONを制御しているか、また、どのような機能を提供しているかについて述べる。

なお、記述言語にはC++を用いている。

[2] ORIONハードウェアの概要

ORIONは、RISC指向のプロセッサからなる密結合マルチプロセッサシステムである。その構成を図1に示す。全プロセッサから共有されるメインメモリに加えて、各プロセッサモジュールごとに他からはアクセスできないローカルメモリを備えているのが大きな特徴である。ローカルメモリはメインメモリよりもアクセスが高速である。

4Gバイトの論理空間は、ハードウェアによって16個のリージョンに分割されており、リージョン毎にローカルメモリを使うかメインメモリを使うかを指定することができる。

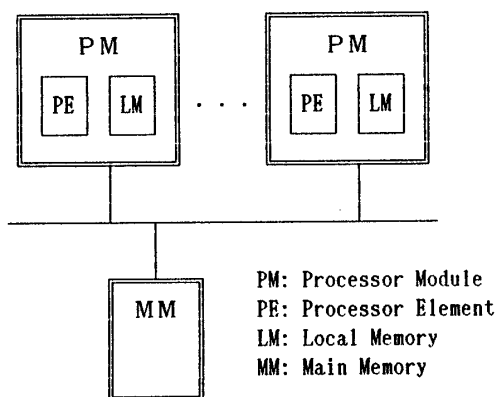


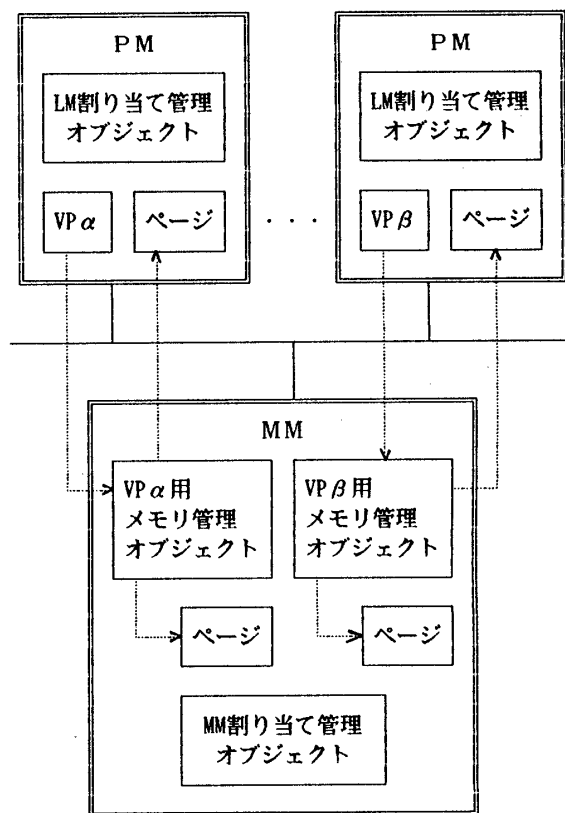
図1. ORIONの構成

[3] マルチプロセッサ対応

密結合マルチプロセッサでは、共有メモリアクセスによるバスの競合が大きな問題となる。ORIONでは、ローカルメモリを有効に使用し、共有メモリアクセスを極力少なくすることによって、この問題を回避することができる。

OMNIカーネルでは、カーネルを構成するオブジェクトのうち、各プロセッサモジュール間で共有する必要のあるものとそうでないものとを明確に区別し、それぞれ、メインメモリ、ローカルメモリに配置している。メインメモリ上の共有オブジェクトをアクセスする際にはスピンドック (test&set命令によるビジーループ) またはセマフォによる排他制御を行っている。

図2に実メモリ管理の構成を示す。メモリ管理オブジェクトはメインメモリ上に、各メモリの割り当て管理オブジェクトはそのメモリ上に置かれている。



PM: Processor Module LM: Local Memory
VP: Virtual Processor MM: Main Memory

図2. OMNIの実メモリ管理

A Memory Management in the Operating System OMNI
Tatsuro YANO, Shinobu HASEBE, Yutaka IIMA
Oki Electric Industry Co.,Ltd.

メモリ管理においては、対象となる資源がローカルメモリであるかメインメモリであるかの区別は管理しているが、ローカルメモリがどのモジュールに属するかという情報は管理していない。その代わりに、次のような方法を用いている。

- ・ローカルメモリの割り当て／解放
割り当て／解放を行おうとする仮想プロセッサ^[4]が、その時走行していたモジュールに対して作用する。
- ・ローカルな情報の管理
共有オブジェクトのモジュールごとの共有カウンタなど、ローカルに管理しなければならない情報は、ローカルメモリを使う。

この方式により、複雑なハードウェア構成にもかかわらずメモリ管理は非常に簡素になっている。また、プロセッサモジュールの数に依存しないので、モジュール追加などのシステム構成の変更にも柔軟に対応できるという利点がある。

仮想プロセッサの動的なマイグレーションは、現在サポートしていない。マイグレーションは、スワッピングを契機として発生する。2次記憶にスワップアウトされた仮想プロセッサがスワップインされる時には、負荷の最も小さいモジュールが選択され、もとのモジュール上で実行を再開するとは限らない。

[4] OMNIカーネル・メモリ管理の機能

4.1 メモリ管理クラス

論理空間は、logical_space, virtual_region, segmentの3つのクラスにより管理されている。サービス層は、これらのクラスのオブジェクトを操作することにより、論理空間を構築する。図3に論理空間の構成を示す。

以下に各クラスの概要を示す。

- ・logical_spaceクラス
仮想プロセッサの論理空間を管理する。このクラスのオブジェクトは、仮想プロセッサと対応して作成される。
- ・virtual_regionクラス
仮想プロセッサの論理空間を構成する要素であり、メモリの種別および仮想空間^[1]でのアドレスを与える。オブジェクトは要求に応じて作成される。
- ・segmentクラス
virtual_regionがハードウェアで区切られたリージョンを管理するのにに対して、segmentはソフトウェア的に区切られた領域を管理する。segment単位で、メモリの保護コード、2次記憶領域（ページング専用デバイスまたはサービスの指定するファイル）、共有等を設定することができる。オブジェクトは要求に応じて作成される。

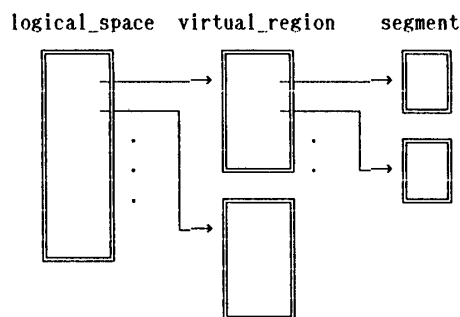


図3. 論理空間の構成

4.2 仮想記憶管理

仮想プロセッサの論理空間は、ワーキングセットによって管理されている。ワーキングセットから追い出されたページは、2次記憶に退避されるが、そのページが割り当てられていた実メモリは、他の用途に使用されなければ、次に参照された時には即座に使用可能となる。

仮想プロセッサの論理空間を他の論理空間にコピーする場合には、仮想記憶管理用オブジェクトおよびメモリの「一時的な共有」が行なわれ、物理的なコピーは実際に必要となるまで遅延される。これにより、論理空間の高速なコピーを実現している。

segmentを共有することにより、論理空間の一部を他の仮想プロセッサと共有することができる。共有領域もまた仮想記憶管理によって管理されており、実装メモリ量を越える共有領域を持つことが可能である。

[5] おわりに

本稿では、オペレーティングシステムOMNIのメモリ管理について述べた。メモリ管理の特徴としては、次のようなものがあげられる。

- ・モジュールの数に依存しないこと
- ・ワーキングセットによる仮想記憶管理
- ・copy on writeによる論理空間の高速なコピー
- ・memory mapped file

OMNIの試作はほぼ終了している。これからは、ハードウェアも含めたシステム全体の性能評価を行なう予定である。

参考文献

- [1] 阿部、他「ORIONアーキテクチャの概要」、情報処理学会第37回全国大会2N-5。
- [2] 宮本、他「ORIONプロセッサの概要」、情報処理学会第37回全国大会2N-6。
- [3] 飯間、他「オペレーティングシステムOMNIの概要」、第38回全国大会。
- [4] 長谷部、他「オペレーティングシステムOMNIのプロセッサ管理」、第38回全国大会。