5F－2

軽量オブジェクトを基礎としたオペレーティングシステムの設計と実装

谷森 徹　繁田 聡一　清水 謙多郎　芦原 評
電気通信大学 情報工学科

1 はじめに

本稿では、オペレーティングシステムを保護されたオブジェクトの集合体として実現する手法について述べる。オブジェクトとはデータ構造とそれを操作する手続き (メソッド) からなる保護された実体である。保護ドメインを明確に規定するとともに、モジュール間の高速な呼出しを実現することを目指す。そのため、我々は、以下のような構成上のサポートを行うこととした。

- メソッド呼出しは、保護されている。
- メソッド呼出しのセマンティクスは、局所的な手続き呼出しと同様である。
- メソッド呼出しには、局所的な手続き呼出しと同程度の時間しか要しない。

また、システムの構造化設計を支援するため、アクセス許可の条件を柔軟に設定できる、キー/ロック方式を拡張した保護機構 [1] を用意する。

2 保護されたモジュールの実現法

Hydra[6] は手続きを単位としてドメインを構成することができ、それだけ粒度の細かい保護を実現するが、手続き呼出しのたびにカーネルが介在するため効率が悪い。Mach と多くのマイクロカーネルベースのシステムでは、プロセスがドメインであり、プロセス間通信においてドメインの切替えが生じる。ドメインの切替えはアドレス空間とスレッドの切替えを伴う。Spring[5] は、個々にアドレス空間を持つドメインから構成される。スレッドがほかのドメインを呼び出す時、保護されたエントリポイント door を呼び出す。この際、スレッドは切り替わらずアドレス空間のみが切り替わる。Lipto[4] も、同様の機構を実現し、ドメインを越える呼出しは proxy を介して行なう。Opal[3] は、単一仮想アドレス空間内にセグメントを単位とした複数のドメインを持つことができる。ドメインを越えたオブジェクトの呼出しは potal を介して行ない、スレッドは切り替わらない。SPIN[2] は、強く型づけされたプログラミング言語により効率的で粒度の細かい保護と高速な呼出し機構を実現する。異なるドメインのオブジェクトを同じアドレス空間上に配置することができるが、特定の言語での利用が前提となる。

本システムでは、オブジェクトごとにドメインを構成でき、ドメインを越えた呼出しにおいて、最初の呼出しではカーネルが介在するが、その後はユーザレベルで効率的な呼出しが行なわれる。

3 基本設計

本システムは、スレッドとオブジェクトを基本的な構成要素とする。各スレッドは、基本的な実行状態、アドレス空間内に割り当てられるスレッド固有のデータ領域、スタック領域のほか、サブジェクトとしての権限を表すスレッドキーリスト (スレッドが保持するキーのリスト)、およびスレッドが呼び出すメソッドのアドレスを記録する MAT(Method Activation Table) からなる。アドレス空間は1つの保護ドメインに対応し、ユーザが明示的に生成することができる。1つのアドレス空間上で複数のスレッドを走行させることができるが、それらのスレッドにはすべて同じアクセス権が与えられる。

スレッドがオブジェクトを呼び出すと、その時アクセス権のチェックが行われ、アクセスが許されるならば、オブジェクトがそのスレッドのアドレス空間にアクティベートされる。いったんアクティベートされたオブジェクトのメソッドは、アドレス空間内の MAT にそのアドレスが登録され、以後、MAT を使った間接呼出しにより、カーネルを介さずにメソッド呼出しが行われる。

このようなオブジェクトの呼出し機構は、ユーザが定義したオブジェクトに対しても、オペレーティングシステムの構成要素となるオブジェクトに対しても区別なく、統一的に適用される。オペレーティングシステムは、カーネルモードで動作する nucleus と、種々のサービスを実現するサービスオブジェクトから構成される。サービスオブジェクトとしては、現在のところ、ネームサー

The Design and Implementation of an Operating System Based on Light-Weight Objects
Toru Tanimori, Soichi Shigeta, Kentaro Shimizu and Hyo Ashihara
Department of Computer Science,
The University of Electoro Communications
1-5-1, Chofugaoka, Chofu-shi, Tokyo, Japan.

バ、ファイルサーバ、バッファサブシステム、デバイスドライバ、ページ置換えポリシーなどを実装中である。

4 オブジェクトの機構

各オブジェクトは、オブジェクト識別子、キーリスト、アクセス制御リスト (ACL)、外部参照表 (ERT)、自分自身のメソッドのアドレスを保持しているメソッドテーブル、その他コードやデータなどから構成される。

4.1 オブジェクトの呼出し

オブジェクトをまたいだ参照はすべて MAT を利用して行なわれる。MAT はオブジェクトがアクティベートされた時カーネルにより ERT から作られ、スレッドのアドレス空間内のユーザにとって読み出し専用のセグメントに置かれる。

MAT エントリには、最初カーネルにトラップするルーチンのアドレスが設定されている。これにより、スレッドが最初にメソッドを呼び出した時に、カーネルに制御が移る。カーネルは指定されたメソッドが呼出し可能かどうか、スレッドのキーリストとオブジェクトの ACL を照合して判定する。呼出しが許可された場合は、次のような操作が行なわれる。

オブジェクトがすでにいずれかのスレッドによってアクティベートされている時は、その仮想記憶上のイメージを共有するようページテーブルを設定し、オブジェクトは呼出し元のアドレス空間に取り込まれる。まだオブジェクトがアクティベートされていなければ、二次記憶上よりイメージが仮想記憶上にロードされ、そのオブジェクトの情報がシステムに記録され、アクティベートされる。このような操作はユーザに透過である。

4.2 オブジェクトのディアクティベート

システムは、アクティベートされているオブジェクトごとに、そのオブジェクトを現在呼び出しているスレッドの数をカウントする参照カウンタを MAT に保持している。すべてのオブジェクトの呼出しが完結し、参照カウンタの値が 0 になっとき、その間 ACL の変更がなされていれば、その時点で MAT エントリの無効化とオブジェクトのディアクティベートが行なわれる。MAT エントリの無効化により、次の呼出し時に再びカーネルにトラップし、カーネルによる保護のチェックが行なわれる。ACL の変更がなされていない場合は、定期的もしくは空きメモリ領域の減少によって起動されるガーベジコレクタにより、オブジェクトのディアクティベートが行なわれる。

4.3 オブジェクトの記憶管理

オブジェクトはオブジェクトレポジトリ (object repository) に格納される。オブジェクトレポジトリは、分散システム上で位置透過な、永続的な記憶域を提供する。オブジェクトが他のノードのレポジトリに格納されている場合は、プロクシオブジェクトが生成され、リモートのオブジェクトの通信が行なわれる。オブジェクトのアクティベートはシステムが自動的に行なうため、ユーザから見た場合、スレッドが分散システム内に分散するオブジェクトを自由に呼び出して利用できるようなイメージが与えられる。

5 現状

現在、我々はこのシステムを Intel x86 CPU を持つ PC AT 上での実装を行なっている。現在は、マイクロカーネルとオブジェクト呼出しを実現するための主要な機能の実装を行なっている。また、上記のオブジェクトを利用するための実行可能形式を生成できるようにするため、既存の C 言語処理系に変更を行なっている。ただし、言語仕様レベルのサポートは必要ではなく、上記の機構は特定の言語に依存するものではない。

参考文献

[1] 繁田 聡一, 谷森 徹, 清水 謙多郎, 芦原 評. 単一アドレス空間におけるオブジェクトのきめの細かい保護機構の設計, 情報処理学会第 53 回 (平成 8 年後期) 全国大会.

[2] B. N. Bershad, S. Savage, P. Pardyak, et. al., "Extensibility, Safety and Performance in the SPIN Operating System" *Proc. of the Fifteenth ACM Symposium on Operating Systems Principles*, pp. 267-284, 1995.

[3] J. S. Chase, H. M. Levy, M. J. Feeley and E. D. Lazowska, "Sharing and Protection in a Single Address Space Operating System" *ACM Trans. on Computer Systems*, Vol. 12, No. 4, pp. 271-308, 1995.

[4] P. Druschel, L. L. Peterson and N. C. Hutchinson, "Beyond Micro-Kernel Design: Decoupling Modularity and Protection in Lipto" *Proc. of the 12th ICDCS*, pp. 512-520, 1992.

[5] G. Hamilton and P. Kougiouris, "The Spring nucleus: A microkernel for objects" *Proc. of the Summer USENIX Conference*, pp. 147-159, 1993.

[6] W. A. Wulf, R. Levin and S. P. Harbison, *Hydra/C.mmp: An Experimental Computer System.* McGraw-Hill, 1981.