マイクロカーネル Lavender におけるプロセス管理方式

5Z－5

佐脇 秀登† 芝 公仁† 豊岡 明† 毛利 公一† 大久保 英嗣††
†立命館大学大学院理工学研究科 ††立命館大学理工学部情報学科

1 はじめに

現在構築中のマイクロカーネル Lavender は，メモリ，プロセス，スレッド，プロセス間通信，割り込みに関する機能を持つ．マイクロカーネルの技術は，多くのオペレーティングシステムで採用され，その有効性が認められている．しかし，そのマイクロカーネルにもまだ多くの課題が残されている [1]．マイクロカーネル方式のオペレーティングシステムでは，プロセス間の協調作業が頻繁に発生し，それらのオーバヘッドが問題となる．Lavender ではそれらの協調作業の支援を目的とした，レジデントアドレス空間とプロセスグループ機能を用意している．

(1) プロセスグループ機能

1 つの仮想アドレス空間内に複数のユーザプロセスを同時に存在させることを可能にする機能である．プロセスグループ機能により，プロセス間手続き呼び出しなどの協調作業は，スレッドのプロセス間移動として実現される．このスレッドのプロセス間移動により，プロセスグループ内の制御の移行のオーバヘッドを軽減することができる．また，プロセス間で共有メモリを確保することでプロセス間の協調作業も容易に実現できる．

(2) レジデントアドレス空間

Lavender における仮想アドレス空間は，1 つのカーネルアドレス空間，1 つのレジデントアドレス空間，複数のユーザアドレス空間からなる．カーネルアドレス空間には保護されたカーネルのコードとデータが配置され，ユーザアドレス空間には複数のプロセスが配置される．レジデントアドレス空間は，ユーザアドレス空間の一種であり，カーネルアドレス空間と同じく，ユーザアドレス空間の切り替えに影響されず，常に仮想アドレス空間上に存在する．レジデントアドレス空間内に配置されたシステムサーバなどのプロセスは，全てのプロセスと同じグループに属しているように見える．これにより，システムサーバとユーザプロセスの間でも，同一アドレス空間内のジャンプを用いたスレッドのプロセス間移動や，共有メモリによる協調作業が容易に実現できる．

本論文では，以上の機構を利用したプロセス間手続き呼び出しを実現する方式について述べる．

Process Management in Lavender Micro Kernel
Sawaki Hideto †, Masahito Shiba †, Akira Toyooka †, Koichi Mouri †and Eiji Okubo ††
†Graduate School of Science and Engineering, Ritsumeikan University
††Department of Computer Science,
Faculty of Science and Engineering, Ritsumeikan University

2 プロセス・スレッドモデル

Lavender におけるプロセスは静的な概念であり，1 つの仮想アドレス空間に属し，1 つ以上のセグメントに分割された実行環境である．ただし，プロセスが属するアドレス空間は，プロセスグループ機能によって他の複数のプロセスと共有されていてもよい．

スレッドは，動的な概念であり，レジスタセットとスタックを持った実行実体である．スレッドはプロセスが提供する環境の中で動作する．そして，スレッドは 1 つのプロセス内に複数あってもよい．

3 ネームサーバ

ネームサーバは，プロセス間で名前空間を共有するためのサーバであり，ユーザアドレス空間で動作する．名前空間とは資源に割付けられた名前を意味している．名前空間を共有する資源には次のようなものが挙げられる．

(1) プロセス間通信時のポート番号

ネームサーバは，プロセスがポートを割り当てられたとき，プロセス間通信サーバの依頼によって，そのポートの番号と名前をデータベースに登録する．また，プロセスがポートを削除したとき，ポートの番号と名前をデータベースから削除する．プロセスが通信を行おうとするとき必要となるポート番号を調べるために，送信側のプロセスはポート名を指定してネームサーバに検索の依頼を行う．ネームサーバはそれに対する検索を行い，プロセスに対応するポート番号を返す．

(2) スレッドのプロセス間移動時の移動先アドレス

プロセスグループ機能により，同一アドレス空間内に配置されたプロセス間では，スレッドのプロセス間移動を用いてオーバヘッドの少ない RPC が可能となる．

サーバプロセスは起動時に，遠隔実行させてもよい手続きをネームサーバのデータベースに登録する．登録されるデータは，手続き名とそのアドレス，及び所属するプロセスグループの名前からなる．

クライアントプロセスは，実行時に呼び出したい手続き名とプロセスグループ名を指定してネームサーバに検索の依頼を行う．ネームサーバは，それに対する検索を行い，対応するアドレスを返す．アドレスは，実際には，その手続きのみからなるセグメントの番号と，オフセットなどからなる．クライアントのスレッドは，このアドレスによってのみ移動先を特定できる．

4　プロセス間手続き呼び出し

一般に RPC はメッセージ通信を利用して行われる．メッセージ通信は，カーネルを介したメッセージのコピーにより実現される．これは，送信側プロセスからカーネルへ，カーネルから受信プロセスへと計 2 回のコピーとなり，オーバヘッドが大きくなるといった問題がある．この問題を解決するため，Lavender ではプロセスグループ機能を用いる．

同一アドレス空間内に置かれたプロセスは，共有を許可した部分のみを他のプロセスと共有し，その他の部分は保護される．共有の許可は実行時にネームサーバへの登録により行い，この時点で識別子 (関数名など) の衝突などは検査される．

共有はデータ，コード共に可能で，ページテーブルの書き換えにより同一ページを複数のページテーブルにマッピングする．この方法は実際にはページのコピーは行われないので高速ではあるが，共有の単位がページであるために，共有したい部分より多く公開されてしまう恐れがある．これはページの区切りまでの僅かな部分で，平均するとページサイズの約半分であるがセキュリティホールとなる危険性を持っているため，放置するべきではない．この問題を防ぐため，共有する部分は別セグメントに確保し，このセグメントのみを共有する．これによって余分な部分の漏洩という問題を解決することができる．

コードについても，上記のデータの共有と同様の機構を使うことによりサーバプロセスの手続きを自プロセスへ取り込んでいるかのように見せることができる．これにより，プロセスグループ間での RPC は，実際にはプロセスの実行実体であるスレッドのプロセス間移動に置き換えられる．

このスレッドの移動は，アドレス空間が同じであれば，移動先のアドレスを知ることにより可能となる．アドレスは，実際には移動先の手続きのみからなるセグメントの番号と，そのセグメントの先頭からのオフセットである．これらの値は，ネームサーバへの問い合わせによって取得することができる．呼び出したい手続き名を指定してネームサーバに検索の依頼を行うと，手続き名と呼び出し元のプロセスの所属するプロセスグループをキーに検索が行われる．ネームサーバ，は呼び出し元と同一ホスト内のプロセスに該当する手続きが見つかれば，その手続きのアドレスを含む構造体を返す．アドレス構造体は，ホストネーム，プロセスグループ名，セグメントの番号，セグメントの先頭からのオフセットからなる．

スレッドの移動は，このアドレス構造体を引数としたライブラリを呼び出すことによって行われる．ライブラリ手続きでは，まず最初にその呼び出しがローカルかリモートかを調べる．リモートであれば，このライブラリは通常の RPC のように手続きのアドレスと引数をメッセージとして組み立てて通信を行うスタブとして働く．ローカルであれば，該当するアドレスへのセグメント間手続き呼び出し命令のみが実行され，スレッドは僅かなオーバヘッドで他プロセスへ移動する．

従来の RPC と比較すると，各種オーバヘッドが軽減される．それらには，以下のようなものが挙げられる．

- カーネルを介さずに行うことができる．
- アドレス空間の切り替えがない．
- コンテキストの切り替えがない．
- 協調プロセスがスケジューリングされるのを待たなくてもよい．

また，プロセスグループ内におけるスレッドのプロセス間移動では，識別子によるネームサーバへの登録と，その識別子を引数としたライブラリ手続きの呼び出しだけといった単純な形をとっており，従来の RPC のようにスタブを作成する必要はない．ただし，RPC のようにローカルコールと同一のインタフェースをとれないという欠点を持つことになる．

なお，他プロセスのコードの利用に関しては SPIN でもアプリケーションレベルライブラリとして提案されている [1]．これは，Lavender と異なって，サーバなどの協調プロセスの機能をアプリケーションの一部に取り込む機能である．SPIN はこれにより，カーネルを介さずにコードの共有を実現している．

5　おわりに

本論文では，スレッドのプロセス間移動を中心に Lavender におけるプロセス管理方式について述べた．Lavender のスレッドのプロセス間移動が十分な効果を発揮するのは、実装上ローカルホスト内の呼び出しに限られる．このようなローカルホスト内での呼び出しの最適化を利用するには、リモートホスト上の手続きの実行時に以下の手法を用いることが有効であると考えられる．ネームサーバが手続き名からアドレスの検索を行い、手続きがリモートに存在したときは、ライブラリが自動的にリモートの手続きの転送を行い、レジデントアドレス空間に手続きを配置し、ネームサーバに登録を行うというものである．これ以降、この手続きの実行はオーバヘッドの少ない Lavender のスレッドのプロセス間移動によって行われる．

参考文献

[1] B. N. Bershad, C. Chambers, S. Eggers, C. Maeda, D. McNamee, P. Pardyak, S. Savage and E. G. Sirer: *"SPIN - An Extensible Microkernel for Application-specific Operating System Service"* ,Technical Report UW-CSE-94-03-03, Department of Computer Science and Engineering, University of Washington (1994).