2N-4 *Tender* オペレーティングシステムにおけるプロセス移動機能

石井 陽介† 谷口 秀夫††
† 九州大学工学部電気情報工学科 †† 九州大学大学院システム情報科学研究院

1 はじめに

プロセス移動機能とは、プロセスが走行する計算機を変更する機能である。分散環境においてプロセス移動機能を実現することで、負荷分散や協調処理のオーバヘッドを削減できる。さらに、保守や点検のために計算機を停止させなくてはならない場合、動作継続が必要なプロセスを他の計算機に移動させて走行させることが可能になる。

本稿では、我々が開発している ***Tender***(The ENduring operating system for Distributed EnviRonment) オペレーティングシステム[1] におけるプロセス移動機能について述べる。

2 プロセス移動機能

2.1 プロセスの構成

Tender ではプログラム構造を重視し、OS の操作対象を資源として分離し独立化させている。このため、プロセスを複数の資源に分離している。プロセスを構成する資源を図 1 に示す。矢印は、処理の依存関係を表している。また、資源「演算」は、プロセスへのプロセッサ割り当て単位を資源化したもので、プロセスとは独立して存在する。プロセスは、演算を確保することで、プロセッサの割り当てを受けて走行できる。

2.2 プロセス移動処理の特徴

Tender では、各資源が独立して存在できる。これにより、プロセス移動時に以下が可能となり、プロセス移動処理の高速化が期待できる。

(1) 必要な資源だけを移動 プロセスを構成する資源を全て移動させる必要はなく、移動に必要な資源だけを移動させることができる。

(2) 資源の再利用 資源の事前用意や保留により、資源の生成や削除を伴う処理を高速化できる。

3 実現方式

3.1 ヘテロ仮想記憶

プロセス移動機能を実現するために、ヘテロ仮想記憶[2] を利用する。ヘテロ仮想記憶には、プロセスが同一計算機内の仮想記憶空間の間を移動できる機能が実現されている。この機能を、遠隔の計算機上に存在する仮想記憶空間に移動できるように拡張することで、プロセス移動機能を実現する。

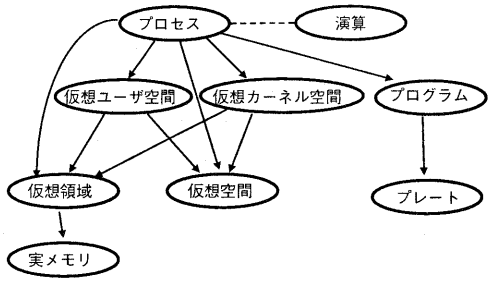

図 1 プロセスを構成する資源

3.2 同一計算機内の移動処理

既に実現されている同一計算機内のプロセス移動の様子を図 2 に示す。ここで、仮想領域は、外部記憶装置あるいは外部記憶装置と実メモリのデータ格納域を仮想化した資源である。仮想空間は、仮想アドレスの空間であり、仮想アドレスを実アドレスに変換する変換表に相当する。仮想ユーザ空間は、プロセッサが仮想アドレスによってカーネル／ユーザモードでアクセス可能な空間であり、仮想領域を仮想空間に「貼り付ける」ことにより生成される。ここで、「貼り付ける」とは、仮想空間が持つアドレス変換表に、当該の仮想領域のデータ格納域情報を設定することに相当する。プロセスが利用するテキスト部、データ部、BSS 部、ユーザスタック部は、仮想ユーザ空間上に存在する。同一計算機内のプロセス移動は、プロセスが利用する仮想ユーザ空間を移動先の仮想ユーザ空間に変更すること、すなわち、仮想領域を貼り付けている仮想空間を変更することにより行う。

3.3 遠隔の計算機への移動処理

プロセスを遠隔の計算機へ移動させるためには、移動元の計算機 (以降、ローカルと呼ぶ) に存在する移動対象のプロセス (以降、移動プロセスと呼ぶ) と同じコンテキストを持つプロセスを、移動先の計算機 (以降、リモートと呼ぶ) に生成し、ローカルの移動プロセスを削除する必要がある。ここで、コンテキストとは、プロセスが利用するテキスト部、データ部、BSS 部、ユーザスタック部、およびレジスタ群のことを指す。

移動処理の主体は、移動プロセスに関する情報の転送回数を削減するために、ローカルとした。なお、移

Process Migration Mechanism on ***Tender***
Yousuke ISHII† and Hideo TANIGUCHI††
†Department of Electrical Engineering and Computer Science, Faculty of Engineering, Kyushu University
††Graduate School of Information Science and Electrical Engineering, Kyushu University
Email: ishiiyou@swlab.csce.kyushu-u.ac.jp
tani@csce.kyushu-u.ac.jp

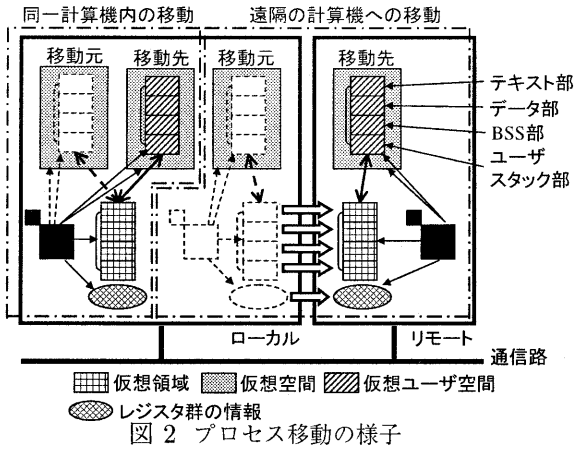

図 2 プロセス移動の様子

動処理の中で、リモートが処理を行う必要がある場合は、ローカルが遠隔手続呼出制御 [3] を利用して、リモートに対して処理を依頼する。遠隔手続呼出制御とは、遠隔の計算機上にある資源を、自計算機上の資源と同様に操作できるようにするものである。

リモートへのプロセス移動を行う際の処理手順を以下に述べ、その様子を図 2 に示す。

(1) リモートにプロセス生成
(2) 生成したプロセスの変身
(3) ローカルの移動プロセス削除

処理 (1) では、遠隔手続呼出制御を利用して、リモート上に移動先となるプロセスを生成する。生成処理を高速に行うために、カーネルプロセスを生成する。

処理 (2) では、処理 (1) で生成したプロセスを、遠隔手続呼出制御を利用して移動プロセスに変身させる。プロセスの変身とは、プロセスが持つコンテキストを変更することである。以下に、プロセス変身処理の内容について述べる。

(I) 動作空間の変更 処理 (1) ではカーネルプロセスが生成され、自身が持つプロセス管理表には、カーネル用の仮想空間が登録されている。そのため、移動先となる仮想空間をプロセス管理表に登録し、プロセスが利用する仮想ユーザ空間を、移動先の仮想空間上に生成できるようにする。

(II) 実行プログラムの変更 移動先となるプロセスの実行プログラムを、移動プロセスが利用する実行プログラムに変更する。ここで、実行プログラムとは、プロセスとして走行するプログラム、すなわち、プロセスが利用する資源「プログラム」のことである。資源「プログラム」とは、プログラムのテキスト／データの大きさと先頭番地、プログラムの開始番地の情報からなり、プログラムの実行形式を隠蔽している。プログラムの内容は、プレート上に存在している。ここで、資源「プレート」とは、永続的な記憶を提供するものである。プロセス移動の際、移動プロセスのプログラムの内容は、移動プロセスが利用している仮想領域に既に読み込まれているので、移動プロセスを構成する資源「プログラム」と資源「プレート」をリモートへ移動させる必要はない。処理内容は、まず、テキスト部として利用するための仮想領域と仮想ユーザ空間を生成する。次に、移動プロセスが利用しているテキスト部の仮想領域を、先に作成した仮想領域へ移動させる。仮想領域の移動を行うために、分散共有メモリ [3] の機能を利用して実メモリ間の複写を行う。実メモリ間で複写を行う方法は、NFS のようなネットワーク上でのファイルの共有機構を利用する方法 [4] と比べて、外部記憶装置に対する入出力操作が不要になるので、処理が速い。データ部、BSS 部、ユーザスタック部は、処理の高速化のために、それぞれが利用するための仮想領域と仮想ユーザ空間の生成だけを行う。

(III) 開始位置の変更 移動プロセスの途中状態から走行再開できるように、移動先となるプロセスのコンテキストの内、データ部、BSS 部、ユーザスタック部、レジスタ群の情報を変更する。処理内容は、ローカルの移動プロセスが利用しているデータ部、BSS 部、ユーザスタック部の仮想領域を、分散共有メモリの機能を利用してリモートに移動させ、(II) で予め作成しておいた仮想領域へその内容を複写する。また、レジスタ群の情報は、変身処理の引数として取るようにする。

処理 (3) では、ローカルの移動プロセスを削除する。処理の高速化のために、プロセス削除の際、プロセスの構成資源を資源再利用のために保留しておく。

4 おわりに

本稿では、***Tender*** におけるプロセス移動機能の実現方式について述べた。今後の課題としては、遠隔の計算機へのプロセス移動機能の実装と評価がある。

＜謝辞＞検討に協力頂いた、九州大学大学院システム情報科学府の田端利宏氏に感謝致します。

参考文献

[1] 谷口 秀夫 他: “資源の独立化機構による ***Tender*** オペレーティングシステム”, 情報処理学会論文誌, Vol.41, No.12, pp.3363-3374(2000)
[2] 谷口 秀夫, 長嶋 直希, 田端 利宏: “単一仮想記憶と多重仮想記憶を共存させたヘテロ仮想記憶の実現”, 情報処理学会研究報告, Vol.98., No.33, pp.87-94(1998)
[3] 下崎 誠, 谷口 秀夫: “***Tender*** オペレーティングシステムにおける分散共有メモリ機構の設計”, 電子情報通信学会技術研究報告, Vol.99, No.251, pp.33-40(1999)
[4] 森山 茂男, 多田 好克: “利用者レベルで実現したプロセス移送ライブラリ”, 情報処理学会 OS 研究会報告 91-OS-5, pp.41-47(1991)