

Tender におけるプロセス変身機能の設計

4 L-2

石井 陽介† 田端 利宏† 谷口 秀夫‡

†九州大学大学院システム情報科学府 ‡九州大学大学院システム情報科学研究院

1 はじめに

プログラムを実行する際、OS はその動作をプロセス単位で制御する。本稿では、プロセスの実行環境を変更するプロセス変身機能について述べる。また、我々が開発している *Tender* オペレーティングシステム^[1] におけるプロセス変身機能の実現方式について述べる。

2 プロセス管理

2.1 従来 (UNIX) のプロセス生成実行法

UNIX では、プロセスの生成と実行を `fork/exec` 方式を用いて実現している。ただ、この方式は、`fork` 時に行う親プロセスの状態複写が、`exec` 時に置き換えられるので、複写時に利用する CPU 資源とメモリ資源が無駄となる欠点がある。この欠点を解決するために、`vfork` や `copy-on-write` がある。しかし、`vfork` は子プロセスが親プロセスのアドレス空間の内容を変更できるので、親プロセスとの整合性が損なわれる恐れがある。また、`copy-on-write` は制御が複雑な上に、分散環境下では物理的な転送を常に必要とするので、処理のオーバーヘッドが大きい。

2.2 分散環境も考慮したプロセス生成実行法

元来、UNIX はスタンドアロン環境での利用を想定して設計されている。このため、分散環境の利点を生かしたプロセス生成実行法を提供していない。そこで、分散環境で利用する OS は、分散環境も考慮したプロセス生成実行法を提供することが望ましい。このために、以下の二点が考えられる。

第一に、プロセス生成実行法は、`fork/exec` 方式ではなく、アドレス空間を生成した後に直接プログラムをロードする方式を用いるべきである。これにより、プロセス生成時の無駄なメモリ間複写が省け、プロセス生成処理の高速化が期待できる。

第二に、計算機間のプロセス移動や、利用資源の再利用のように、プロセスの実行環境を柔軟に変更できる機能を実現するべきである。

ここでは、プロセス変身機能によって後者を実現す

ることとした。プロセス変身機能とは以下の三つの機能を指す。

(機能 1) 実行プログラムの変更

(機能 2) 開始位置の変更

(機能 3) 動作空間の変更

以降では、この三つの機能について述べる。

3 プロセス変身機能

3.1 実行プログラムの変更

実行プログラムの変更機能とは、プロセスとして実行されるプログラムを別のプログラムに変更する機能である。これにより、複数のプログラムを一つのプロセスとして逐次的に実行できる。

3.2 開始位置の変更

開始位置の変更機能とは、変更対象プロセスが次に走行する開始位置を変更する機能である。開始位置の変更には、以下の二つの機能がある。

(1) 初期状態に変更

(2) 指定された位置に変更

機能 (1) により、プロセスの再起動^[2] が可能になり、プロセス生成と消滅のオーバーヘッドを削減できる。機能 (2) により、チェックポイント機能や既存プロセスの情報からプロセスの任意の位置における状態を保存しておき、プロセスを保存した状態から再開させる。これにより、プロセスの実行途中で起る障害からの高速な回復処理を実現できる。

3.3 動作空間の変更

動作空間の変更機能とは、プロセスが利用する仮想記憶空間を別の空間に変更する機能である^{[3][4]}。動作空間の変更には、以下の二つの機能がある。

(1) 同一計算機内の別空間に変更

(2) 他計算機上の空間に変更

機能 (1) により、プロセス間の処理の疎密度に合わせた空間の変更や、データが存在する空間へプロセスが移動することによるデータの排他処理が可能となる。機能 (2) により、計算機間のプロセス移動が実現できる。このため、負荷分散、協調処理のオーバーヘッド削減、および保守や点検による計算機の停止に伴うプロセス停止の回避が可能となる。

4 *Tender* における実現方式

プロセス変身機能を提供する基本インタフェースを表 1 に示す。なお、一回の呼び出しで、表 1 に示す機

Design of Process Transformation Facility on *Tender*
Yousuke ISHII†, Toshihiro TABATA† and Hideo
TANIGUCHI†
Graduate School of Information Science and Electrical
Engineering, Kyushu University
Email: ishiiyou@swlab.csce.kyushu-u.ac.jp
tabata@swlab.csce.kyushu-u.ac.jp
tani@csce.kyushu-u.ac.jp

表 1 提供する基本インタフェース

変更対象	形式	機能
実行プログラム	trans_proc(pid,plateid,argv)	pid で指すプロセスのプログラムを plateid で指すプログラムに変更し、引数 argv を渡す。
開始位置	restart_proc(pid,restart_arg,argv)	pid で指すプロセスの開始位置を restart_arg で指定された位置に変更する。もしくは、初期状態に変更して引数 argv を渡す。
動作空間	move_proc(pid,vmid)	pid で指すプロセスの動作空間を vmid で指す空間に変更する。

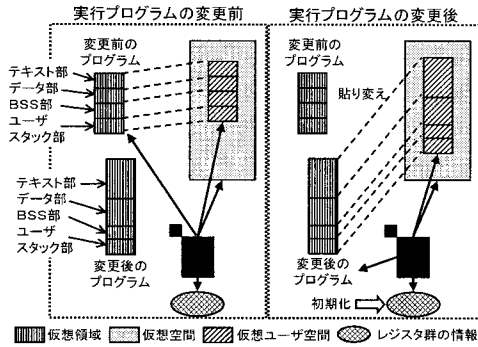


図 1 実行プログラムの変更

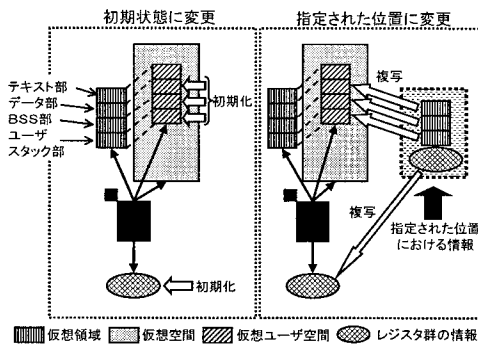


図 2 開始位置の変更

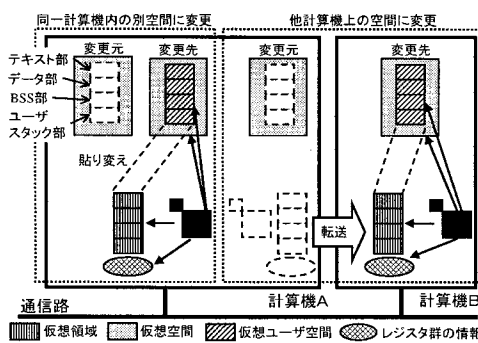


図 3 動作空間の変更

能を複数組み合わせることで実行することが可能である。

4.1 実行プログラムの変更

実行プログラムの変更の様子を図 1 に示す。図において、仮想領域は、実メモリあるいは外部記憶装置のデータ格納域情報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、「剥す」ことにより削除される。

変更後のプログラムが既に仮想領域にロードされている場合は、プロセスが利用している仮想領域を変更後のプログラムがロードされているものに変更し、仮想空間への貼り変えを行う。ロードされていない場合は、新規に仮想領域を生成し、仮想空間への貼り変えを行ってから変更後のプログラムをロードする。

4.2 開始位置の変更

開始処理の変更の様子を図 2 に示す。初期状態に変更する際は、データ部、BSS 部、ユーザスタック部、およびレジスタ群の内容を初期化する。指定された位置に変更する際は、指定された位置における情報をデータ部、BSS 部、ユーザスタック部、およびレジスタ群に複写する。

4.3 動作空間の変更

動作空間の変更の様子を図 3 に示す。同一計算機内の別空間に変更する場合は、仮想領域の貼り変えを行う。他計算機上の空間に変更する場合は、変更先の計算機の仮想空間上にプロセスを生成し、生成したプロセスの実行プログラムと開始位置の変更を行う。その後、変更元の計算機上のプロセスを削除する。

5 おわりに

Tender におけるプロセス変身機能の設計について述べた。残された課題として、プロセス変身機能の実装と評価がある。

参考文献

- [1] 谷口 秀夫, 青木 義則, 後藤 真孝, 村上 大介, 田端 利宏: “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] 田端 利宏, 谷口 秀夫: “プロセス再起動機能の提案と評価”, 情報処理学会コンピュータシステムシンポジウム, Vol.2000, No.13, pp.91-98(2000)
- [3] 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)
- [4] 石井 陽介, 谷口 秀夫: “*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”, 情報処理学会第 62 回全国大会講演論文集 (分冊 1), pp21-22(2001)