

Tender における永続制御機能の設計と実現

5 F - 2

市川 正也 谷口 秀夫 牛島 和夫
九州大学 大学院システム情報科学研究科

1 はじめに

我々は、プログラム構造に重点をおいた **Tender** オペレーティングシステムを開発している[1]。**Tender** では、オペレーティングシステム (以降、OS と略す) が制御し管理する対象を資源と名づけている。文献 [2] では、資源「プレート」を永続化する永続制御部の基本機能について報告した¹。資源「プレート」は、既存の OS のファイルに相当する資源である。しかし、その実体はメモリ空間上に存在し、プログラムからは仮想アドレスで参照できる。このため、永続制御部は、メモリ空間上のデータを外部記憶装置に書き出すことにより、資源「プレート」を永続化している。

本稿では、資源「プロセス」の永続化について述べる。資源「プロセス」の永続化においては、永続化された資源「プレート」を有効に利用することにより、効率的な実現を図った。

2 資源「プロセス」の永続化

2.1 プロセスとメモリ関連資源の関係

Tender のプロセスとメモリ関連資源の関係を図 1 に示す。仮想領域^[3]は、外部記憶装置あるいは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^[3]とは、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。さらに、仮想領域を仮想空間に「貼り付ける」ことにより、プロセスが仮想アドレスでアクセス可能となる。このとき、カーネルモードでのみアクセス可能なものを仮想カーネル空間^[3]、カーネルモード及びユーザモードでアクセス可能なものを仮想ユーザ空間^[3]と名付けた。テキスト部、データ部、BSS 部、ユーザスタック部といったプロセスの実体部分は、仮想ユーザ空間を利用し、プロセスを管理し制御する OS 処理は、仮想カーネル空間を利用する。

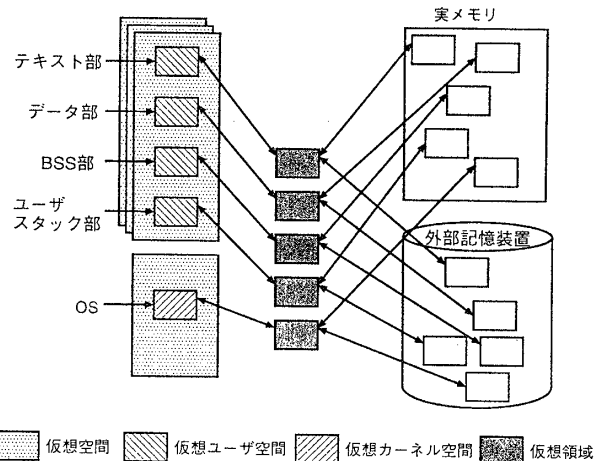


図 1 プロセスとメモリ関連資源の関係

2.2 課題と対処

資源「プロセス」を永続化するためには、以下の部分の永続化が必要である。

- (1) プロセスの実体部分
- (2) プロセスを管理、制御する OS 処理が用いる管理表
- (3) プロセスを管理、制御する OS 処理の手続き部
- (4) プロセスの走行に必要な資源 (例えば、資源「演算」^[4])
- (5) プロセスが利用する資源 (例えば、資源「時計」)

上記の中で、(3) や、(4)(5) の資源を管理し制御する OS 処理の手続き部は、プログラム走行により変更されることはなく、カーネルのロードモジュールの形で外部記憶装置上に存在するため、メモリ上の情報を永続化する必要はない。

先に述べたように、(1) は、資源「仮想ユーザ空間」を利用している。一方、(2) や、(4)(5) の資源の実体部分及び管理表は、資源「仮想カーネル空間」を利用している。したがって、資源「プロセス」を永続化するには、資源「仮想ユーザ空間」と資源「仮想カーネル空間」を永続化する必要がある。

この課題に対処するために、資源「プレート」を用いる。資源「プレート」は、仮想空間上の任意のアド

Design and Implementation of Persistent Control on **Tender**

Masaya ICHIKAWA, Hideo TANIGUCHI and Kazuo USHIJIMA

Graduate School of Information Science and Electrical Engineering, Kyushu University

Email:ichikawa,tani,ushijima@csce.kyushu-u.ac.jp

¹ 論文では、資源「プレート」を、資源「データ」と表記しているが、その後改名した。

レスに貼り付けることができる。このため、資源「仮想ユーザ空間」や資源「仮想カーネル空間」の代わりに資源「プレート」を利用することで上記課題に対処できる。例えば、プロセスの実体部分を永続化する場合、プロセスを生成する際に資源「プレート」を仮想空間上に貼り付け、その上にテキスト部やデータ部などを読み込む。

3 実現方式

資源「仮想ユーザ空間」や資源「仮想カーネル空間」の代わりに資源「プレート」を利用する方法の実現方式として、以下の3つの方式がある。

(方式1) 資源管理処理部を変更

各資源管理処理部の中で、資源「仮想ユーザ空間」や資源「仮想カーネル空間」を利用している部分を、資源「プレート」を利用するよう変更する。

(方式2) 仮想ユーザ空間、仮想カーネル空間管理処理部を変更

永続化の可否を判断し、仮想ユーザ空間や仮想カーネル空間の生成あるいは資源「プレート」の生成を行うように変更する。

(方式3) 資源「プレート」を利用した、仮想ユーザ空間、仮想カーネル空間インタフェース提供処理の実現

資源「プレート」を利用して、仮想ユーザ空間や仮想カーネル空間管理の処理部が提供するインタフェースと同じインタフェースを提供する処理を新たに作成する。

方式1は、全ての資源管理処理部について変更が必要となり、かなりの工数を要する。また、プロセスのプログラム自体が直接資源「仮想ユーザ空間」あるいは資源「仮想カーネル空間」を利用する場合は対処できないという問題がある。

方式2は、仮想ユーザ空間と仮想カーネル空間の管理処理部を変更するだけで良い。しかし、次の問題がある。仮想ユーザ空間や仮想カーネル空間の生成処理は、指定された仮想領域を指定された仮想空間に貼り付けることでその処理を行う。そのため、一つの仮想領域を複数の仮想空間に貼り付けることが可能であり、仮想領域の共有が可能になっている。しかし、仮想領域を資源「プレート」で置き換えるこの方式は、その処理が仮想ユーザ空間の管理処理部あるいは仮想カーネル空間の管理処理部に閉じるため、資源「プレート」を仮想ユーザ空間と仮想カーネル空間の間で共有できないという問題がある。

方式3は、仮想ユーザ空間と、仮想カーネル空間の管理処理部のインタフェースをまとめて提供する方式であるため、上記の方式1や方式2のような問題はない。したがって、方式3を採用する。

以上の流れについて、各方式の様子を図2に示す。**Tender**では、全ての資源操作を資源インタフェース制御(RIC)が制御している[5]。このため、方式3の実現において、呼び出し資源管理処理部の変更をRIC内で行え、プロセスのプログラムはまったく意識しなくて良い。

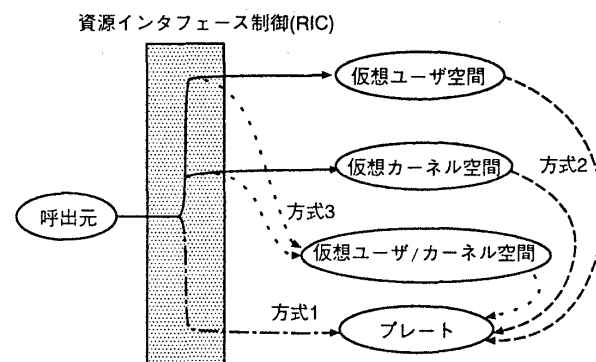


図2 実現方式

4 おわりに

本稿では、**Tender**オペレーティングシステムの永続制御機能について、資源「プロセス」の永続化について述べた。資源「プロセス」の永続化を実現するための課題として、永続化する必要のあるものを明らかにし、課題に対する対処と実現方式について述べた。実現においては、永続化された資源「プレート」を有効に利用することにより、効率的な実現を図った。これにより、プロセスの動作を永続的に保証することが可能になる。

今後は、本稿で述べた機構を**Tender**に実装し、性能評価を行う予定である。

参考文献

- [1] 谷口秀夫：“分散指向永続オペレーティングシステム**Tender**”，情報処理学会コンピュータシステムシンポジウム，シンポジウム論文集 Vol.95, No.7, pp.47-54(1995).
- [2] 市川 正也，谷口 秀夫，牛島 和夫：“**Tender**における永続制御機能”，情処研報，Vol.97, No.56, pp.31-36(1997).
- [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”，情処研報，Vol.98, No.33, pp.87-94(1998).
- [4] 村上 大介，青木 義則，谷口 秀夫，牛島 和夫：“**Tender**における資源「演算」の扱い”，情報処理学会第51回全大，5L-8(1995).
- [5] 後藤 真孝，谷口 秀夫，牛島 和夫：“**Tender**における資源管理方式”，情報処理学会第51回全大，5L-7(1995).