

Tenderにおける効率的なページアウト規則の実現Efficient Page Out Rule in **Tender**

濱口 知之†

Tomoyuki Hamaguchi

山内 利宏†

Toshihiro Yamauchi

谷口 秀夫†

Hideo Taniguchi

1. はじめに

オペレーティングシステム（以降、OS と略す）のメモリ管理機構として、仮想記憶が広く利用されている。仮想記憶をオンデマンドページングで実現する場合、ページアウト規則として LRU 規則を利用することが多い。

Tender オペレーティングシステム（以降、**Tender** と略す）[1]は、データを永続化する手段として、ファイルではなく、資源「プレート」[2]を実現している。プレートは、その内容を仮想記憶によりメモリ上に存在させ、メモリ上の内容を永続化する。この環境では、ページアウト規則として、LRU 規則ではなく、メモリを効率的に扱える規則が必要である。本稿では、プレートの性質を考慮したページアウト規則について述べる。

2. **Tender** オペレーティングシステム

2.1 資源の分離と独立化

Tender では、OS の操作する対象を資源として分離し、独立化している。資源には、資源名と資源識別子を付与し、資源操作のインタフェースを統一している。さらに、各資源を操作するプログラム部品を資源ごとに分離し、共有プログラムを排除している。また、各資源の管理情報も資源ごとに分離し、各資源が他資源の管理表を参照することを禁止している。

Tender におけるメモリ関連資源の関係について、図 1 に示し、以下で説明する。資源「仮想領域」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。資源「実メモリ」は、実メモリ上の領域を資源化したものである。外部記憶装置上の領域の種類として、永続化領域と仮想化領域がある。永続化領域は、既存 OS のファイルシステム領域に相当する。資源「永続ユニット」は、永続化領域の管理単位である。仮想化領域は、メモリ上のデータを一時的に保存するための領域であり、既存 OS のスワップ領域に相当する。資源「仮想ユニット」は、仮想化領域の管理単位である。資源「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アドレス変換表に相当する。資源「仮想ユーザ空間」は、資源「仮想領域」をユーザ用の資源「仮想空間」に貼り付けて生成する。「貼り付け」とは、仮想アドレスを実アドレスに対応付けることである。資源「仮想カーネル空間」は、資源「仮想領域」をカーネル用の資源「仮想空間」に貼り付けて生成する。

2.2 プレート機能

プレート機能は、永続データを仮想記憶空間上で管理し、その領域をすべて外部記憶装置上の永続化領域にもマッピングしている。また、OS や応用プログラム（以降、AP と略す）にメモリ操作に閉じた永続データの操作イン

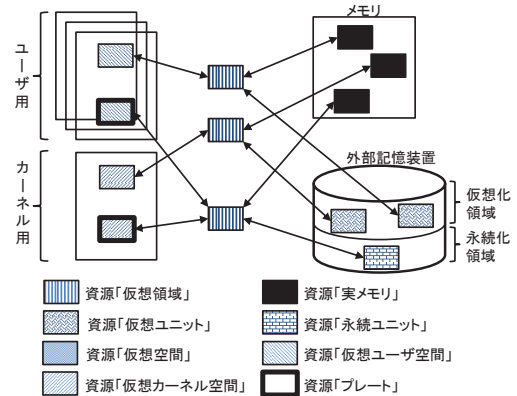


図 1 メモリ関連資源の関係

タフェースを提供する。さらに、プレート機能は、メモリ上の永続データを対応する外部記憶装置上の領域に書き出すことで、OS や AP に永続化を意識せず永続データを管理する。プレートのメタデータなどの管理情報は、仮想空間上のプレート管理表で管理し、既存 OS のファイルシステムと異なり、外部記憶装置上にメタデータを持たせず、外部記憶装置を永続化のためにデータを読み書きする領域として扱う。**Tender** では、プレート機能で永続化される仮想記憶空間上の仮想カーネル空間と仮想ユーザ空間を資源「プレート」として資源化している。

プレートの提供する機能の1つとして、プレートのサイズ変更機能[3]がある。プレートのサイズ変更機能のインタフェースを表 1 に示す。プレートの利便性を向上させるため、プレートのサイズ変更機能は、任意の位置へ任意の大きさの領域の挿入や削除を可能にする。

3. ページアウト規則

3.1 LRU 規則の問題点

ページアウト処理におけるページアウト対象の選定には、LRU 規則を利用することが多い。LRU 規則は、リストを用いて各ページの情報を管理している。リストの更新では、各ページに対応するページテーブルエントリの参照ビットを周期的に確認し、参照ビットの立っていないページをリストの末尾につなぎかえる。これにより、最近アクセスされていないページは、リストの末尾付近に集められる。ページアウト処理では、リストの末尾にあるページから順にページアウトする。

プレートを利用する環境において、LRU 規則を利用すると、以下の2つの問題が発生する可能性がある。

- (問題 1) プレートより先にプロセスのテキスト部やデータ部がページアウト対象となること
- (問題 2) 頻繁にプレートの大きさが拡大される場合、プレートの一部はページアウト対象となりにくいこと

† 岡山大学大学院自然科学研究科, Graduate School of Natural Science and Technology, Okayama University

表1 プレートのサイズ変更機能のインタフェース

形式	機能
<code>resize_plate(plateid, size, offset)</code>	プレート <code>plateid</code> に対して、 <code>size</code> が負の値の場合は、 <code>offset</code> から <code>size</code> 分の領域を削除し、 <code>size</code> が正の値の場合は、 <code>offset</code> から <code>size</code> 分の領域を挿入する。

（問題 1）は、プレートへのアクセス周期がプロセスのテキスト部やデータ部へのアクセス周期より短くなることにより起こる。プレートのページには、一時的にアクセス周期の短くなるものがある。当該ページは、やがてアクセス周期が長くなるため、アクセス周期が長くなったページアウトしてよい。しかし、アクセス周期が長くなった直後は、LRU 規則のリストでは、プロセスのテキスト部やデータ部のページは、当該ページよりも末尾の方にある。このため、プロセスのテキスト部やデータ部のページが当該ページより先にページアウト対象に選定され、ページフォールの発生する可能性が高くなる。

（問題 2）は、プレートへ頻繁にデータを追加し、プレートをページ単位で拡大することにより起こる。プレートの拡大は、プレートのサイズ変更機能を利用し、何らかのデータを追加するために行われる。したがって、拡大後は、拡大によって挿入されたページにアクセスがあり、拡大前にアクセスされていたページへのアクセスは少なくなると予測される。しかし、プレートの拡大速度が大きいと、拡大前にアクセスされていたページの参照ビットは、立ったままとなる。このため、LRU 規則では、当該ページを優先してページアウト対象に選定できない。したがって、メモリを効率的に扱えない。

（問題 1）と（問題 2）の発生する事例として、Web サーバ実行がある。Web サーバのアクセスログは、Web サーバのリクエスト処理ごとにプレートの末尾のページに追加される。プレートの大きさを超える追加には、プレートのサイズ変更機能を利用し、プレートの末尾にページを挿入してプレートを拡大することで対処する。プレートの拡大後は、挿入されたページにアクセスログが追加される。したがって、Web サーバへのアクセスが頻繁に起こると、ページフォールの発生率が高くなることや、適切なページを選定できないことが起こりうる。

3.2 新ページアウト規則

3.1 節で述べた問題は、特定の性質を持つプレートに起因する。そこで、問題の原因となるプレートを選定し、個別にページアウト処理を行うことで対処する。問題の原因となるプレートの特徴は、以下の 2 つである。

（特徴 1）拡大時にプレートの末尾へページを追加する

（特徴 2）大きさが単調増加する

（特徴 1）と（特徴 2）を満たすプレートは、プレートのサイズ変更処理で以下の 2 つの情報を取得し、選定する。

（情報 1）ページの追加された位置

（情報 2）拡大された回数

（情報 1）より、（特徴 1）を満たすプレートか否かを判定できる。また、（情報 2）より、（特徴 2）を満たすプレートか否かを判定できる。プレートの選定後は、選定したプレートのどのページをページアウトするかを選定する必要がある。当該プレートは、（特徴 1）と（特徴 2）を満たすプレートであることから、末尾のページにデータが追加される。そこで、当該プレートの末尾のページ以外から LRU 規則によりページアウト対象を選定する。

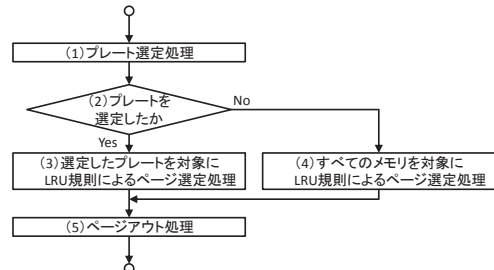


図2 ページアウト対象の選定

以上より、新ページアウト規則は、図2に示す処理流れでページアウト対象を選定する。図2を以下に説明する。

- (1)（情報 1）と（情報 2）を基にプレートを選定する。
- (2) プレートを選定した場合は、選定したプレートを対象にページを選定するため、(3)の処理を行う。選定しなかった場合は、(4)の処理を行う。
- (3) 選定したプレートを対象に末尾のページを除いて LRU 規則によりページを選定する。
- (4) すべてのメモリを対象に LRU 規則によりページを選定する。
- (5) ページアウト処理を行う。

本対処により、問題の原因となるプレートが存在する場合は、プロセスのテキスト部やデータ部ではなく、問題の原因となるプレートのページを必ずページアウト対象に選定可能である。

4. おわりに

Tenderにおいて、プレートの性質を考慮したページアウト規則を述べた。プレートを利用する環境では、LRU 規則を利用すると、プレートより先にプロセスのテキスト部やデータ部がページアウト対象になってしまう。また、頻繁にプレートの大きさが拡大される場合、プレートの一部はページアウト対象になりにくい。そこで、サイズが単調増加し、かつ拡大時にプレートの末尾へページを追加するプレートを対象とする規則を述べた。

残された課題として、本規則を利用したページアウト処理の実現と評価がある。

謝辞 本研究の一部は、科学研究費補助金基盤研究（B）（課題番号：24300008）による。

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] 稲本慎司, 谷口秀夫: **Tender** におけるデータの永続化方式, 情報処理学会研究報告, 2000-OS-85, No.75, pp.101-108 (2000).
- [3] 大本拓実, 田端利宏, 乃村能成, 谷口秀夫: **Tender** における資源「プレート」のサイズ変更機能, 情報処理学会第 67 回全国大会講演論文集, 4M-8, No.1, pp.43-44 (2005).