

Tender における資源「プレート」のサイズ変更機能

大本 拓実[†] 田端 利宏[‡] 乃村 能成[‡] 谷口 秀夫[†]

[†]岡山大学工学部 [‡]九州大学大学院システム情報科学研究院

1. はじめに

計算機は、プロセス管理、メモリ管理などのオペレーティングシステム（以下、OS）の動作に必要なさまざまな管理情報を、揮発性メモリ上に保持している。しかし、計算機が何らかの原因で電源切断された場合、揮発性メモリ上のデータは消失されてしまう。このため、電源再投入後に揮発性メモリ上のデータを用いた処理を続行することはできない。

我々は *Tender* オペレーティングシステム（以下、*Tender*）において、仮想記憶空間上のデータを永続化するプレート機能[1]を用いて計算機処理の永続化の実現を目指している。永続化すべきデータは流動的に変化するため、プレートのサイズを変更する機能が必要である。ここでは、プレートのサイズ変更機能について述べる。

2. プレート機能

2.1 *Tender* オペレーティングシステム

Tender では、OS が制御し管理する対象を資源と呼ぶ。資源の種類ごとに、当該資源を管理する管理表と、当該資源を制御するプログラムがある。また、各資源は資源識別子と資源名で識別される。

Tender におけるメモリ関連の資源の関係を図 1 に示し、以下に説明する。資源「仮想空間」とは、特定のアドレス領域を持つ仮想的な空間であり、仮想アドレスから実アドレスへの変換表に相当する。資源「仮想領域」とはメモリイメージを仮想化した領域であり、その実体は実メモリまたは外部記憶装置上に存在する。資源「永続ユニット」とは、外部記憶装置上の永続化領域の管理単位である。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「仮想ユニット」とは、外部記憶装置上の仮想化領域の管理単位である。仮想化領域は、既存 OS のスワップ領域に相当する。資源「仮想ユーザ空間」は、仮想領域をユーザ用の仮想空間に貼り付けることで生成される。「貼り付ける」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、仮想領域をカーネル用の

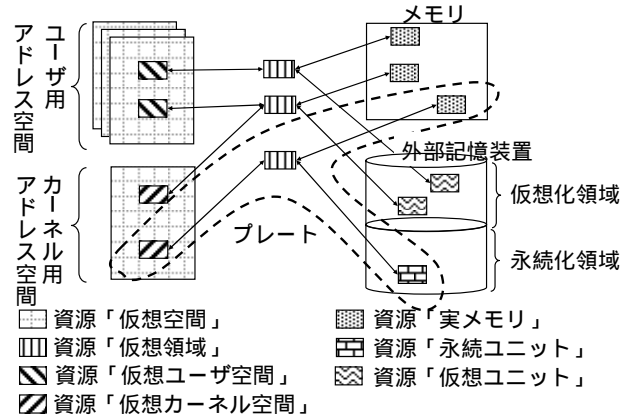


図1 メモリ関連の資源の関係

仮想空間に貼り付けることで生成される。

2.2 資源「プレート」

プレートは、揮発性メモリに不揮発性を付与するため、仮想記憶機構を利用して実現される。従来の仮想記憶機構は、プログラムの実行に不足する実メモリを補うために、外部記憶装置を用いてメモリ空間を大きく見せる機構である。一方、プレートを支える仮想記憶機構は、メモリを不揮発性記憶媒体のように見せるために動作する。プレートの内容はメモリ上に存在し、そのメモリの内容と永続ユニットの同期を取る。つまり、揮発性記憶媒体であるメモリと不揮発性記憶媒体である永続ユニットとの組み合わせにより、プレートは利用者から不揮発性記憶媒体として利用できる。

プレートでは、メモリと外部記憶装置間のデータの授受をすべてOSが行う。このため、利用者は、データ処理をメモリ操作に閉じて行うことができ、データの永続化のためのファイル操作は不要である。

3. プレートのサイズ変更機能

3.1 機能仕様

プレートは、揮発性メモリに不揮発性を付与する。メモリとしての利便性の良さを確保するため、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。

3.2 サイズ変更方法

プレートのサイズを変更するには、当該プレートに対応するメモリと永続ユニットに領域の挿入と削除の機能を実現しなくてはならない。メモリと永続

Resize of Resource "Plate" on *Tender*

[†]Takumi Ohmoto, Yoshinari Nomura and Hideo Taniguchi

[†]Faculty of Engineering, Okayama University

[‡]Toshihiro Tabata

[‡]Faculty of Information Science and Electrical Engineering, Kyushu University

表 1 挿入や削除方法

種類	位置	大きさ	対処
1	管理単位の境界	管理単位の整数倍	指定した位置に挿入や削除
2	管理単位の境界	任意	末尾に挿入や削除&複写
3	任意	管理単位の整数倍	指定した位置に挿入や削除&複写
4	任意	任意	末尾に挿入や削除&複写

ユニットは固定長の管理単位(メモリはページ単位、永続ユニットはデータブロック単位)で管理される。このため、領域を挿入や削除する位置が管理単位境界か否か、挿入や削除する大きさが管理単位の整数倍か否かにより、表 1 に示す 4 種類に分類できる。種類 1 は、領域の挿入や削除のみでサイズ変更を実現できる。種類 2, 3, 4 では複写を行うが、種類 3 は、指定した位置へ領域を挿入や削除することにより、複写は挿入や削除する位置の管理単位の領域のみでよい。種類 2 と 4 は、領域の挿入や削除を行うメモリまたは永続ユニットの末尾へ領域を挿入や削除する。このため、領域を挿入や削除する位置以降の内容を複写し、データをずらさなくてはならない。この場合、領域の挿入や削除を末尾ではなく、指定した位置へ行っても、複写すべきデータ量は変化しないので、処理統一のため、挿入や削除する位置に関わらず、末尾へ領域の挿入や削除を行う。なお、永続ユニットに関しては、ディスク I/O 時間が大きいので、複写には留意が必要である。

3.3 同期方法

現在の *Tender* ver10.0 では、プレートは、永続ユニット上のデータに対応するメモリを全ページ保持している。このため、メモリと永続ユニットの同期を取る方法として、3.2 節述べたような永続ユニット内の複写を行わず、メモリの内容を永続ユニットに書き出すことにより、メモリと永続ユニットの同期を取ることにした。

3.4 同期契機

サイズ変更処理によって複写を行う場合、メモリと永続ユニットの同期の必要性が生じる。つまり、プレートのサイズ変更は、メモリと永続ユニットの同期契機のひとつと考えられる。

サイズ変更処理では、即座に同期を取る場合(同期方式)と取らない場合(非同期方式)がある。同期方式では、挿入や削除する位置以降のメモリの内容を即座に永続ユニットに書き出す。非同期方式では、メモリと永続ユニットの同期をすぐには行わない。両方式の比較を表 2 に示し、以下に説明する。処理時間とは、プレートのサイズ変更処理に要する時間のことである。永続性とは、突然の電源切断の

際に、電源再投入後、処理に復帰できるかどうかである。同期方式と非同期方式は処理時間と永続性が逆の関係になる。

表 2 同期と非同期の比較

方式	処理時間	永続性
同期	長い	高い
非同期	短い	低い

サイズ変更処理には、領域の挿入と削除の 2 種類がある。その 2 種類について、各方式との関係を考察する。

領域を挿入する場合、何らかのデータを追加するために領域を挿入するので、挿入する領域は直後に更新されると予測される。このため、同期方式の場合、挿入する領域は、サイズ変更処理のため複写を行う場合と更新後の 2 度に渡って、永続ユニットと同期を取るようになってしまう。したがって、領域を挿入する場合、非同期方式が好ましいといえる。

一方、領域を削除する場合、不必要なデータを削除するために領域の削除を行うので、直後に更新される可能性は低いと予測される。したがって、領域を削除する場合、同期方式が好ましいといえる。

4. おわりに

Tender における資源「プレート」のサイズ変更機能について述べた。本機能では、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。これにより、プレートは、メモリとしての利便性の良さを確保できる。今後は、プレート上で更新のあった部分のみを永続ユニットに書き出す部分書き出し機能、メモリと永続ユニットの同期の契機を検討する予定である。

参考文献

- [1] 野司, 田端利宏, 谷口秀夫: "Tender におけるプレート機能を利用した資源の永続化", FIT(情報科学技術フォーラム)2002, B-46, pp.179-180 (2002)。