

# アクセス保護機能を持つバッファ管理方式

4F-11

箱守 聡 谷口 秀夫

NTTデータ通信(株) 開発本部

## 1. はじめに

オペレーティング・システム(以降、OSと略す)が行なう処理時間の多くは、入出力の時間が占めている。そのため、例えば、メモリ上の情報(データ)を通信回線や磁気ディスク(DK)に効率良く出力することが求められている。

最近では、入出力にDMAC(Direct Memory Access Controller)を利用することが多い。DMACが一度に扱えるデータは、実メモリ上の連続領域である。そのため、論理メモリ空間をプロセスに提供しているOSは、プロセスが要求した入出力のデータについて、実メモリ上の連続領域毎に繰り返しDMACを起動している。したがって、入出力処理は複雑になっている。このため、いくつかのOSでは、論理メモリ空間の領域に合わせて実メモリ上も連続領域とするバッファプール機能が実現されている。一方、最新のDMACは、実メモリ上の異なる複数の領域をまとめて操作することができる(チェーン機能と呼ばれることがある)。これにより、入出力の実行処理は高速になっているが、チェーン機能を利用するための処理がOSに新たに必要になっている。

本稿では、従来のOSがデータ格納用として提供しているバッファプール機能などについて問題点を明かにする。さらに、これらの問題点を解決する方式を提案し、その基本構造や特徴について述べる。

## 2. 従来方式

### 2.1 方式概要

プロセスが入出力データを置くことができる論理メモリ空間として、大きく次の2つに分類できる。1つは、データ部、スタック部、およ

び共有メモリである。もう1つは、バッファプールである。これらについて、メモリマッピング表により実メモリを論理メモリ空間に割り付けた様子を図1に示し、以下に説明する。

データ部、スタック部、および共有メモリは、プロセス毎の論理メモリ空間に割り付けられている。共有メモリは、プロセス間で共有できるように、対応する実メモリを複数の論理メモリ空間に割り付けている。これらのものは、実メモリの基本管理単位(以降、ページと呼ぶ)の集合をメモリマッピング表により論理メモリ空間の連続領域に割り付けている。

バッファプールは、プロセスに共通な論理メモリ空間に割り付けられている。実メモリ上でのアドレス連続性が保証されている。

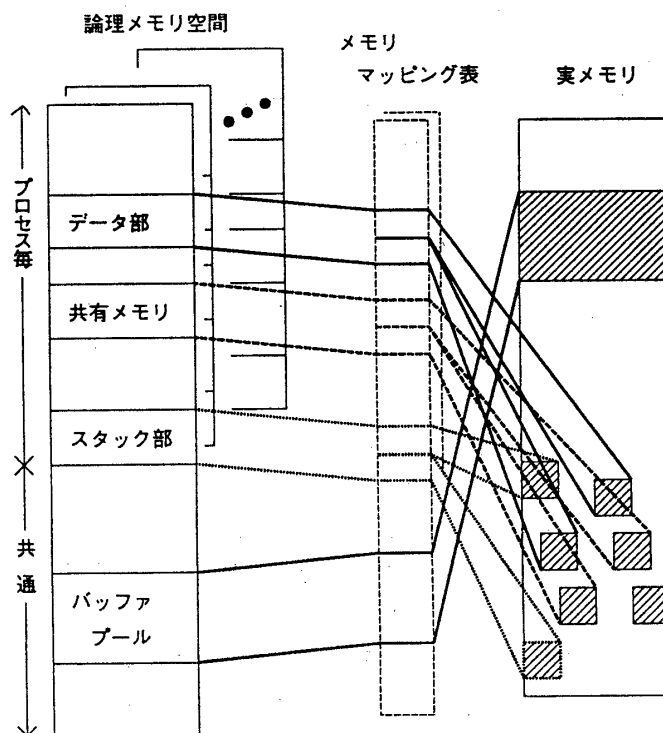


図1 従来方式

## 2. 2 問題点

データ部、スタック部、および共有メモリは、プロセス毎の論理メモリ空間に割り付けられているため、プロセス間での不当アクセスを保護できるものの、

[問題点1] 論理メモリ空間の領域に割り付けられた実メモリが連続でない、

という問題がある。このため、次のようなことになってしまう。

(1) 論理メモリ空間のアドレスと実メモリのアドレスの間の変換処理が複雑である。

(2) プロセスからの1つの入出力要求に基づき、複数回の入出力動作が必要になる。そのため、入出力処理が遅い。

一方、バッファプールは、論理メモリ空間の領域に割り付けられた実メモリは連続

であるものの、

[問題点2] プロセス間の不当アクセスを防止できない、

という問題がある。このため、次のようなことになってしまう。

(1) サービス処理の誤動作の発見が遅れる。  
(2) 誤動作時の原因追及が難しい。

## 3. 提案方式

### 3. 1 基本構造

提案する方式は、2. 1節で述べたバッファプールを基に改善したものである。提案方式について、実メモリと論理メモリ空間の関係を図2に示し、その基本的な対処を以下に説明する。

[対処1] 1つのバッファプールの大きさをページ長の整数倍とする。これにより、各バッファプール単位で、必要なプロセスの論理メモリ空間に割り付けることを可能にする。

[対処2] プロセス毎の論理メモリ空間だけでなく、OSの論理メモリ空間にも割り付ける。

[対処3] メモリマッピング表により、複数のプロセス毎の論理メモリ空間に割り付け可能とする。

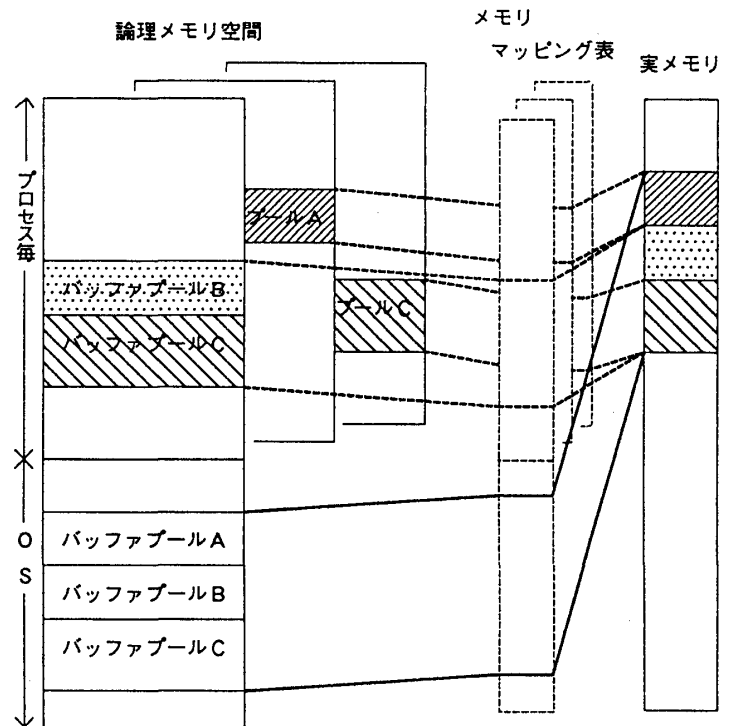


図2 提案方式

### 3. 2 特徴

提案方式は、3. 1節に述べた基本構造であるため、以下の特徴を持つ。

- (1) 2. 1節で述べたバッファプールを基にしているため、実メモリ上でのアドレス連続性が保証できる。つまり、[問題点1]を解決できる。
- (2) [対処1]により、論理メモリ空間を共有しないプロセス間での不当アクセスを防止できる。つまり、[問題点2]を解決できる。
- (3) [対処2]により、OS内でのバッファプール操作が容易である。
- (4) [対処3]により、プロセス間でバッファプールを共有できる。

## 4. おわりに

1つのバッファプールの大きさをページの整数倍とし、実メモリ上でアドレスが連続な領域をプロセス毎の論理メモリ空間に割り付けるバッファ管理方式を提案した。本方式は、プロセス間での不当アクセスを防止でき、かつ入出力処理が速い特徴を有する。