

前回:  
キャッシュ

今回:  
仮想記憶  
ハードディスクと主記憶を組み合わせて、主記憶を拡張する

キーワード:  
仮想アドレス, 物理アドレス, アドレス変換, ページ

---

## 第13回 仮想記憶

### 9. 仮想記憶 [尾内] p127

#### 9.1 ソフトウェア側からの要求

9.1.1 複数のプログラムの同時実行  
ひとつのCPUを時間的に切り替えながら実行する  
(最近マルチプロセッサ, マルチコアもあるが)

複数のプログラムがメモリ上に存在する必要

プログラムは固定番地に置くようにコンパイルされるのが普通  
サブルーチンの入口番地 (呼出し命令の飛び先番地)  
大域変数の番地 (ロード/ストア命令の番地)  
なども決まった番地になる

複数のプログラムが存在すると番地が競合

9.1.2 実メモリの不足  
メモリを大量に必要とするプログラム  
そうでなくても多数のプログラムの実行でメモリ不足

9.1.3 フラグメンテーション fragmentation  
メモリの割り当てと解放を繰り返すと  
「空き領域」が細かく分散してしまう

まとまった大きな領域がとれなくなる

9.1.4 同一コードの共有  
同一プログラムを複数実行する場合

プログラムの命令部分は書き換わらない  
-> 同じプログラムを実行するなら共有可能  
(データは書き換わるから共有不可)  
プログラムの命令部分はメモリ上に一組だけ置くことにしてメモリ節約

同一プログラムでなくても, 共通するライブラリ関数を呼ぶことがある  
-> 共通ライブラリ(の命令部分)を共有

これらの要求を解決するために仮想記憶方式 (virtual memory)

#### 9.2 しくみ

9.2.1 アドレス空間の分離  
仮想アドレス (論理アドレス) virtual address  
CPUが指定するアドレス

物理アドレス (実アドレス) physical address  
主記憶に付けられたアドレス

アドレス変換の機構 (図7.20, 7.21)  
CPUの指定する仮想アドレスを主記憶の物理アドレスに変換して  
主記憶にアクセスする  
「変換表」

ページ  
主記憶の(固定長の)単位  
サイズは4KBから256KB

この単位でアドレス変換

<アドレス変換の効果>

変換表をプログラムごとに持てば番地の競合は解決  
実行するプログラムを切り替える際に変換表も切替

変換表を適切に設定すると、プログラム間でのメモリの共有が可能  
プログラムの命令部分の共有  
データの共有も可能(書き換わるので注意が必要)

### 9.2.2 主記憶と二次記憶(ハードディスク)

仮想アドレス空間を二次記憶上にも置く  
必要な部分だけ主記憶に持つてくる

主記憶をキャッシュとして使うことに相当

<効果>

主記憶の不足は解消

というわけで、ページ方式の仮想記憶について説明  
(ほかにセグメント方式というのもある)

### 9.3 キャッシュと仮想記憶の対応関係

キャッシュ	仮想記憶
-----	-----
キャッシュメモリ	主記憶
主記憶	二次記憶(ハードディスク)
ブロック(ライン)	ページ
「ミス」	「ページフォールト」「ページエラー」

### 9.4 仮想アドレスと物理アドレスの対応(図7.21)

仮想アドレスの分割

下位ビット ページ内オフセット (page offset)  
上位ビット 仮想ページ番号 (virtual page number)

物理アドレスの分割

下位ビット ページ内オフセット (page offset)  
上位ビット 物理ページ番号 (physical page number)

仮想ページ番号のビット数 > 物理ページ番号のビット数

(20 > 18)

仮想アドレス空間の方が広い

32ビット仮想アドレス空間 - 4GB

<しかし>

最近は4GB以上の主記憶も普通

そこで64ビットの仮想アドレス空間に移行

Windows 7 (x64) - 64ビット対応のOSの例

### 9.5 ページ表(変換表) 図7.22

仮想ページ番号 - 物理ページ番号 の対応表

仮想ページ番号でインデックス(添字として使う)

主記憶上に置く

プロセス(プログラムの個々の実行)ごとに異なるページ表をもつ

プロセス切り替えのときにページ表も切り替える

ページ表の先頭番地を指す特別なレジスタが存在(ページ表レジスタ)

制御用情報

有効ビット (対応する物理ページが主記憶上にあるか)

ダーティビット (書き換えが行われたか)

アドレス変換の処理

ページ表を参照する

有効ビットが1だったら(物理アドレスに変換して)主記憶を読み書き

有効ビットが0だったらページフォールト

#### 9.5.1 仮想記憶によるメモリの保護

プロセスごとに独立した仮想アドレス空間を持つ  
他のプロセスのメモリへのアクセスを禁止

ページ単位でアクセス制限可能(書き込みアクセスビット)  
命令領域への書き込みの防止に有効

#### 9.6 ページフォールト

物理ページは主記憶ではなく二次記憶に存在する  
(どこにもない場合もあり)

二次記憶に存在する場合の処理

二次記憶から主記憶にコピー(ページ単位)

主記憶にスペースがなければ

追い出して置き換え

置き換えには時間がかかる

なるべくページフォールトが起きないように工夫が必要

#### 9.7 書き込み

ライトスルーかライトバックか  
(キャッシュの項で説明済み)

二次記憶は非常に遅い

ライトバックを採用

置き換えのときに書き出し

ダーティビットが1のときのみ