



仮想アドレスと物理アドレスの変換

アドレス変換

address traslation

メモリーを参照する度に生じるのでソフトウェアによる対応では遅すぎる

多くのCPUがアドレス変換などを行うためのメモリー管理用のハードウェアを備えている

このハードウェアをメモリー管理ユニット, MMU (Memory Management Unit)

という

MMUはページテーブルと呼ばれる仮想アドレスと物理アドレスの変換表を使って処理を行う

ページテーブルの実現法とハーディング方式

仮想アドレスの1つ1つに物理アドレスが紐づけられるが、個々にテーブル内に確保するとなると、仮想アドレスの数だけテーブルのエントリが必要となる

↳ テーブルが大きくなりすぎる

仮想アドレス空間上の連続した固定長のメモリー領域をひとまとめにし、この領域に対して、連続した物理空間上のメモリー領域を割りあつるようにする

↳ この連続したひとまとめの領域を ページ と呼ぶ

仮想アドレス空間上のページを仮想ページ or 論理ページと呼ぶ

物理アドレス空間上のページを物理ページ, ページフレーム, フレームなどと呼ぶ

ページの大きさは4キロバイトや8キロバイト程度が多い

## 仮想アドレスから物理アドレスへの変換手順の例

ページサイズを  $2^{12}$  バイト = 4KB とする

仮想アドレス, 物理アドレス 共に 32 ビット幅とする

- 1) 32 ビットの仮想アドレスの上位 20 ビットが仮想ページの番号に相当する  
( $= 32 - 12$ )
- 2) MMU はページテーブルを参照し, 問い合わせのあった仮想ページに対応する物理ページ番号を得る. 物理ページ番号は物理アドレスの上位 20 ビットである
- 3) 仮想アドレスの下位 12 ビットをそのまま物理アドレスの下位 12 ビットとして使用すれば, 物理アドレスが得られる

通常, ページテーブルは物理メモリ上に置かれているため, 実行効率が低い

そこで, CPU は ページテーブルのエンTRIES をキャッシュする TLB (table look-aside buffer)

という特殊なハードウェアが搭載されている

TLB は 100 エントリ程度の連想記憶で, 最近参照した ページテーブルのエンTRIES を保持している.

ある仮想アドレスを参照したときに, TLB 内に対応する物理ページのエンTRIES があれば, 高速にアドレス変換を行うことができる

上の手順で問題が生じる場合はないか?

2) において, 対応する物理ページが見つからない場合がある

↳ ミッシング不在という

TLB エントリはキャッシュエントリであり、メモリには影響が  
ない。

ミッシング不在とは仮想ページに物理ページが割り当てられていない状態を指す



マッピング不在の仮想ページに対してアドレス変換を要求した場合、

ページフォルトが生ずる

ページフォルトは割込みの一種である例外で、オペレーティングシステムが処理する

不正なメモリアドレスの場合には、segmentation error  
とすることがある

仮想アドレス空間を用いた方式の場合、不連続な物理メモリ領域を合わせて使う  
ことができるので、fragmentationにより、大きな領域がとれなくなるという問題はな

たが、ページ方式では、ページ毎にメモリを管理するため、ごく少量のメモリ  
の使用しない場合には、ページ内の大部分がムダになる場合が生ずる

→ 内部断片化という

ページサイズを小さくすれば回避できるが、その場合はページテーブルが  
大きくなるなどのメモリ管理上のオーバーヘッドが小さくなる

スワップ等  
large page

## 6-6 ページの保護

各仮想ページに対して      1) 読出し (read)      2) 書込み (write)  
3) 実行 (execute)

という属性を指定できるようになっている

テキスト領域は read と execute を許可 (write は禁止)

データ領域は read と write を許可 (execute は禁止)

スタック領域は全て許可

## MMU (メモリ管理ユニット) の動作

1) ページテーブルを参照

マッピング不存在の場合はページフォルトを発生

2) マッピング不存在の場合は、ページテーブルのエンタリに記載されている保護属性を参照し、要求された操作が許可されているか確認

3) 要求された操作が許可されている場合は、例外 (protection fault) を  
メモリ保護違反 発生

4) 要求された操作が許可されている場合は、仮想アドレスを物理アドレスに変換し処理を行う。

## 6.7 仮想記憶

仮想記憶：メモリ量という物理的な制約を取り除くための資源の  
抽象化技術

仮想記憶では二次記憶（通常ハードディスク）をあたかも物理メモリの一部として扱い、物理的に搭載されたメモリ量を越えるメモリを利用可能とする

物理メモリにないデータは、二次記憶上のスワップ (swap) 領域とよばれる領域に書き出される

スワップ領域への書き出しは「ページ単位」で行われる



## 仮想記憶の動作の概要

(スワップアウトとスワップイン)

↑

1) 物理メモリに入り切らないページは、その内容をスワップ領域に書き出す。  
スワップ領域に書き出されたページは、マッピング不在としてみる。

2) メモリ参照の際に参照された仮想ページがマッピング不在であれば、物理メモリを参照する。

3) マッピング不在の場合、MMUによりページフォルトが起る。  
その結果、OSにより以下の処理が行われる。

a) 参照された仮想ページがスワップ領域に書き出されているかを確認する。  
もし、参照された仮想ページが書き出されていない場合は、不正なメモリ参照と判断

→ プロセスの停止、スクリーンション違反

b) 参照された仮想ページがスワップ領域に存在すれば、その内容を物理メモリ上に読み込む。(スワップイン)  
読み込みが終了した時点で、仮想ページと物理ページのマッピングをページテーブルに登録し、メモリ参照を行ったプロセスに制御を戻す

OSが各仮想ページの「状態」を管理する必要がある

各仮想ページについて、スワップ領域に書き出されているかどうか、  
書き出されている場合、スワップ領域のどこにあるか

これらの情報をアドレス空間記述表 として保持

ページテーブルは MMU (ハードウェア) が参照し、メモリ参照の際に常に参照される  
or TLB

アドレス空間記述表はページフォルトが起った際に OS が参照する。