

計算機方式論

第11章 ページング方式

-その1-

1

仮想記憶方式

- 原理的には、次の2つを基とした4つの方式がある。

ページング方式

多重レベルページング方式

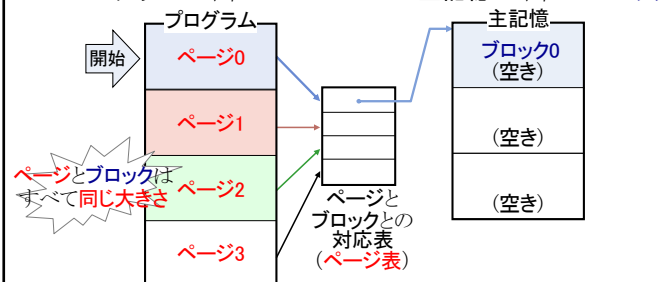
セグメンテーション方式

セグメンテーション-ページング方式

2

ページング方式 -ブロックの大きさ

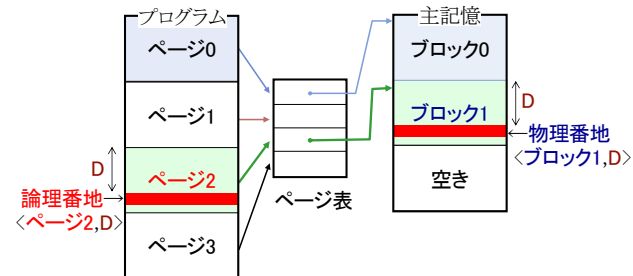
- プログラム (仮想記憶) と主記憶を一定の大きさで区切った分散固定区画割当方式をとる。
- プログラムの単位…ページ



- プログラムの実行は、最初に必要なページを主記憶上の空ブロックにロードし、表でつなぎ、そのページから実行を開始。

3

ページング方式-ページ表

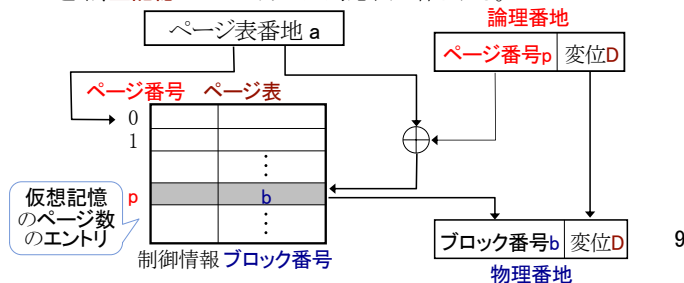


- ページが必要となる度にページ不在の割り込みが生じ、そのページを主記憶の空ブロックにロードする
…デマンドページング (要求時ページング, demand paging)
- ページ表を使って、番地変換を行う。

4

直接写像方式 (direct mapping, index access)

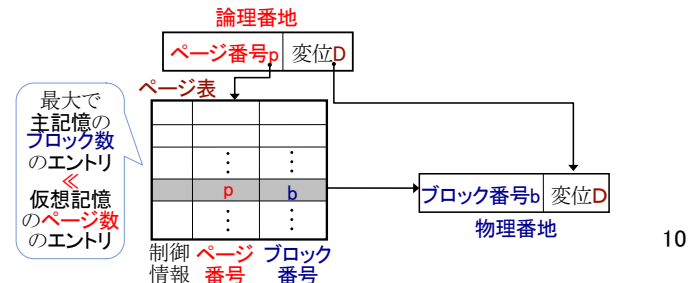
- ページ番号をインデックスとした表を作り、各エントリにそのページに対応するブロック番号をいれる。そのページが主記憶に存在しなければ、そのエントリは‘空白’にしておく。
⇒
表のサーチは速いが、ページ番号の種類だけエントリが必要！
通常、主記憶上にこの方式の対応表が作られる。



9

連想写像方式 (associative mapping)

- ページ番号pのページとブロック番号bのブロックとが対応するとき、それらの対<p,b>をページ表に登録する。
↓
表のサーチの時間はかかるが、高々ブロック番号の種類だけのエントリで済む。高速な記憶装置を採用して、ペーシング方式やセグメンテーション方式の連想記憶装置に用いられる。



10

連想記憶装置

- ペーシング方式の番地変換…マッピング方式
データや命令をアクセスする度にページ表をひく
ページ表は主記憶にとられるので、
番地変換のために主記憶アクセスが必要
1命令の実行に、マッピング方式をとらない場合に比べ、
倍以上の主記憶アクセスを要する…ノイマンの隘路！
⇒ 連想記憶装置
ページ表の一部を格納し、高速アクセスできる。
連想写像方式で番地変換。
- IBM System/370では、TLB (Translation Lookaside Buffer) とよばれた。

11

連想記憶による番地変換

- 論理番地のページ番号で連想記憶をひく。
 - ページ番号が登録されていれば、
対応するブロック番号を求め、物理番地を得る。
 - 連想記憶にページ番号が登録されていなければ、
ページ表をひき、表にブロック番号が登録されていれば、
物理番地を得るとともに、
連想記憶に対<ページ番号, ブロック番号>を登録する。
 - ページ表にブロック番号が登録されていなければ、ページ切れ割込で、ページを主記憶にロードし、ページ表にブロック番号を登録し、物理番地を得るとともに、
連想記憶に対<ページ番号, ブロック番号>を登録する。
- (長所) 高速の番地変換。
(短所) 高価なため、小容量。
⇒ 連想記憶装置の内容は、古いものから捨てる。

12

