

計算機方式論

第11章 オーバレイ処理/ 再配置方式/ 仮想記憶方式

1

オーバレイ処理と再配置方式

- 一度に主記憶に納まりきらないプログラムの実行

オーバレイ処理方式

再配置方式

レジスタ方式

マッピング方式

2

オーバレイ処理方式

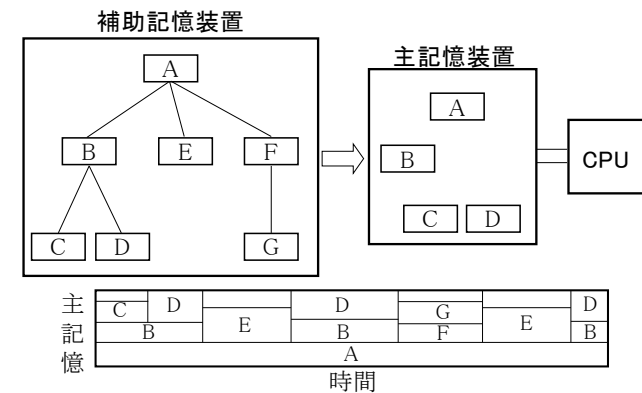
- (1)プログラムを予め複数の部分に分けて作成
(各部分<主記憶)。
- (2)プログラムの進行に応じて、どの段階でどの部分を
主記憶のどこにロードするかを決める
…ユーザが全てプログラムで指定



多重プログラミングシステムでは事実上不可能!!

3

オーバレイ処理方式



4

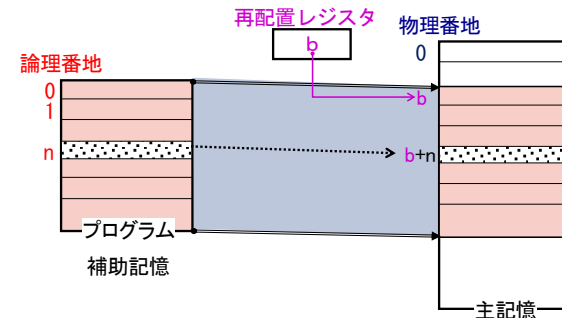
再配置方式

- 多重プログラミングシステムでの主記憶割当て:
プログラムの実行開始時に主記憶を割当て、プログラムの実行途中でも、再度主記憶が割当てられる方式。
⇒ プログラムの作成・翻訳・結合時において、プログラム実行時の主記憶の番地(物理番地)を指定できない。
- プログラム内での番地は、物理番地とは独立に割り当てる
… 論理番地 または 再配置可能番地 という
↓
実行時に、論理番地を物理番地に変換する(再配置する)
ハードウェアが必要…再配置(リロケーション)機構
- 再配置方式
レジスタ方式…レジスタを使う
マッピング方式…表を使う

5

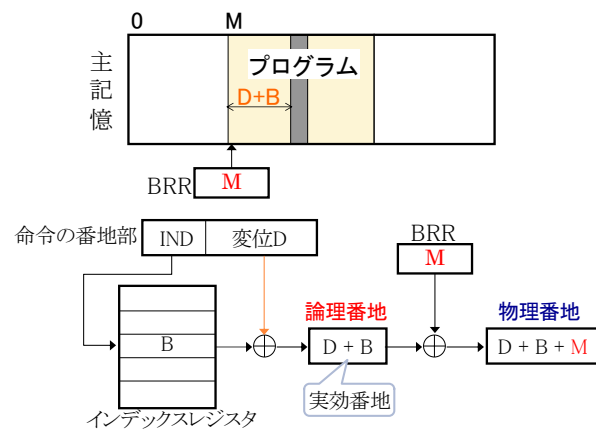
レジスタ方式

- 一個または複数のレジスタ(再配置レジスタ)に、プログラムまたはその一部の開始番地をセットして、実行時に番地を再配置レジスタで修飾する。



6

レジスタ方式の例-NEAC2200



7

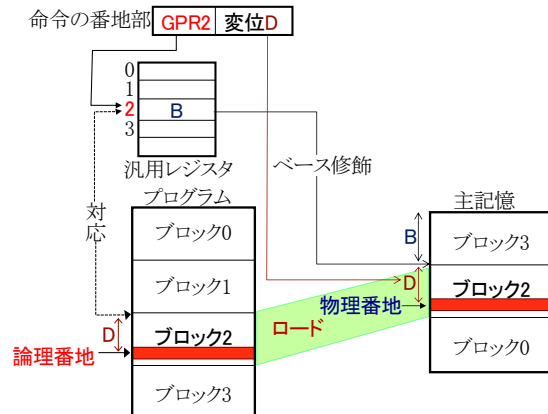
レジスタ方式の例-IBMシステム360

- プログラム…ブロック(4KB単位)に分割
16個の汎用レジスタ ← 対応 → ブロック番号
論理番地 = ブロック内の相対番地 D
+
ブロックに対応する汎用レジスタ n
- 主記憶…ブロック(4KB単位)に分割
実行時:
プログラムのブロック 主記憶上の開始番地 B
↓ ロード ↓ セット
主記憶の空ブロック 対応する汎用レジスタ n
物理番地 = ブロック内の相対番地 D
+
汎用レジスタ n の値 B

8

レジスタ方式の例-IBMシステム360

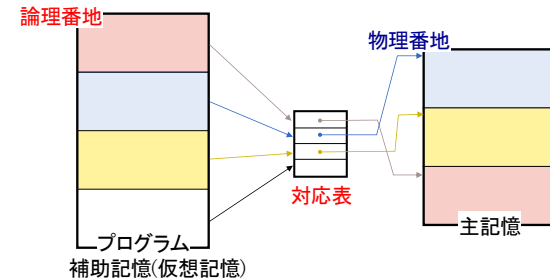
分散固定区画割当



9

マッピング方式

- プログラムをページ/セグメントとよばれる部分に分ける。ページ/セグメント単位に物理番地と論理番地の対応表を作る。
- 論理番地をアクセスする度に、この対応表を調べ、論理番地を物理番地に変換、対応する物理番地をアクセス。
- マッピング方式は、仮想記憶方式と呼ばれている。



10

仮想記憶方式

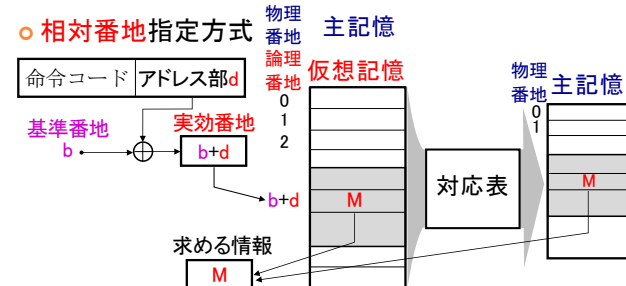
- プログラマは、論理空間(仮想番地空間)を対象としてプログラムを組む。
- 論理空間はブロック(ページ、セグメント)に分割し、ブロック単位で物理空間(実空間)にロードする。
- ブロックの対応づけは、このためのハードウェアとソフトウェアの助けで行なう(マッピング方式)。
- CPUの記憶管理機構(MMU)が、論理番地/物理番地の変換とブロック置換の管理を行う(前出再配置機構)。

[注意] 仮想番地空間は補助記憶装置上で、実空間は主記憶装置上で実現される。

11

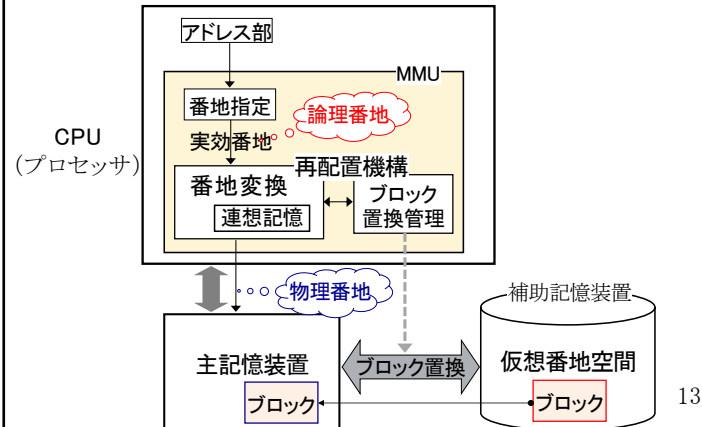
仮想記憶の番地指定

- 「命令形式」の章の、直接番地/間接番地指定、絶対番地/相対番地指定では、主記憶番地(物理番地)を指定したが、仮想記憶方式を採用したシステムでは、仮想記憶の番地(論理番地)を指定する。



12

記憶管理機構(MMU)



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

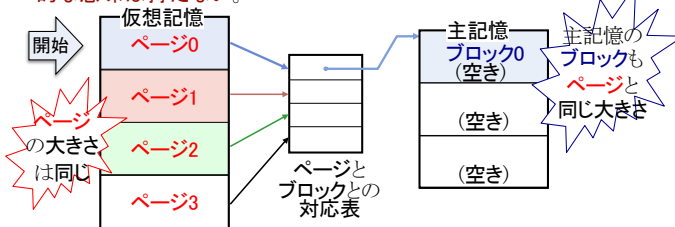
セグメンテーション方式

セグメンテーション-ページング方式

14

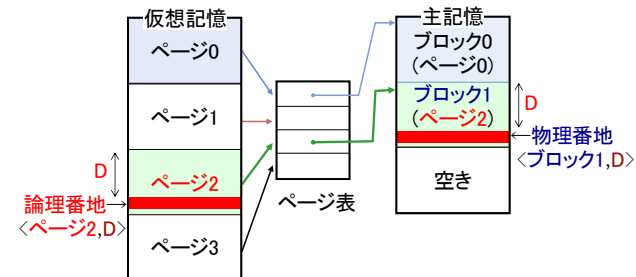
ページング方式-ページとブロック

- 仮想記憶(論理空間)を固定長分割し、ページとよぶ。主記憶(物理空間)もページと同じ大きさで区切り、ブロックとよぶ。分散固定区画割当方式を採用。
- 仮想記憶上にロードされたプログラムの各ページは、ロードモジュールを一定のページ長で分割したものにならず、論理的な意味を持たない。



- 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。

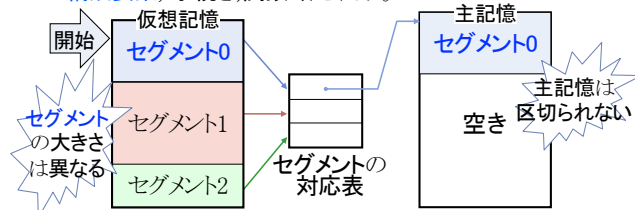
ページング方式-番地変換



- 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)

セグメンテーション方式-セグメント

- 仮想記憶(論理空間)を**可変長**で分割し、**セグメント**とよぶ分散**可変**区画割当方式をとる。
- 仮想記憶上にロードされたプログラムの各セグメントは、言語のクラスやモジュールに相当し、**論理的にまとまったもの(プログラムの構成要素; 手続き,関数等)**を表す。



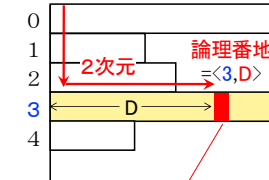
- 主記憶にない**セグメント**が必要になったら、**セグメント不在**の割り込みが生じ、制御プログラムがそのセグメントを主記憶にロードし、表で**接続**する。

17

セグメンテーション方式-2次元番地

- 仮想記憶(プログラム)をセグメントに分割し、**論理番地**=<セグメント番号,セグメント内番地>
... **2次元番地**:セグメント単位に番地付け

セグメンテーション方式



論理番地は**セグメント3**のモジュール内の**変位D**に位置する**プログラム構成要素**の**所属**を表す。前後に文など挿入・削除してもこの**論理的関係**は**変わらない**!

ページング方式



論理番地<3,D>= $3 \times \text{ページ長} + D$ 、プログラムを**固定長ページ**で単純に**分割**した結果を表す。文などの挿入・削除で、ページ番号が**変わりうる**!

18