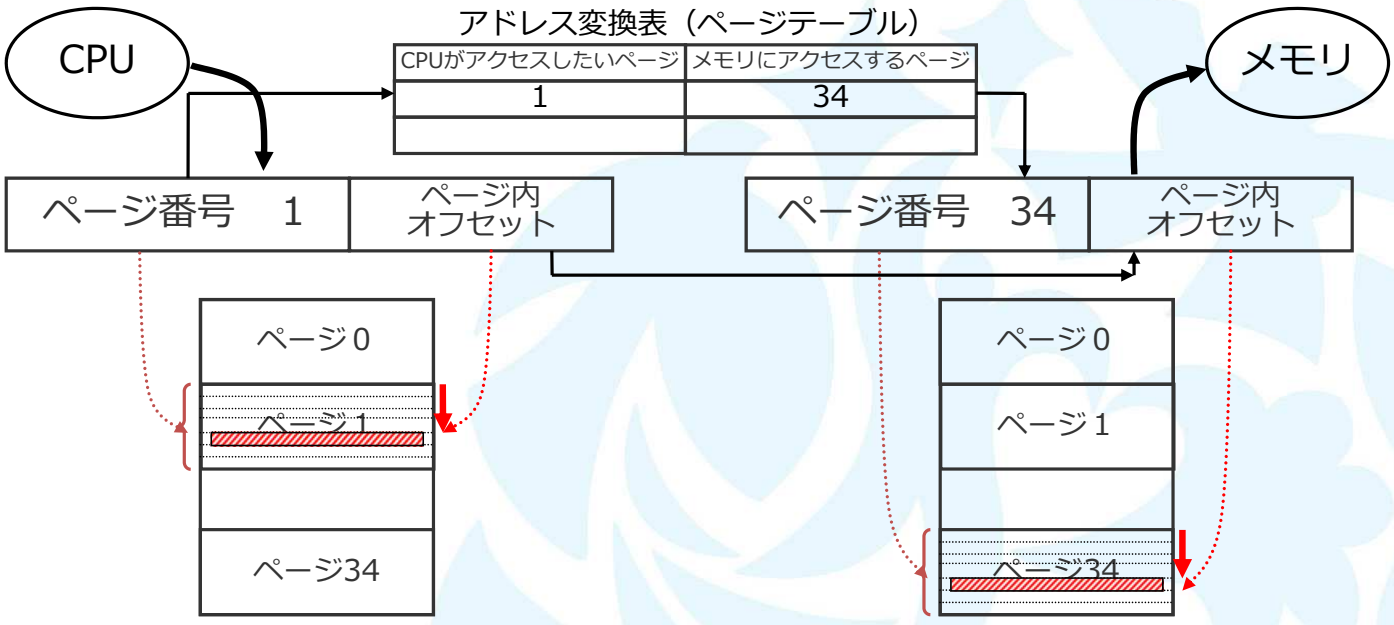
東邦大学

生命の科学で未来をつなぐ

ページングのアドレス変換の性能

東邦大学

復習：ページングの仕組



東邦大学

システムの性能は？

- 一般に、
  余分なことをする ⇒ 余分な時間がかかる ⇒ 性能劣化
- ページングで性能劣化？

システムの性能は？

- 一般に、
  余分なことをする ⇒ 余分な時間がかかる ⇒ 性能劣化
- ページングで性能劣化？
  - メモリをアクセスするたびに、アドレス変換する
    １命令当り、メモリを２回～３回アクセス

東邦大学

システムの性能は？

- 一般に、
  余分なことをする ⇒ 余分な時間がかかる ⇒ 性能劣化
- ページングで性能劣化？
  - メモリをアクセスするたびに、アドレス変換する
    １命令当り、メモリを２回～３回アクセス
- 変換表(ページテーブル)を引く時間が余分にかかる
  - 表は主記憶上に置くのでそのアクセス時間が余分
    主記憶アクセスするのに変換表のために更に１回アクセス
    （表の大きさから考えて、主記憶に置かざるを得ない）

システムの性能は？

- 一般に、
  余分なことをする ⇒ 余分な時間がかかる ⇒ 性能劣化
- ページングで性能劣化？
  - メモリをアクセスするたびに、アドレス変換する
    １命令当り、メモリを２回～３回アクセス
- 変換表(ページテーブル)を引く時間が余分にかかる
  - 表は主記憶上に置くのでそのアクセス時間が余分
    主記憶アクセスするのに変換表のために更に１回アクセス
    （表の大きさから考えて、主記憶に置かざるを得ない）
- さまざまな工夫をして影響を軽減　⇒ ⇒

工夫１　変換表を引く時間？

- 変換表のエントリを一発で見つける (検索しない)

工夫１　変換表を引く時間？

- 変換表のエントリを一発で見つける (検索しない)
  - 変換前のページ番号Pを、変換表の行数(P行目)にする
    P行目 ⇒ 変換出力の場所は(表の先頭+(P×行の長さ))
  - （もし１つ１つ先頭からエントリを探すと
    １行チェックする毎に主記憶読出して時間がかかる）

工夫１　変換表を引く時間？

- 変換表のエントリを一発で見つける (検索しない)
  - 変換前のページ番号Pを、変換表の行数(P行目)にする
    P行目 ⇒ 変換出力の場所は(表の先頭+(P×行の長さ))
  - （もし１つ１つ先頭からエントリを探すと
    １行チェックする毎に主記憶読出して時間がかかる）
  - この方法の代償として、表のエントリ数は(仮想)ページの数だけ必要になる
    ⇒ 後で 仮想ページ数»物理ページ数 にした時不利

工夫２　変換表を「キャッシュ」する

- キャッシュ
  ＝メモリの一部を小さい高速メモリにコピーする
- 変換表の一部を、高速で読出せるメモリにコピー
  - CPU側にある小さな高速メモリにコピーを置く

工夫２　変換表を「キャッシュ」する

- キャッシュ
  ＝メモリの一部を小さい高速メモリにコピーする
- 変換表の一部を、高速で読出せるメモリにコピー
  - CPU側にある小さな高速メモリにコピーを置く
    TLB (Translation Lookaside Buffer) と呼ばれる

工夫２　変換表を「キャッシュ」する

- キャッシュ
  ＝メモリの一部を小さい高速メモリにコピーする
- 変換表の一部を、高速で読出せるメモリにコピー
  - CPU側にある小さな高速メモリにコピーを置く
    TLB (Translation Lookaside Buffer) と呼ばれる
  - 主記憶に比べて非常に高速に読める
  - 但し、メモリキャッシュ（アーキテクチャ授業参照）の時と同様に、
    いつもキャッシュ上にある (ヒット) わけではないので
    外れ (ミス) れば主記憶上の表を見に行く～遅くなる

ここまでのまとめ　ページングの性能は

- アドレスの変換が必要なので、その分だけ遅くなる
  - 命令実行性能に対する影響は大きい
- 早くするためいろいろな工夫をしている
  - 表の引き方　～　アドレスから表位置を直接求める
  - キャッシュと同じ考え方を使ったＴＬＢ

ページングに関わる性能の問題が理解できましたか？