デマンドページングの性能３
性能と参照の局所性

性能とミス率②

- （ミス率×ｂ）は？
- ｂ＝ミス時アクセス時間／ヒット時アクセス時間
  の値は、ほぼハードで決まってしまう
  - 主記憶（半導体メモリ）　～　数nS
    ハードディスク　～　数mS
    ⇒　ｂ＝10^6ぐらい
- ミス率（ページフォルト率）が決め手になる
  - ｂ×ミス率＝１　（期待値がメモリアクセスの２倍程度）
    になるためには
    ミス率が　10^{-6}　程度でなくてはならない

言いだしっぺの議論の筋道（続）

- モデルはこれでいいとして、
  ミス率　（or ヒット率）　はどうなるのか？

言いだしっぺの議論の筋道（続）

- モデルはこれでいいとして、
  ミス率　（or ヒット率）　はどうなるのか？
  - 10^{-2} や10^{-3} （100回や1000回に1回）では多すぎる

東邦大学

言いだしっぺの議論の筋道（続）

- モデルはこれでいいとして、
  ミス率　（or ヒット率）　はどうなるのか？
  - 10^{-2} や10^{-3} （100回や1000回に1回）では多すぎる
- 実際に10^{-6}よりずっと小さくなるのか？

言いだしっぺの議論の筋道（続）

- モデルはこれでいいとして、
  ミス率　（or ヒット率）　はどうなるのか？
  - 10^{-2} や10^{-3} （100回や1000回に1回）では多すぎる
- 実際に10^{-6}よりずっと小さくなるのか？
  - 実測してみたら、「なる」　⇒　十分使える！

東邦大学

言いだしっぺの議論の筋道（続）

- モデルはこれでいいとして、
  ミス率　（or ヒット率）　はどうなるのか？
  - 10^{-2} や10^{-3} （100回や1000回に1回）では多すぎる
- 実際に10^{-6}よりずっと小さくなるのか？
  - 実測してみたら、「なる」　⇒　十分使える！
- «なぜか？»

言いだしっぺの議論の筋道（続）

- モデルはこれでいいとして、
  ミス率　（or ヒット率）　はどうなるのか？
  - 10^{-2} や10^{-3} （100回や1000回に1回）では多すぎる
- 実際に10^{-6}よりずっと小さくなるのか？
  - 実測してみたら、「なる」　⇒　十分使える！
- «なぜか？»　「参照の局所性」

東邦大学

言いだしっぺの議論の筋道（続）

- モデルはこれでいいとして、
  ミス率　（or ヒット率）　はどうなるのか？
  - 10^{-2} や10^{-3} （100回や1000回に1回）では多すぎる
- 実際に10^{-6}よりずっと小さくなるのか？
  - 実測してみたら、「なる」　⇒　十分使える！

- «なぜか？»　「参照の局所性」
  - 参照の局所性とは　⇒　次ページ
  - 局所性の出る原因　⇒　いくつか見てみる

参照の局所性（locality）とは

- 教科書 p107 （定義）⇒
  あるアドレス範囲 (例: ページ) への参照があったときに
  - 近い時間に同じ範囲への参照が起こる確率が高い
    （時間的局所性）

参照の局所性（locality）とは

- 教科書 p107 （定義）⇒
  あるアドレス範囲 (例: ページ) への参照があったときに
  - 近い時間に同じ範囲への参照が起こる確率が高い
    （時間的局所性）
  - その近くのアドレスへの参照が起こる確率が高い
    （空間的局所性）

参照の局所性（locality）とは

- 教科書 p107 （定義）⇒
  あるアドレス範囲 (例: ページ) への参照があったときに
  - 近い時間に同じ範囲への参照が起こる確率が高い
    （時間的局所性）
  - その近くのアドレスへの参照が起こる確率が高い
    （空間的局所性）
- 要するに、局所性が成り立つと
  - 近辺を含む範囲が主記憶にコピーされていれば
    その後しばらくは、その範囲内へのアクセスが多い
    ということ

東邦大学

参照の局所性（locality）が成り立つと

- メモリ参照（アクセス）が一定のページ範囲内なら
  - ページは既に物理メモリ上にあるのでミスしない

参照の局所性（locality）が成り立つと

- メモリ参照（アクセス）が一定のページ範囲内なら
  - ページは既に物理メモリ上にあるのでミスしない
  - 直前でなくても、「最近」であれば、
    そのページが物理メモリ上に残っているはず

東邦大学

参照の局所性（locality）が成り立つと

- メモリ参照（アクセス）が一定のページ範囲内なら
  - ページは既に物理メモリ上にあるのでミスしない
  - 直前でなくても、「最近」であれば、
    そのページが物理メモリ上に残っているはず
  - そのページが追出されると、無くなる（ミスになる）
    ⇒　追出しアルゴリズム（後述）に依存する？
    なるべく、次に参照する部分を追出さないようにすればよいだろう（後述）

参照の局所性はかなりよく成り立つ

- メモリの参照は、命令か、データ (オペランド) か

参照の局所性はかなりよく成り立つ

- メモリの参照は、命令か、データ (オペランド) か
- 命令はたいていはアドレス順に実行される
  - 後述する理由で、局所性が成り立つことが多い

参照の局所性はかなりよく成り立つ

- メモリの参照は、命令か、データ (オペランド) か
- 命令はたいていはアドレス順に実行される
  - 後述する理由で、局所性が成り立つことが多い
- データは近傍がアクセスされるとは限らないが

東邦大学

参照の局所性はかなりよく成り立つ

- メモリの参照は、命令か、データ (オペランド) か
- 命令はたいていはアドレス順に実行される
  - 後述する理由で、局所性が成り立つことが多い
- データは近傍がアクセスされるとは限らないが
  - 変数は、メモリ内ではたいてい近傍に配置される

参照の局所性はかなりよく成り立つ

- メモリの参照は、命令か、データ (オペランド) か
- 命令はたいていはアドレス順に実行される
  - 後述する理由で、局所性が成り立つことが多い
- データは近傍がアクセスされるとは限らないが
  - 変数は、メモリ内ではたいてい近傍に配置される
  - 変数同士が近傍なら、参照は少数ページ内に留まるはず

参照の局所性はかなりよく成り立つ

- メモリの参照は、命令か、データ (オペランド) か
- 命令はたいていはアドレス順に実行される
  - 後述する理由で、局所性が成り立つことが多い
- データは近傍がアクセスされるとは限らないが
  - 変数は、メモリ内ではたいてい近傍に配置される
  - 変数同士が近傍なら、参照は少数ページ内に留まるはず
  - そうでない配置もあり得る（後述）

    ⇒　局所性は成り立たず、アクセスが遅くなる

参照の局所性の概念が理解できましたか？