

# ページ置き換えの 動作とアルゴリズム (1) イントロ

## 前回のまとめ①

- 大きなメモリ      ページによるアドレス変換
- ↓                      ↓
- ページ分割したメモリ内容を  
ハードディスク（二次記憶）に置く

## 前回のまとめ①

- 大きなメモリ      ページによるアドレス変換  
    ↓                      ↓
- ページ分割したメモリ内容を  
    ハードディスク（二次記憶）に置く
- （必要に応じてHDへ戻す）ではなくて  
    （必要に応じてHDから物理メモリへ持ってくる）

## 前回のまとめ②

- 「必要に応じて」  
    ⇒ ページフォルト割込でハード的に知る  
    その後の「持ってくる」処理はソフト

## 前回のまとめ②

- 「必要に応じて」  
⇒ ページフォルト割込でハード的に知る  
その後の「持ってくる」処理はソフト
- 「必要に応じてディスクから持ってくる」は  
十分に早いのか？ 性能が落ちないのか？  
- ディスクは遅い ~ 時間が $10^6$ ぐらいかかる

## 前回のまとめ②

- 「必要に応じて」  
⇒ ページフォルト割込でハード的に知る  
その後の「持ってくる」処理はソフト
- 「必要に応じてディスクから持ってくる」は  
十分に早いのか？ 性能が落ちないのか？  
- ディスクは遅い ~ 時間が $10^6$ ぐらい違う  
- キーワード ⇒ 「参照の局所性」  
によりめったに起こらない

今回は  
「物理ページ枠がなくなったとき」  
を考えます。す

6

## 物理ページ枠がなくなったとき

- 物理メモリは（仮想メモリ空間より）小さい
- 「空きページ」を使ってゆくが  
いずれは使い尽くして、無くなる

7

# 物理ページ枠がなくなったとき

- 物理メモリは（仮想メモリ空間より）小さい
- 「空きページ」を使ってゆくが  
いずれは使い尽くして、無くなる
- 空きページが無くなったら、  
新しくページを物理メモリに持って来たい  
ときに困る

8

# 物理ページ枠がなくなったとき

- 物理メモリは（仮想メモリ空間より）小さい
- 「空きページ」を使ってゆくが  
いずれは使い尽くして、無くなる
- 空きページが無くなったら、  
新しくページを物理メモリに持って来たい  
ときに困る



- 要らなさそうなページを選んで「追出す」

9

## で、今回の話

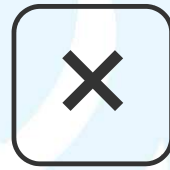
- 追出し (ページアウト) のアルゴリズム  
+
- 「実はそれによって性能が変わる」という話

## で、今回の話

- 追出し (ページアウト) のアルゴリズム  
+
- 「実はそれによって性能が変わる」という話
  - 追出したページがすぐ後に必要になるのだと、追出し+読み込みが無駄に起こって性能劣化
  - 将来使わないページを追出すのが理想
  - でも、将来のことは分からない…



では、次の節で  
具体的なアルゴリズムを  
議論しましょう



↓  
次へ