

システムプログラム概論

Memory management 2/2

2007/4/20

門林 雄基

奈良先端科学技術大学院大学

Copyright(C)2007 Youki Kadobayashi. All rights reserved.

1

ここまでのトピック

- メモリ階層 (memory hierarchy)
- メモリ資源の競合と調停
 - Swap in, swap out
- 仮想メモリ: ソフトウェア + ハードウェアでの解決
- アドレス変換
 - 物理アドレスから仮想アドレスへ
 - Paging, segmentation
 - Page table, segment descriptor

Copyright(C)2005 Youki Kadobayashi. All rights reserved.

2

今日の講義のポイント

- ページ管理のアルゴリズム
- 仮想メモリと性能
- 仮想メモリと障害の隔離

Copyright(C)2005 Youki Kadobayashi. All rights reserved.

3

Paging: Page table entry

- ハードウェアが更新
 - Referenced
 - Modified
- OSが更新
 - Present/absent
 - Protection
 - Caching disabled
 - Page frame number
- (MOS p. 210)

Copyright(C)2005 Youki Kadobayashi. All rights reserved.

4

Paging: Page fault

- 仮想ページ番号(n)に対応する物理ページなし
present bit = 0
プロセス(A)が仮想ページ内のメモリへアクセス
page fault
- Page fault 発生時、当該プロセスを一時中断しOSの仮想記憶モジュールを実行：
 - 該当ページ(n)をディスクから復帰 (page in)
present bit = 1
 - 当該プロセス(A)の処理を続行

Copyright(C)2005 Youki Kadobayashi. All rights reserved.

5

Paging: Page replacement

- 物理ページ数には限りがある
使っていない仮想ページをディスクへ退避
(page out)
 - Present bit = 0
- どのページを選んで退避させるか？ さまざまなアルゴリズムが考えられた
 - NRU, FIFO, second chance, LRU...

Copyright(C)2005 Youki Kadobayashi. All rights reserved.

6

NRU page replacement algorithm

- MOS 4.4.2 (p. 216)

FIFO page replacement algorithm

- MOS 4.4.3 (p. 217)

Second chance

- MOS 4.4.4 (p. 217)

LRU page replacement algorithm

- MOS 4.4.6 (p. 218)

Paging: Caching modified data

- 一般的な用語 (メモリ階層で使われる)
 - Write-through: ただちに書き戻す
 - Write-back: キャッシュから追い出されるまで書き戻さない
- Reliability / performance trade-offs
- ページアウトでは write-back 方式を採用

ワーキング・セット (Working set)

- $w(k, t)$ (MOS p. 223)
- k most recent memory references at time t
- スラッシング (Thrashing)
 - ワーキングセット・サイズがキャッシュサイズを超えると起きる

仮想メモリと性能

仮想記憶の功罪

- プログラミングは単純に
 - メモリ容量を気にする必要が無い
 - マルチプログラミングが単純に
- ディスクさえあれば動き続ける
 - スラッシングしていても...
 - 本当はメモリが足りないのに
- メモリ不足をどうやって判定するか？

Copyright(C)2005 Youki Kadobayashi. All rights reserved.

13

仮想メモリと性能: BSD UNIXの場合

● vmstat で全体のメモリ使用状況を把握

```
$ vmstat 1
procs          memory          page          disks          faults          cpu
r b w    avm    fre flt re  pi  po  fr  sr  ad0 md0  in  sy  cs  us  sy  id
2 0 0    147580 108404 100 0  1  0  82  0  0  0  116 696 95  3  1 96
0 0 0    147236 108404  16 0  0  0  0  0  0  0  105 223 53  0  0 100
0 0 0    147236 108404  15 0  0  0  0  0  0  0  103 186 44  1  0 99
0 0 0    147236 108404  15 0  0  0  0  0  0  0  110 176 44  0  0 100
0 0 0    147236 108404  15 0  0  0  0  0  0  0  107 254 49  1  0 99
```

● psで各プロセスのメモリ使用状況を把握

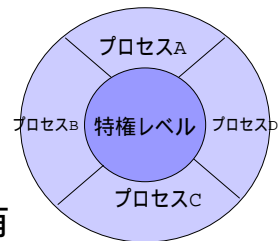
```
$ ps axOtsiz,rss,vsz
PID TSIz  RSS  VSZ  TT  STAT      TIME COMMAND
   0     0    0    0  ??  DLs    0:00.00 (swapper)
   1   248   316   544  ??  ILs    0:00.01 /sbin/init --
 284 1324 16316 17380  ??  R      0:10.10 /usr/X11R6/bin/XFree86 :0
 381 1156  8360  9752  p1  S      0:01.71 xemacs (xemacs-21.1.14)
```

Copyright(C)2005 Youki Kadobayashi. All rights reserved.

14

仮想メモリと障害の隔離

- プロセス内のメモリアクセス違反の検出
 - Segmentation fault
- 他のプロセスとのアドレス空間の分離
 - 特権レベルでのアドレス空間の生成



- プロセス間のアドレス空間の共有

Copyright(C)2005 Youki Kadobayashi. All rights reserved.

15

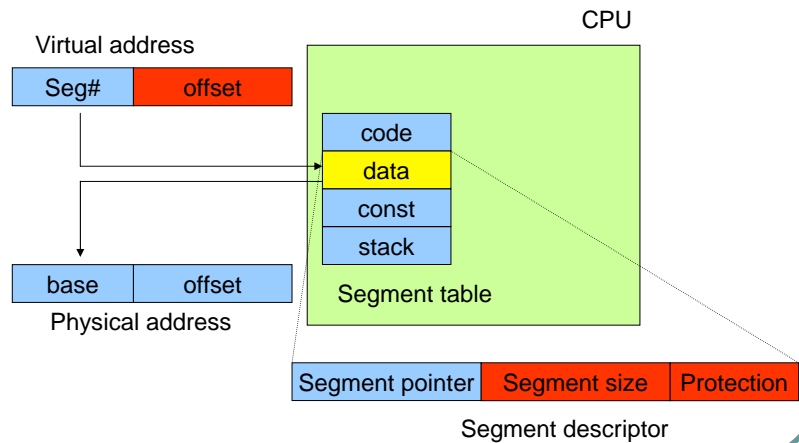
Segmentation fault

- プログラムのバグ等で、許可された範囲を超えてメモリアクセスを行った場合
- セグメント記述子に記載されたサイズを超えてメモリアクセス
or
- セグメント記述子に記載された権限を越えてメモリアクセス
 - segmentation fault

Copyright(C)2005 Youki Kadobayashi. All rights reserved.

16

Segmentation fault

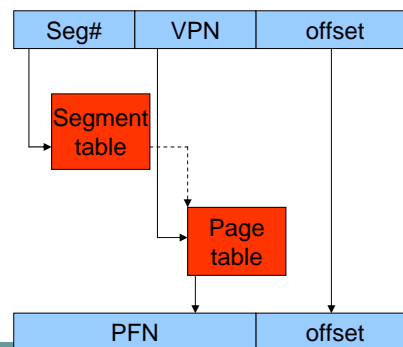


Copyright(C)2005 Youki Kadobayashi. All rights reserved.

17

特権レベルでのアドレス空間の生成

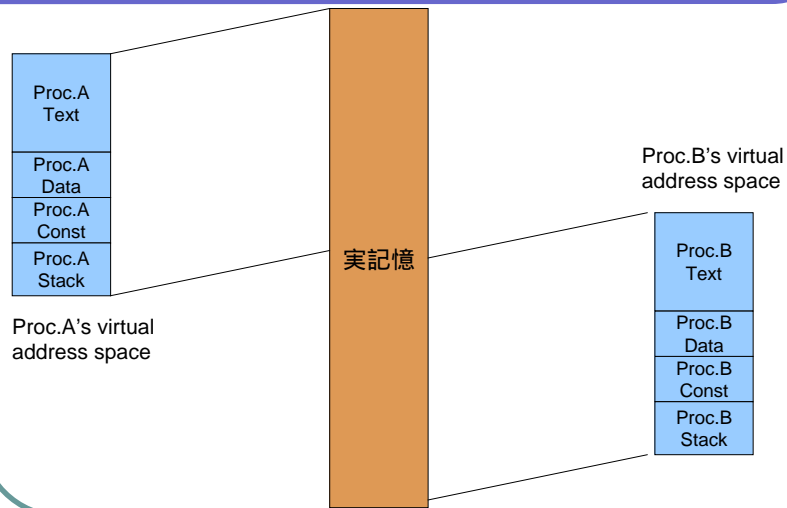
- アドレス空間の操作はOSのカーネル(プロセッサの特権レベル)でのみ許可される
- 下記の情報をOSカーネルが設定する
 - Page table register
 - Table size register
 - Page table
 - Segment descriptor
 - Segment table
- (MOS p. 261)



Copyright(C)2005 Youki Kadobayashi. All rights reserved.

18

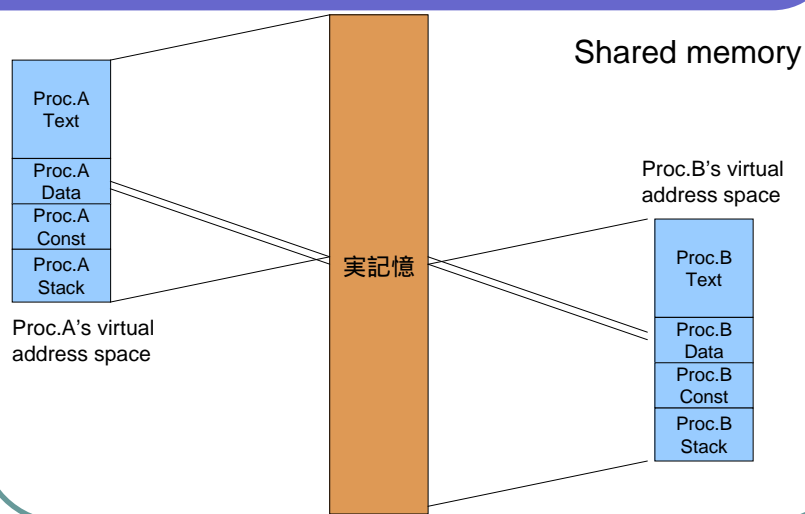
アドレス空間の分離



Copyright(C)2005 Youki Kadobayashi. All rights reserved.

19

アドレス空間の共有



Copyright(C)2005 Youki Kadobayashi. All rights reserved.

20

まとめ

- アドレス変換の効率化 - TLB
- ページ管理の方式
 - Page fault, page replacement
 - ページ置き換えアルゴリズム
 - Write back, write through
- メモリ管理と性能
 - Working set
 - Thrashing
- メモリ管理と障害の隔離
 - アドレス空間の管理 - 生成(分離)、共有
 - Segmentation fault

レポート

- 提出期限: 4/26 17:00(日本時間)
- A棟3F 山口研究室入り口レポート提出ボックス