

CS162
Operating Systems and
Systems Programming
Lecture 15

Page Allocation and
Replacement

October 22, 2008

Prof. John Kubiawicz

<http://inst.eecs.berkeley.edu/~cs162>

Review: Demand Paging Mechanisms

- **PTE helps us implement demand paging**
 - Valid \Rightarrow Page in memory, PTE points at physical page
 - Not Valid \Rightarrow Page not in memory; use info in PTE to find it on disk when necessary
- **Suppose user references page with invalid PTE?**
 - Memory Management Unit (MMU) traps to OS
 - » Resulting trap is a “Page Fault”
 - **What does OS do on a Page Fault?:**
 - » Choose an old page to replace
 - » If old page modified (“D=1”), write contents back to disk
 - » Change its PTE and any cached TLB to be invalid
 - » Load new page into memory from disk
 - » Update page table entry, invalidate TLB for new entry
 - » Continue thread from original faulting location
 - TLB for new page will be loaded when thread continued!
 - While pulling pages off disk for one process, OS runs another process from ready queue

Cache

Goals for Today

- **Precise Exceptions**
- **Page Replacement Policies**
 - **Clock Algorithm**
 - **Nth chance algorithm**
 - **Second-Chance-List Algorithm**
- **Page Allocation Policies**
- **Working Set/Thrashing**

Note: Some slides and/or pictures in the following are

adapted from slides ©2005 Silberschatz, Galvin, and

10/26/08 Gagne. Many slides generated from my lecture notes 15.3

by Kubiawicz