CS 423/523 Assignment 4

Published: Apr. 7, 2016
Due: Apr. 15, 2016 (11:59pm)

Total: 30 points

Please upload your solutions to classes*v2. To do so, please enter classes*v2, then click the “Assignment” button on your left-hand toolbar, and finally click “Assignment4” to upload your assignment.

If you know you are going to submit your assignment late, please let us know in advance (send an email to cs423ta@cs.yale.edu).

Any and all resources may be used as long as you cite them, with the exception of collaborating with other people. Please do not copypasta your definitions from Wiki. We do not really care if you do this, but you are not really learning anything.

If you have ANY questions, please do not hesitate to let us know (email, office hours, etc.)
Part 1: OS definitions (1-3 line answers, 5 x 2 points each = 10 points)

1. Address binding

Address binding is the process of mapping the program’s logical or virtual addresses to corresponding physical or main memory addresses.

2. Virtual memory

Virtual memory is a mechanism that separates logical and physical memory, allowing us to use secondary memory as if it were part of main memory. It maps logical main memory addresses to physical addresses in main or other memory.

3. What’s the motivation of proposing inverted page table? In other words, compared with a page table, what’s the difference.

Each process needs a page table, but inverted page table is a global page table which records and tracks all the processes’ memory information (e.g., PID and virtual page number), thus reducing the needed memory for storing each individual page table.

4. Difference between paging and segmentation (please list three different points)

1. Paging has linear address space and segmentation has many address spaces
2. Programmer does not need to know how paging is implemented, but for segmentation, programmer needs to know
3. Paging is a contiguous range of memory addresses but segmentation is an independent address, and each segment has addresses from 0 to max value
4. In paging, physical memory is divided into fixed-size blocks, but the segmentation does not.

(I guess they may answer differently, but any reasonable answers should be fine)

5. Lazy swapping

A flavor of paging where a page is never brought into memory unless the page is needed.

Part 2: Multiple choice (6 x 2 points each = 12 points)

1. Double buffering:
   a. Adds overhead
   b. Is used for context switching
   c. Both of the above
   d. None of the above

2. A forward mapped page table:
a. Pages a page table
b. Is faster than having normal page table
c. Both of the above
d. None of the above

3. Copy on write:
   a. Modified pages are copied back to disk
   b. Any two (or more) processes can share a page, as long as they do not write to it
   c. Both of the above
   d. None of the above

4. Global frame allocation:
   a. Means processes cannot control their page fault rate
   b. Want better throughput
   c. Both of the above
   d. None of the above

5. I/O interlocking
   a. Helps with prepaging
   b. Prevents eviction of some pages
   c. Both of the above
   d. None of the above

6. Least Recently Used (LRU) algorithm
   a. Needs past knowledge than future
   b. Needs special hardware
   c. Both of the above
   d. None of the above

Part 3: A longer question (2 x 4 points each = 8 points)

Suppose we have the following reference string: 041424342404 and three memory frames. Please draw the replacement pictures (like page32 in chapter 9) for First-In-First-Out (FIFO) and Least-Recently-Used (LRU) algorithms, respectively. In addition, please compute the number of page faults for each algorithm.

Please answer the following questions:
   1. Draw replacement picture for FIFO algorithm and compute the number of page faults.

E: empty frame

EEE, 0EE, 04E, 041, 041, 241, 241, 231, 234, 234, 234, 034, 034

Page fault is 7
2. Draw replacement picture for LRU algorithm and compute the number of page faults.

E: empty frame

EEE, 0EE, 04E, 041, 041, 241, 241, 243, 243, 243, 243, 240, 240

Page fault is 6