CIS 415 – Operating Systems

Homework Assignment 3
Fall 2016 – Prof. Sventek

Due at 5:00pm on Tuesday, 8 November 2016

All questions must be answered by you without outside assistance. **Submission is via Canvas.** You may submit either a plain text (.txt) or a PDF (.pdf) file. Succinct, concise answers to the questions are preferable to long, rambling ones.

Textbook Questions (50 points)

1. OSC 7.16: In a real computer system, neither the resources available nor the demands of processes for resources are consistent over long periods (months). Resources break or are replaced, new processes come and go, and new resources are purchased and added to the system. If deadlock is controlled by the banker's algorithm, which of the following changes can be made safely (without introducing the possibility of deadlock), and under what circumstances? [6 points]
    a. Increase *Available* (new resources added)
    This could safely be changed without any problems.
    b. Decrease *Available* (resources are permanently removed from the system)
    This could have an effect on the system and introduce the possibility of deadlock as the safety of the system assumed there were a certain number of available resources.
    c. Increase *Max* for one process (the process needs or wants more resources than allowed)
    This could have an effect on the system and introduce the possibility of deadlock.
    d. Decrease *Max* for one process (the process decides it does not need that many resources)
    This could safely be changed without any problems.
    e. Increase the number of processes
    This could be allowed assuming that resources were allocated to the new process(es) such that the system does not enter an unsafe state.
    f. Decrease the number of processes
    This could safely be changed without any problems.
2. OSC 8.13: Compare the memory organization schemes of contiguous memory allocation, pure segmentation, and pure paging with respect to the following issues: [9 points]
    a. External fragmentation
    b. Internal fragmentation
    c. Ability to share code across processes

    The contiguous memory allocation scheme suffers from external fragmentation as address spaces are allocated contiguously and holes develop as old processes die and new processes are initiated. It also does not allow processes to share code, since a process's virtual memory segment is not broken into noncontiguous finegrained segments. Pure segmentation also suffers from external fragmentation as a segment of a process is laid out contiguously in physical memory and fragmentation would occur as

segments of dead processes are replaced by segments of new processes. Segmentation, however, enables processes to share code; for instance, two different processes could share a code segment but have distinct data segments. Pure paging does not suffer from external fragmentation, but instead suffers from internal fragmentation. Processes are allocated in page granularity and if a page is not completely utilized, it results in internal fragmentation and a corresponding wastage of space. Paging also enables processes to share code at the granularity of pages.

3. OSC 8.25: Consider a paging system with the page table stored in memory. [5 points]
   a. If a memory reference takes 50 nanoseconds, how long does a paged memory reference take?
   100 nanoseconds: 50 nanoseconds to access the page table and 50 nanoseconds to access the word in memory
   b. If we add TLBs, and 75% of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes 2 nanoseconds if the entry is present.)
   Effective access time = 0.75 × (52 nanoseconds) + 0.25 × (100 nanoseconds) = 64 nanoseconds.
4. OSC 9.14: Assume that a program has just referenced an address in virtual memory. Describe a scenario in which each of the following can occur. (If no such scenario can occur, explain why.) [12 points]
   a. TLB miss with no page fault
   page has been brought into memory, but has been removed from the TLB
   b. TLB miss and page fault
   page fault has occurred
   c. TLB hit and no page fault
   page is in memory and in the TLB, most likely a recent reference
   d. TLB hit and page fault
   cannot occur - the TLB is a cache of the page table; if an entry is not in the page table, it will not be in the TLB.
5. OSC 9.21: Consider the following page reference string:

   7, 2, 3, 1, 2, 5, 3, 4, 6, 7, 7, 1, 0, 5, 4, 6, 2, 3, 0, 1

   Assume demand paging with 3 frames, how many page faults would occur for the following replacement algorithms? [18 points]
   a. LRU replacement – 18

7	7	7	1	1	1	3	3	3	7	7	7	7	5	5	5	2	2	2	1
	2	2	2	2	2	2	4	4	4	4	1	1	1	4	4	4	3	3	3
		3	3	3	5	5	5	6	6	6	6	0	0	0	6	6	6	0	0

   b. FIFO replacement - 17

7	7	7	1	1	1	1	1	6	6	6	6	0	0	0	6	6	6	0	0
	2	2	2	2	5	5	5	5	7	7	7	7	5	5	5	2	2	2	1
		3	3	3	3	3	4	4	4	4	1	1	1	4	4	4	3	3	3

c. Optimal replacement - 13

7	7	7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	5	5	5	5	5	5	5	5	5	4	6	2	3	3	3
		3	3	3	3	3	4	6	7	7	7	0	0	0	0	0	0	0	0

Note: Like all assignments in this class **you are prohibited from copying any content from the Internet or sharing ideas, code, configuration, text or anything else or getting help from anyone in or outside of the class, except where noted**. Consulting online sources is acceptable, but under no circumstances should anything be copied. Failure to abide by this requirement will result in sanctions ranging from zero on the assignment to dismissal from the class.