CMP426-697 Operating Systems
Assignment 3
Department of Mathematics and Computer Science,
Lehman College, the City University of New York, Fall 2011

Due by 6 PM on November 29th (at the beginning of the second exam)
Only Hard Copy Submission (submitting handwriting document is fine)
No email submission is accepted.

Students can discuss the questions together. But copying answers from other students' work is not allowed. All students involved in copying answers will get 0 point for the assignment.

You do not need to submit Q.1 and Q.2 answers.

[Q.1] Practice Homework 7-1 (on the Web Site
http://comet.lehman.cuny.edu/jung/cmp426697/cmp426697.html
or
http://dragonserver.lehman.edu/jung/cmp426697/cmp426697.html
)

Homework 7-1: Practice Linux semaphore examples (System V semaphores, POSIX semaphores)

- file copy example (fileioS.c pv.c based on System V semaphore): compile by $gcc -o fileioS fileioS.c pv.c; run by $./fileioS sourceFileName destinationFileName
- Producer Consumer example (buffer.h, buffer.c based on POSIX semaphores): compile by $gcc -o consumerProducer buffer.c -pthread; run by $./consumerProducer 5 3 3 (as an example 5 seconds to run the program 3 Producer threads created 3 Consumer threads created.

[Q.2] Practice Homework 7-3 (on the Web Site
http://comet.lehman.cuny.edu/jung/cmp426697/cmp426697.html
or
http://dragonserver.lehman.edu/jung/cmp426697/cmp426697.html
)

Homework 7-3: Practice consumer producer program example in Java (example is in Java Multithreading and Synchronization Mechanism)

[Q.3] Answer the following questions.

(a) Why deadlock detection is preferred solution to deadlock avoidance in real system environment (such as in a DBMS environment)?
(b) What is the content of matrix Need based on the table shown below?
(c) If a request from P1 arrives for (0,4,2,0), can the request be granted immediately (based on the table below)?

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

[Q.4]Let us assume we have a simple paging system with the following parameters: 2^{32} bytes of physical memory; page size of 2^{10} bytes; 2^{16} pages of logical address space.

(a) How many bits are needed to represent a logical address?
(b) How many bytes in a frame?
(c) How many bits are necessary to specify a unique frame number?
(d) How many entries do we have in the page table?
(e) How many bits are required for each page table entry? Let us assume we have a reference bit, valid/invalid, and dirty bit.

[Q.5] A virtual address *a* in a paging system is represented by a pair (*p, w*), where *p* is a page number and *w* is an offset value within the page. Let us assume *z* be the number of bytes in a page. Show how we can calculate *p* and *w* as functions of *z* and *a*.

[Q.6]Let us assume we have a simple segmentation system that has the following segment table.

Segment Number	Base Address	Limit (Length)
0	660	248
1	1752	422
2	222	198
3	996	604

Show how to derive physical address for each of the following addresses. Indicate if you have any segmentation faults.

(a) 0, 198
(b) 2, 156
(c) 1, 530
(d) 3, 444
(e) 0, 260

[Q.7] Let us assume the per-process page table of a running process be as follow. All numbers are decimal, and addresses are byte addresses. The page size is 1K bytes. Starting address of a frame *x = x * 1K*

Page Number	Valid/Invalid Bit	Reference Bit	Dirty Bit	Frame Number	Time Loaded	Time Referenced
0	1	1	0	4	60	161
1	1	1	1	7	120	160
2	0	0	0	--- N/A	--- N/A	
3	1	1	1	2	30	162
4	0	0	0	--- N/A	--- N/A	

5	1	1	1	0	20	163

(a) What's difference between virtual address and physical address?
(b) Translate the following virtual address to physical address based on the per-process page table (i) 1052 (ii) 2221 (iii) 5499
(c) Let us assume maximum number of frames allocated for a process is 4. Let us assume we use local replacement. Also assume the current pointers the second chance and enhanced second chance algorithms are pointing to the page number 0. Which frame will be selected as a victim to be replaced, by the following replacement algorithms (strategies), when a page fault occurs? Explain why in each case. (i) FIFO (ii) LRU (iii) Second Chance (iv) Enhanced Second Chance. (v) Optimal.

[Q.8] Consider a paging system with the page table stored in memory.

(a) If a memory reference takes 200 nanoseconds, how long does a paged memory reference take?
(b) If we add associative registers (TLBs), and 90 percent 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 10 nanoseconds if the entry is there.)

[Q.9] Let us assume page fault service time is 50 milliseconds and a memory access time is 50 nanoseconds. Let *p* be the probability of having no page fault. Show how you calculate the lower bound of *p* to tolerate only 5% memory access performance degradation.

[Q.10] Answer the following questions

(a) Define working set model.
(b) Explain how working set model is used for virtual memory management in the Windows XP operating system.