Spring 2015 COMP 3511 Homework Assignment #3 Solution

Handout Date: April 2 Due Date: April 16

Name: ______________ ID: ___________________E-Mail: _____________

Please read the following instructions carefully before answering the questions:

- You should finish the homework assignment **individually**.
- There are a total of **4** questions.
- When you write your answers, please try to be precise and concise.
- Fill in your name, student ID, email and Section number at the top of each page.
- Please fill in your answers in the space provided, or you can type your answers in the MS Word file.

- **<u>Homework Collection</u>: the hardcopy** is required and the homework is collected in collection box #16. The collection boxes locate outside Room 4210, near lift 21 (there are labels attached on the boxes).

1. (20 points) Multiple choices

   1) Which of the following conditions must be satisfied to solve the critical section problem?
   ①Aging ②Mutual Exclusion ③Deadlock ④Progress ⑤Bounded Waiting

   A) ①②③⑤
   B) ②③④⑤
   C) ②④⑤
   D) ③④⑤

   Answer: C

   2) At a particular time of computation the value of a counting semaphore is 7. Then 20 P operations and 15 V operations were completed on this semaphore. The resulting value of the semaphore is ____.

   A) 42
   B) 2
   C) 7
   D) 12

   Answer: B

   3) ____________ occurs when a higher-priority process needs to access a data structure that is currently being accessed by a lower-priority process.

A) Deadlock
B) Priority inversion
C) A race condition
D) A critical section

Answer: B

4) Which of the following condition is required for deadlock to be possible?

A) Mutual exclusion
B) A process may hold allocated resources while awaiting assignment of other resources.
C) No resource can be forcibly removed from a process holding it.
D) All of the mentioned.

Answer: D

5) One way to ensure that the circular wait condition never holds is to ________.

A) impose a total ordering of all resource types and to determine whether one precedes another in the ordering
B) to let a process wait for only one resource at a time
C) to never let a process acquire resources that are held by other processes
D) All of these.

Answer: A

6) A system has 12 magnetic tape drives and 3 processes: P0, P1, and P2.
Process P0 requires 10 tape drives, P1 requires 4 and P2 requires 9 tape drives.
Currently, 5 tape drives are allocated to P0, 2 are allocated to P1, 2 are allocated to P2.
Which of the following sequence is a safe sequence?

A) P0, P1, P2
B) P1, P0, P2
C) P2, P0, P1
D) P1, P2, P0

Answer: B

7) The solution to starvation is _______________.

A) the number of rollbacks must be included in the cost factor
B) the number of resources must be included in resource preemption
C) resource preemption be done instead

D) All of these

Answer: A

8) An address generated by a CPU is referred to as a _______.

A) physical address
B) logical address
C) post relocation register address
D) Memory-Management Unit (MMU) generated address

Answer: B

9) Given the logical address 0xABA8 (in hexadecimal) with a page size of 256 bytes, what is the page offset?

A) 0xAB
B) 0xA8
C) 0xA
D) 0xA800

Answer: B

10) Consider a logical address of 32 pages with 1024 words per page; mapped onto a physical memory of 16 frames. Thus, _______bits are required in the logical address; ______bits are required in the physical address.

A) 15, 14
B) 15, 16
C) 14, 16
D) 14, 15

Answer: A
$2^5 = 32$; $2^{10} = 1024$; 15bits needed by logical address.
$2^4 = 16$; $2^{10} = 1024$; 14bits needed by logical address.

2. (30 points) Please answer the following questions in a few sentences

1) (3 points) What are the three general ways that a deadlock can be handled?

Answer: A deadlock can be prevented by using protocols to ensure that a deadlock will never occur. A system may allow a deadlock to occur, detect it, and recover from it. Lastly, an operating system may just ignore the problem and pretend that deadlocks can never occur (1 points each).

2) (5 points) Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock-free.

Answer: Suppose the system is deadlocked. This implies that each process is holding one resource and is waiting for one more. Since there are three processes and four resources, one process must be able to obtain two resources. This process requires no more resources and, therefore it will return its resources when done.

3) (4 points) What are the differences between internal and external fragmentation?

Answer: External fragmentation occurs when there is sufficient total free memory to satisfy a memory request, yet the memory is not contiguous, so it cannot be assigned. Internal fragmentation occurs when a process is assigned more memory than it has requested and the wasted memory fragment is internal to a process.

4) (6 points) Please illustrate how swapping and sharing can increase the degree of multiprogramming.

Answer: Swapping enables processes to be copied from memory to a backing store and later copied back to memory, which allows more processes to run than those that can fit into memory at one time. Sharing code and data among different processes reduce the redundancy, thus save memory space for more processes to run.

5) (6 points) Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Which algorithm makes the most efficient use of memory?

Answer:
First-fit: 212K is put in 500K partition; 417K is put in 600K partition; 112K is put in 288K partition (new partition 288K = 500K − 212K); 426K must wait.
Best-fit: 212K is put in 300K partition; 417K is put in 500K partition; 112K is put in 200K partition; 426K is put in 600K partition.

Worst-fit: 212K is put in 600K partition; 417K is put in 500K partition; 112K is put in 388K partition; 426K must wait.
In this example, best-fit turns out to be the best.

6) (6 points) Consider a paging system with the page table stored in memory. If a memory reference takes 200 nanoseconds, how long does a paged memory reference take? If we add TLBs, and 75 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 zero time, if the entry is there.)

Answer: It takes 400 nanoseconds to complete a paged memory reference: 200 nanoseconds to access the page table and 200 nanoseconds to access the word in memory.
Effective access time = 0.75 × (200 nanoseconds) + 0.25 × (400 nanoseconds) = 250 nanoseconds.

3. (20 points) Consider the following snapshot of a system:

	Allocation	Max	Available
	A B C D	A B C D	A B C D
P_0	0 0 1 0	0 1 1 2	1 1 3 1
P_1	2 0 0 1	2 1 2 2	
P_2	4 0 2 1	6 1 5 2	
P_3	1 2 3 0	2 4 5 2	
P_4	1 2 0 3	2 2 4 6	

Answer the following questions using the banker's algorithm:

1) (8 points) Illustrate that the system is in a safe state by demonstrating an order in which the processes may complete.

Answer: P1, P0, P2, P4, P3.

2) (6 points) If a request from process P0 arrives for (0, 1, 0, 1), can the request be granted immediately?

Answer: No. The need matrix is as follows:

Need
A B C D
0 1 0 2
0 1 1 1
2 1 4 1
1 2 2 2
1 0 4 3

Since request0 < need0, update available,

	Allocation	Need	Available
	A B C D	A B C D	A B C D
P_0	0 1 1 1	0 1 0 2	1 0 3 0
P_1	2 0 0 1	0 1 1 1	
P_2	4 0 2 1	2 1 3 2	
P_3	1 2 3 0	1 2 2 2	
P_4	1 2 0 3	1 0 4 3	

The current state is NOT safe, since available resource D is less than the need of every process.

3) (6 points) If a request from process P4 arrives for (1, 0, 1, 0), can the request be granted immediately?

Answer: Yes. The need matrix is as follows:

Need
A B C D
0 1 0 2
0 1 1 1
2 1 4 1
1 2 2 2
0 0 3 3

Since request4 < need4, update available,

	Allocation				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	1	1	1	0	1	0	2	0	1	2	1
P1	2	0	0	1	0	1	1	1				
P2	4	0	2	1	2	1	3	2				
P3	1	2	3	0	1	2	2	2				
P4	2	2	1	3	0	0	3	3				

The current state is safe. A feasible execution sequence is P1, P0, P2, P4, P3.

4. (30 points) Memory management

1) (15 points) Consider the following segment table:

Segment	Base	Length
0	90	100
1	300	99
2	2500	150
3	150	10
4	1283	200

What are the physical addresses of the following logical address?

a) 0, 99

b) 1, 100

c) 2, 56

d) 3, 100

e) 4, 1

Answer:

a) 90 + 99 = 189

b) 101 > 99, illegal reference, trap to operating system

c) 2500 + 56 = 2556

d) 101 > 10, illegal reference, trap to operating system

e) 1283+1=1284

2) (15 points) Consider a paging system: the page size is 1K bytes, and the physical memory is 64K bytes. For a given process, it contains 6 pages and its page table is shown below (note that page number is the index into the page table, and itself is not part of the page table).

Page No.	Frame No. (decimal)
0	11
1	3
2	9
3	40
4	23
5	1

a) Please give the corresponding physical address (in binary) of the decimal virtual address 2211. Briefly show your steps. (5 points)

Answer:

Virtual memory space is 64KB → virtual memory address is 16 bits.

Page size is 1KB → for the virtual address, the lowest 10 bits are for page offset, and the highest 6 bits are for page index.

$$(2211)_{10} = (\underbrace{0000\ 10}_{\text{page no.}}\underbrace{00\ 1010\ 0011}_{\text{offset}})_2$$

Therefore, the frame id is 9, that would be 00 1001.

The offset is 00 1010 0011. Append the offset to frame id, so the physical address should be 0010 0100 1010 0011.

b) Please give the physical address (in decimal) of the decimal virtual address 4321. Briefly show your steps. (5 points)

Answer:

4321 = 4096 + 225 = 4K + 225. So virtual address 4321 is indexed by page NO. 4. The physical frame NO. corresponding to page NO. 4 is 23.

So the physical address of 4321 is: 23K + 225 = 23777

Note: b) can be also solved using similar approach illustrated in solution a)

c) Consider a process with a logical address space of 64 Mbytes in size. Paging is used with a page size of 1 Kbytes. The physical memory size is 1 Mbytes. Please specify the size of the page table (in bytes). You can ignore all the control bits and suppose each page table entry consists of 4 bytes. Briefly show your steps. (Hint: use hierarchical paging scheme if necessary) (5 points)

Answer:

# of entries for inner page table 64M/1K = 64K (2^16)

# of inner page table size = 64K x 4 = 256K

# of entries for outer page table 256K/1K = 256 Total size of page table is (256+64K)*4 bytes