Ex/MCA/224/44/2015 (Old)

MCA Second Year Second Semester Examination, 2015 (Old)
Operating Systems

Time – 3 Hours **Full Marks – 100**

Answer any five questions

Answers to different parts of a question must be written together

1.

a. Define context switch. What is a dispatcher?

b. Differentiate between preemptive and non preemptive shortest job first scheduling algorithm.

c. Consider the following set of processes that need to be scheduled on a single CPU. All the times are given in milliseconds.

Process Name	Arrival Time	Execution Time
P_0	0	6
P_1	3	2
P_2	5	4
P_3	7	6
P_4	10	3

Determine the average turn-around time, average waiting time using preemptive shortest job first, non preemptive shortest job first and longest remaining time first scheduling algorithm. In longest remaining time first scheduling, ties are broken by giving priority to a process with lowest id.

d. What is a thread? Differentiate between a user level thread and kernel level thread.

(2+2)+2+10+(2+2)=20

2.

a. Explain how the critical section problem can be solved in a system which supports swap machine instruction.

b. What is busy waiting? Propose one implementation of semaphore which overcomes the busy wait problem.

c. Six semaphores *a*, *b*, *c*, *d*, *e*, *f* are used to enforce mutually exclusive access to six different resources A, B, C, D, E, F, among three processes P_0, P_1 and P_2. Each of these processes infinitely execute a critical section where they need to use subsets of the resources, as shown in the following code segments. All the semaphores are binary and are initialized to one.

Process P_0	Process P_1	Process P_2

while(true){
  wait(*a*);
  wait(*b*);
  wait(*c*):
  <Critical Section:
   Use A, B, C>
  signal(*a*);
  signal(*b*);
  signal(*c*);
}
	while(true){
	  wait(*d*);
	  wait(*e*);
	  wait(*b*):
	  <Critical Section:
	   Use D, E, B>
	  signal(*d*);
	  signal(*e*);
	  signal(*b*);
	}
		while(true){
		  wait(*c*);
		  wait(*f*);
		  wait(*d*):
		  <Critical Section:
		  Use C, F, D>
		  signal(*c*);
		  signal(*f*);
		  signal(*d*);
		}

State if there is any possibility of a deadlock in this implementation. Modify orders of some of the wait() operations to prevent the possibility of any deadlock. You cannot move wait() operations across processes, only change the order inside each processes. Justify your answer.

4+(2+4)+8=20

3.

a. Describe the requirements that any solution to the critical section problem must satisfy. Consider the methods used by processes P_1 and P_2 for accessing their critical sections as given below. The initial values of shared Boolean variables S_1 and S_2 are randomly assigned.

$\underline{P_1}$	$\underline{P_2}$
while(S_1==S_2);	while(S_1!=S_2);
<Critical Section>	<Critical Section>
S_1=S_2;	S_1= (not) S_2;

Explain which of the requirements are satisfied by this solution.

b. What are the synchronization requirements of the producer and consumer processes in both bounded-buffer and unbounded-buffer producer consumer problem?

c. Consider the following solution to the bounded buffer producer-consumer problem by using counting semaphores semaphore F, E and binary semaphore S. The semaphore S provides mutually exclusive access to the buffer and is initialized to 1. The semaphore F corresponds to the number of free slots in the buffer and is initialized to N (where N is the size of the buffer). The semaphore E corresponds to the number of elements in the buffer and is initialized to 0.

Producer process	Consumer process
produce an item	wait(E);
wait(S);	wait(S);
wait(F);	<Remove an item from the buffer>
<Append the item to the buffer>	signal(S);
signal(S);	signal(E);
signal(E);	consume the item

State whether the above solution is deadlock free. If not, explain in which case it leads to deadlock.

(3+5)+3+5=20

4.

a. What are the necessary conditions of deadlock? Explain each briefly.

b. What is the difference between deadlock prevention and deadlock avoidance mechanisms?

c. How resource allocation graphs are used for deadlock detection?

d. A system has four resource types and five running processes. Resource types A, B, C and D have 6, 7, 12 and 12 instances respectively. Consider the following snapshot of a system. Allocation column gives current allocation of each process and Maximum column gives maximum needs of each process.

	Allocation				Maximum			
	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	1	2
P_1	2	0	0	0	2	7	5	0
P_2	0	0	3	4	6	6	5	6
P_3	2	3	5	4	4	3	5	6
P_4	0	3	3	2	0	6	5	2

Determine if the system is currently deadlocked. If a request from P2 arrives for (0, 1, 0, 0) can it be granted immediately? If not, which processes may become deadlocked if the whole request is granted?

4+2+2+12=20

5.

a. Explain external fragmentation in the context of contiguous memory allocation. Propose one solution to overcome external fragmentation.

b. Consider a paging system with 40 bit logical address and 24 bit physical address. Page size is 4 KB and size of each page table entry is 2 bytes. Determine the size of page table if single level page table is used. How many bits in each page table entry can be used for storing protection and other information?
Now, assume that you want to implement a multi level page table. Determine how many levels of page table is required if you need to store each page of the page table possibly in non contiguous frames in physical memory. Determine the division of bits of the logical address that is required to address each levels of the multi level page table.
If physical memory access time is 10 nanoseconds, compare the effective memory access time of the single level and proposed multi level paging scheme.
Assume that, A TLB with hit ratio of 90% and access time 1 nanosecond, is introduced to cache recently used page table entries. Compare the effective memory access time of the single level and multi level paging scheme with TLB.

c. What is an inverted page table?

(2+2)+14+2=20

6.

a. Which factors determine minimum number of page frames that must be allocated to a running process in a virtual memory environment?

b. What is page fault? State actions performed by the Operating System to service a page fault.

c. Given five non contiguous memory partitions of 200 KB, 500 KB, 100 KB, 300 KB, and 600 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 250 KB, 400 KB, 150 KB, and 300 KB (in order)? Which algorithm produces largest left over partition?

d. A system uses 4 frames for storing pages of a process in physical memory. Assume that all the frames are initially empty. Determine the total number of page faults that will occur while processing the page reference string given below, using FIFO, Optimal and LRU page replacement policy respectively.
4, 7, 6, 1, 7, 6, 1, 2, 7, 2, 6, 1

2+(1+3)+6+8=20

7.

a. What is thrashing? Explain the causes of it

b. Differentiate between seek time and latency time.

c. What is the maximum size of a disk partition which is formatted using FAT-32 file system? Assume that, allocation unit size is equal to 2048 bytes.

d. Consider a hard disk with 16 recording surfaces (0-15) having 16384 cylinders (0-16383) and each track containing 64 sectors (0-63). Data storage capacity in each sector is 512 bytes. Data are organized cylinder wise and the addressing format is <cylinder no., surface no., sector no.>. A file of size 42797 KB is stored in the disk and the address of the first sector of the file is <1200, 9, 40>. Determine total size of the hard disk and address of the last sector of the file.

(1+2)+3+4+10=20

--------|-------