Reg. No. : | | | | | | | | | | | |

R 3242

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2007.

Fourth Semester

(Regulation 2004)

Computer Science and Engineering

CS 1252 — OPERATING SYSTEMS

(Common to B.E. (Part-Time) Third Semester Regulation 2005)

Time : Three hours — Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. What is Spooling?
2. Classify Real time systems.
3. Define a thread. State the major advantage of threads.
4. State the assumption behind the bounded buffer producer consumer problem.
5. List the four conditions for dead lock.
6. State what is required to support dynamic memory allocation in contiguous memory allocation?
7. What is thrashing?
8. What is pre-paging?
9. Briefly discuss the relative advantages and disadvantages of' sector sparing and sector slipping.
10. Under what circumstances would one use the deferred procedure calls facility in Windows XP?

PART B — (5 × 16 = 80 marks)

11. (a) (i) List and discuss the various services provided by the operating system. (8)

(ii) List and discuss the important modules of an operating system (Layers of an operating system). (8)

Or

(b) (i) In what ways is the modular kernel approach similar to the layered approach? In what ways does it differ from the layered approach? (4)

(ii) How do clustered systems differ from multiprocessor systems? What is required for two machines belonging to a cluster to cooperate to provide a highly available service? (4)

(iii) Compare Batch operating system and Time Sharing operating system. (8)

12. (a) Consider the following five processes, with the length of the CPU burst time given in milliseconds.

PROCESS	BURST TIME
P_1	10
P_2	29
P_3	3
P_4	7
P_5	12

Consider the First Cum First Serve (FCFS), Non Preemptive Shortest Job First (SJF), Round Robin (RR) (quantum = 10 milliseconds) scheduling algorithms. Illustrate the scheduling using Gantt Chart. Which algorithm will give the minimum average waiting time? Discuss. (16)

Or

(b) (i) Show how to implement the wait() and signal() semaphore operations in multiprocessor environments using the TestAndSet( ) instruction. The solution should exhibit minimal busy waiting. Develop Pseudocode for implementing the operations (8)

(ii) Demonstrate that monitors and semaphores are equivalent insofar as they can be used to implement the same types of synchronization problems. (8)

13. (a) (i) Construct a Resource Allocation Graph for the following scenario. At time 't' Process P_1 requests for a resource X, Process P_2 requests for a resource Y. Both the resources are available and they are allocated to the requesting Process. At time t_1 where $t_1 > t$, both the processes are still holding the resources, however process P_1 requests for Y which is held by P_2, process P_2 requests for X held by P_1. Will there be a dead lock? If there is a deadlock discuss the four necessary conditions for deadlock, else justify there is no deadlock. (8)

(ii) With a relevant example show that the implementation of a semaphore with a waiting queue may result in dead lock. (8)

Or

(b) (i) 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 associative registers, and 75 percent of all page-table references are found in the associative registers, what is the effective memory reference time? (Assume that finding a page-table entry in the associative registers takes zero time, if the entry is there.) (4)

(ii) On a system with paging, a process cannot access memory that it does not own, why? How could the operating system allow access to other memory? Why should it or should it not? (6)

(iii) Compare the segmented paging scheme with the hashed page tables scheme for handling large address spaces. Under what circumstances is one scheme preferable over the other? (6)

14. (a) (i) Discuss the hardware support required to support demand paging. (4)

(ii) Give relevant example and discuss situations under which the least frequently used page-replacement algorithm generates fewer page faults than the least recently used page replacement algorithm. Also give an example and discuss under what circumstance does the Opposite holds. (6)

(iii) Give relevant examples and discuss situations under which the most frequently used page-replacement algorithm generates fewer page faults than the least recently used page replacement algorithm. Also give an example and discuss under what circumstance does the opposite holds. (6)

Or

(b) (i) Give an example of an application that could benefit from operating system support for random access to indexed files. (4)

(ii) List and briefly discuss the most common schemes for defining the logical structure of a directory. (12)

15. (a) (i) List and discuss the various methods for implementing a directory. (8)

(ii) Some file systems allow disk storage to be allocated at different levels of granularity. For instance, a file system could allocate 4 KB of disk space as a single 4-KB block or as eight 512-byte blocks. How could we take advantage of this flexibility to improve performance? What modifications would have to be made to the free-space management scheme in order to support this feature? (8)

Or

(b) Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130 Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk-scheduling algorithms?

(i) FCFS (4)

(ii) SSTF (4)

(iii) SCAN (4)

(iv) LOOK (4)
