M.E. COMPUTER SCIENCE AND ENGINEERING

FIRST YEAR

FIRST SEMESTER EXAM 2024

ADVANCED OPERATING SYSTEMS

Time: Three hours Full Marks : 100

Answer any *five* questions. 5 X 20 = 100

1. What are the features of microkernel architecture? What are the advantages and disadvantages of this architecture? How can monolithic kernel be redesigned to adapt the modularity feature of microkernel? 5+10+5=20

2. What are two differences between user-level threads and kernel-level threads? Under what circumstances is one type better than the other? What are the characteristics of a process? What is the difference between *mode switch* and *context switch*? 10+5+5=20

3. Describe the differences among short-term, medium-term, and long term scheduling. Compare Blocking send and Nonblocking send. Consider the following set of four tasks : {(1,3),(1,5),(1,6),(2,10)} where the first number in each subset denotes the execution time and the second number denotes the periodicity of each task respectively. Are these tasks schedulable? Justify your answer. 5+5+10=20

4. Is it possible to have concurrency but not parallelism? Explain. How does *sender-initiated algorithm* work? In what system condition this type of algorithm works better and why? 5+10+5=20

5. Briefly describe Linux File System Structure. What is the purpose and contents of super-block in Linux OS? How does *inode* map to a data block file? 5+5+10=20

6. Provide two programming examples where multithreading does not provide better performance than a single-threaded solution. Explain why interrupts are not appropriate for implementing synchronization primitives in multiprocessor systems. Describe how the compare and swap() instruction can be used to provide mutual exclusion that satisfies the bounded-waiting requirement. 5+5+10=20

[Turn over

7. Solve the dining philosophers problem using monitors instead of semaphores. Write a producer-consumer problem that uses threads and shares a common buffer. However, do not use semaphores or any other synchronization primitives to guard the shared data structures. Just let each thread access them when it wants to. Use sleep and wakeup to handle the full and empty conditions. See how long it takes for a fatal race condition to occur. For example, you might have the producer print a number once in a while. Do not print more than one number every minute because the I/O could affect the race conditions. 10+10=20

8. Explain the difference between preemptive and non-preemptive scheduling. Affinity scheduling reduces cache misses. Does it also reduce TLB misses? What is coherent memory? 5+10+5=20