CE7-R3: REAL TIME SYSTEMS

NOTE:

1. **Answer question 1 and any FOUR questions from 2 to 7.**
2. **Parts of the same question should be answered together and in the same sequence.**

Time: 3 Hours **Total Marks: 100**

1.

a) What do you understand by clock synchronization?

b) What are the properties, which should be satisfied by a transaction in a conventional database?

c) Why do static priority task scheduling algorithms needs less run time overhead compared to the dynamic priority task scheduling algorithms?

d) Give two reasons why deadlock detection is more practical than deadlock avoidance.

e) Why can any interactive application be considered as a type of real-time system?

f) What are the features of Real Time CORBA suitable for distributed real time systems?

g) Can hardware fault-tolerance techniques be easily adapted to provide software fault-tolerance?

(7x4)

2.

a) What are the roles of sensors, actuators and controllers in real-time systems? Give examples of these from practical systems. Depict the general structure of a real-time system by using a schematic diagram, clearly showing the controller, actuators and sensors.

b) In providing system-level fault-tolerance, why are hardware failures more predictable and easier to handle compared to software failures? Explain the N-version scheme to provide software fault-tolerance. What are the shortcomings of this scheme?

c) Discuss the important performance parameters, which you would consider to choose between two real-time computers.

(6+6+6)

3.

a) What do you understand by temporal data? Give some examples of temporal data. Can any database containing temporal data be called a real-time database? Discuss.

b) Would you prefer an optimistic concurrency control protocol or a pessimistic concurrency control protocol for use in real-time database systems? Would your choice depend on the load on the system? Explain your answer.

c) Explain a disk-scheduling algorithm, which can satisfactorily be used in real-time applications.

(6+6+6)

4.

a) Why is debugging and testing of real-time software more difficult than traditional software? Explain how real-time software can be efficiently debugged and tested.

b) Consider a 10Mbps token ring network. The walk time is 1 mSec. The shortest deadline among different messages is 10mSec. The frame size is 512 bytes. Determine the maximum time for which a message may undergo priority inversion under IEEE 802.4 and IEEE 802.5 protocols.

c) What problems would you experience if you used a contention-based protocol such as Ethernet for real-time task communications? Describe a contention-based real-time communication protocol and explain how it overcomes the problems that Ethernet suffers from.

(6+6+6)

5.

a) Is EDF algorithm used for scheduling real-time tasks a dynamic priority-scheduling algorithm? Does EDF compute any priority value of tasks any time? If yes, explain when it is computed and how it is computed. If no, explain what is the concept of priority in EDF then.

b) Determine whether the following set of periodic tasks is schedulable on a uniprocessor using RMA ignoring context switch overhead.

Task	**Processing-time (mSec)**	**Period (mSec)**
T1	28	100
T2	70	200
T3	60	400

Now, Suppose context switch overhead of 1 milli Seconds is to be taken into account, determine the schedulability.

c) What do you understand by exception handling? Explain how you can handle exceptions using a programming language of your choice.

(6+8+4)

6.

a) Why are concurrency control protocols required? Identify the reasons why 2PL is not a suitable protocol for use in real-time databases. How can it be modified for use in real-time database?

b) Why is Real-Time Linux more suitable to support real-time applications compared to the traditional Unix system. Explain your answer with respect to some of the important features required to support running real-time applications. Can Real-Time Linux be used in embedded applications?

c) Examine the pros and cons of using an object-oriented language for real-time application development.

(6+8+4)

7.

a) Explain how clock synchronization can be achieved in a distributed real-time system. Why does presence of Byzantine faults makes clock synchronization problem difficult? What is the minimum number of clocks necessary to ensure proper synchronization in the presence of m Byzantine faults?

b) Are supports for pointer data type and pointer arithmetic desirable for a programming language used in real-time application development? Explain your answer.

c) What do you understand by the term “delay jitter” in a real-time communication application? Identify at least two factors, which contribute to delay jitter in real-time communications and explain how they cause jitter.

(8+5+5)