Total No. of Questions : 12]

SEAT No. : □

P851

[Total No. of Pages : 3

## [4458] - 803

B.E. (Information Technology) (Semester - II)

REAL TIME SYSTEM

(2008 Course) (Elective - III(a))

Time :3 Hours] ***[Max. Marks :100***

Instructions to candidates:

1) ***Answer 3 questions from Section-I and 3 questions from Section-II.***
2) ***Attempt not more than 3 questions of which at least 3 questions must be from each section.***
3) ***Answers to the two sections should be written in separate books.***
4) ***Neat diagrams must be drawn wherever necessary.***

SECTION - I

Q1) a) What is Hierarchical View of Performance measure of RTS? In what way it is different than traditional Measure of performance? **[8]**

b) Explain the following properties of good performance measure. **[8]**

i) Efficient encoding.

ii) Objective optimization criteria.

iii) Objective basis for ranking.

iv) Verifiable fact

OR

Q2) a) Draw and explain Schematic flow chart of a timing estimation of Real Time System. **[8]**

b) Describe in brief the effect of following in estimating the run time of a program. **[8]**

i) Source Code

ii) Compiler

iii) Machine Architecture

iv) Operating System

Q3) a) Classify Priority-driven scheduling approach for HRTS. Draw time line diagram for below example using Rate-monotonic (RM) and Earliest deadline first (EDF) algorithms.

Ex: $T_1 = (2,0.9)$; $T_2 = (5, 2.3)$ **[10]**

P.T.O.

b) Execute following tasks using Deadline monotonic (DM) scheduling algorithm. Ex: T1= (50, 50, 25, 100); T2= (0, 62.5, 10, 20); T3= (0, 125, 25, 50) **[8]**

OR

Q4) a) State the assumption made for the implementation of the Rate Monotonic Scheduling algorithm. What is the easy schedulability test for this algorithm? **[8]**

b) Consider the following set of periodic tasks: **[10]**

T1 = (Release Time, Deadline, Execution Time) = (0, 5, 3);

T2 = (Release Time, Deadline, Execution Time) = (2, 4, 1);

T3 = (Release Time, Deadline, Execution Time) = (6,10,3);

Assigned tasks using Preemptive Earliest Deadline First.

Q5) a) Using example explains the different data typing features that could be useful in a real time programming language. **[8]**

b) Describe the skeleton and optimistic algorithm under the two phase approach to improve predictability of real time transaction. **[8]**

OR

Q6) a) Why main memory database faster than disk-base databases for single processor systems? **[4]**

b) Differentiate Real Time databases and General purpose database. **[4]**

c) Describe the adaptive earliest deadline (AED) algorithms used in transaction priorities. **[8]**

<u>SECTION - II</u>

Q7) a) Explain virtual Time carrier sensed multiple access (VTCSMA) algorithm with flow chart. Draw the VC-RC trajectory for example $n = 2$, $n = 4$. **[10]**

b) Describe the stop and Go multihop protocol and comment on its performance. **[8]**

OR

Q8) a) Discuss the window protocol with suitable example. Discuss the performance of this algorithm. Where is the contention protocol most suitable. **[10]**

b) Describe the Timed Token protocol. Why this protocol is attractive for RTS. **[8]**

Q9) a) Explain in details capability of RTlinux along with specific API for time services. **[8]**

b) Describe which scheduling algorithm is used in RTlinux as against standard linux. **[4]**

c) Explain in detail timer function support in RTlinux, where it is used. **[4]**

OR

Q10)a) Explain in detail capability of VX works along with specific API for time service. **[8]**

b) Explain in detail what is interrupt latency? **[4]**

c) Explain in detail asynchrous timer function support in VX works and where it is used. **[4]**

*Q11)*a) Describe the classification of faults according to their behavior. Explain ‘Fault and Error containment zone’. **[6]**

b) Explain the reliability models for hardware redundancy. **[6]**

c) Write short notes on (any one) : **[4]**

i) Time Redundancy.

ii) Data diversity.

OR

*Q12)*a) Describe the following structures for hardware redundancy. **[8]**

i) Static Pairing.

ii) Shift-out Redundancy.

b) Describe in detail Fault-Tolerant synchronization in hardware and Fault-Tolerant synchronization in software. **[8]**

✦✦✦✦