M.Sc. 1st Semester Examination, 2023

COMPUTER SCIENCE

PAPER – COS-104

Full Marks : 50

Time : 2 hours

Answer **all** questions

The figures in the right hand margin indicate marks

Candidates are required to give their answers in their own words as far as practicable

GROUP – A

1. Answer any *four* from the following questions : 2×4

   (*a*) What is Triple Modular Redundancy (TMR) ?

   (*b*) What do you understand by 'safety critical system' ?

(*c*) What is the difference beween hard and soft real time tasks ?

(*d*) Give one examples for each of SS and SR deadline constraints.

(*e*) What is the difference between aperiodic and sporadic tasks ?

(*f*) Consider a real-time sysem in which tasks are scheduled using foreground-background scheduling. There is only one periodic foreground task T_f: ($\phi_f = 0$, $p_f = 50$ msec, $d_f = 100$ msec) and the background task be $T_B = (e_B = 1000$ msec) Compute the completion time for background task.

(*g*) What do you understand by *scheduling point* of a task scheduling algorithm ?

(*h*) What do you understand by pre-emptive and non-pre-emptive tasks ?

GROUP–B

2. Answer any *four* from the following questions : 4×4

(*a*) A cyclic schedular is to be used to run the following set of periodic tasks on a uniprocessor : $T_1 = (e_1 = 1, p_1 = 4)$, $T_2 = (e_2 = 1, p_2 = 5)$, $T_3 = (e_3 = 1, p_3 = 20)$, $T_4 = (e_4 = 2, p_4 = 20)$. Find an appropriate frame size meeting the required constraints.

(*b*) (*i*) Explain why hardware fault tolerance is easier to achieve compared to software fault-tolerance.

(*ii*) What are the main technique available to achieve to software fault-tolerance ? $2 + 2$

(*c*) Explain how PCP is able to avoid deadlock, unbounded priority inversion and chain blockings. 4

(*d*) Explain using an appropriate example as to why a critical resource can get corrupted if the task using it is pre-empted, and then another task is granted use of the resource.

(*e*) Consider the following set of periodic real-time tasks : $T_1 = (e_1 = 10 \text{ msec}, p_1 = 50 \text{ msec})$, $T_2 = (e_2 = 25 \text{ msec}, p_2 = 150 \text{ msec})$, $T_3 = (e_3 = 50 \text{ msec}, \quad p_3 = 200 \text{ msec})$. Assume that the self suspension times of T_1, T_2 and T_3 are 3 msec, 3 msec and 5 msec, respectively. Determine whether the tasks would meet their respective deadlines, if scheduling using RMA. 4

(*f*) Briefly indicate how Unix dynamically recomputes task priority values. Why is such recomputation of task priorities required ? 2 + 2

(g) What do you understand by jitter associated with a periodic task ? How are these jitters caused ? 2 + 2

(*h*) Consider the following set of three independent real-time periodic tasks.

Task	Start time (mSec)	Processing time (mSec)	Period (mSec)	Deadline m(Sec)
*T*1	20	25	150	100
*T*2	40	10	50	30
*T*3	60	50	200	150

Suppose a cyclic schedular is to be used to schedule the task set. What is the major cycle of the set ? Suggest a suitable frame size. 2 + 2

GROUP – C

3. Answer any *two* from the following questions : 8 × 2

(*a*) Consider the following set of three periodic real-time tasks : $T_1 = (10, 20)$, ($T_2 = (15, 60)$, $T_3 = (20, 120)$ to be run on uniprocessor. Determine whether the task set is schedulable under RMA. 8

(*b*) Discuss briefly various features of the Real-Time operating system.

(*c*) (*i*) Why is it necessary to synchronize the clocks in the distributed real-time system ?

(*ii*) Explain why algorithms that can be used satisfactorily to schedule real-time tasks on microprocessors often are not satisfactory to schedule real-time tasks on distributed systems and vice-versa ? 3 + 5

(*d*) A set of hard real-time periodic tasks need to be scheduled on a uniprocessor using RMA. The following table contains the details of these periodic tasks and their use of three non-preemptive shared resource. Can the tasks T2 and T3 meet their respective deadlines when priority ceiling protocol (PCP) is used for resources scheduling ?

Task	p_i	e_i	R_1	R_2	R_3
$T1$	400	30	15	20	–
$T2$	200	25	–	20	10
$T3$	300	40	–	–	–
$T4$	250	35	10	10	10
$T5$	450	50	–	–	5

p_i indicates the period of the task T_i and e_i indicates its computation time. The period of each task is the same as its deadline. The entries in the R_i columns indicate the time duration for which a task needs the named resource in non-preemptive mode. Assume that after a task releases a resource, it does not acquire the same or any other resource.

[Internal Assessment — 10 Marks]