

OLLSCOIL NA hÉIREANN
THE NATIONAL UNIVERSITY OF IRELAND, GALWAY

SEMESTER II EXAMINATIONS 1999/2000

CT867

Embedded Systems

Professor D. Bell
 Dr. G. Lyons
 Dr. A. Shearer

Time Allowed: **Two Hours**

Answer question 1 and 2 others

1. What is the difference between a hard and a soft real-time application? How do these concepts change the way in which real-time tasks are scheduled.

A Real-Time Computer has to schedule 10 tasks that have three priority levels (1-3). Priority level 1 is the highest and *must* be scheduled by its dead-line. Level 2 *should* be scheduled by its dead-line and level 3 is a non real-time task with no specific deadlines but they must not be starved of resources by the real-time tasks. The table shows the arrival time, priority level, execution time and dead-line for these tasks. Describe a scheduling algorithm which will ensure that the maximum number of tasks get executed within their timing constraints.

Task	Arrival Time	Execution Time	Dead-Line	Priority
A	1	5	10	1
B	1	5	15	1
C	5	3	13	2
D	6	1	15	1
E	8	3	20	3
F	10	2	25	2
G	10	10	40	2
H	12	20	60	2
I	12	20	70	3
J	20	5	30	1

- 2 You have been asked to design a system to control the lock gates of a commercial port. The basic operation is to open and shut the gates when the tide is at a particular level. You must ensure that no ships (or people) are in the vicinity when the gates start to move. Analyse the system using a simple Petri net.
- 3 Describe the main features of a real-time operating system (RTOS). How does an RTOS differ from a conventional operating system? Illustrate your answer using the QNX operating system.
- 4 What are the main requirements of a real-time programming language? How does ADA differ from conventional languages such as FORTRAN and C? Is it possible to make C a more acceptable real-time language?
- 5 What makes a computer control system reliable?

You have been asked to design the flight control system for a small aircraft. The computer systems you have selected have a failure probability of 10^{-4} / year. Ancillary systems, e.g. hardware voting units, have a failure probability of 10^{-6} / year. How would you ensure that the final configuration has a total failure probability that is less than 10^{-10} / year?