University of Glasgow

Friday 22 May 2009
9.30 am – 11.00 am
(Duration: 1 hour 30 minutes)

DEGREES OF MSci, MEng, BEng, BSc, MA and MA (Social Sciences)

REAL TIME AND EMBEDDED SYSTEMS (M)

(Answer 3 out of 4 questions)

This examination paper is worth a total of 45 marks

You must not leave the examination room within the first half hour or the last fifteen minutes of the examination.

Section A

1. **(a)** Consider a system of four independent preemptable periodic tasks:

$T_1 = (8, 1), T_2 = (4, 1), T_3 = (12, 4)$, and $T_4 = (10, 1)$

All jobs have phase equal to zero, and relative deadline equal to their period. Can this system be scheduled on a single processor using the least slack time algorithm? Explain your answer.

[2]

(b) A *maximum schedulable utilisation* test can be used to determine if the system from part (a) can be scheduled using the rate monotonic algorithm. State the expression for the maximum schedulable utilisation of a system of n independent preemptable periodic tasks scheduled on a single processor using the rate monotonic algorithm. The utilisation of the tasks in part (a) is greater than the maximum schedulable utilisation, what does that say about the schedulability of the system?

[3]

(c) How would the schedule for the system from part (a) differ if scheduled using the deadline monotonic algorithm instead of the rate monotonic algorithm? Explain your answer.

[2]

(d) It is not always possible to determine the schedulability of a fixed priority system using simple maximum schedulable utilisation tests. An alternative schedulability test can be used in some cases, based on the concept of a critical instant and timed demand analysis. Explain what are critical instants, and outline how they can be used with time demand analysis to provide a schedulability test for fixed priority systems. Explain why critical instants cannot be used as the basis for a schedulability test for systems using dynamic priority scheduling.

[8]

2. **(a)** A major difficulty in real-time scheduling is caused by the presence of *aperiodic* or *sporadic tasks*. Explain what is the difference between these two types of task. Discuss the role of an *acceptance test* in scheduling such systems, and outline why it is desirable to run an acceptance test when jobs are released.

[4]

(b) How can sporadic jobs be scheduled in a fixed priority system including a number of periodic tasks? Your answer should explain how the sporadic jobs are scheduled, outline how the acceptance test for newly released sporadic jobs works, and explain how you can prove that the periodic tasks meet their deadlines.

[7]

(c) The other major cause of timing variation in real time systems is resource access control, since jobs can block while waiting to gain a lock on some resource. The *priority inheritance protocol* and the *priority ceiling protocol* are two ways of arbitrating access to a resource. Discuss which of these protocols would you prefer to use with a fixed priority system, and explain why you made that choice.

[4]

3. **(a)** The *controller area network* is an example of a network that enforces priority scheduling at the link layer, to make it possible to schedule predictable communications. Explain the operation of a controller area network, and describe how this priority scheduling is enforced.

[6]

(b) Networks that cannot manage traffic timing at the link layer often opt to provide enhanced quality of service by controlling the queuing discipline at the network layer, for example using *weighted fair queuing*. Describe the operation of the weighted fair queuing algorithm. Your answer should include a diagram showing the flow of packets through the router, and an explanation of how packets are queued and how the queues are serviced. You may need to refer to the "finish number" of packets, but you do not need to explain how the finish number is calculated

[6]

(c) From the viewpoint of applications running on the network, what are the advantages of weighted fair queuing, compared to an uncontrolled best effort service?

[3]

4. The recommended reading for this course included two papers: "Absolutely Positively on Time: What would it Take?" by Edward A. Lee (IEEE Computer, July 2005) and "The nesC Language: A Holistic Approach to Networked Embedded Systems" by David Gay *et al*. (Proceedings of the ACM Conference on Programming Language Design and Implementation, San Diego, CA, USA, June 2003). These papers suggest that the languages and programming models used for real-time and embedded systems are outdated and ineffective, and that a radical re-think is needed to effectively meet the challenges inherent in implementing such systems. Do you agree with this assessment? Discuss the issues raised by these papers, and considered in the lectures, and explain the reasoning for your answer.

[15]