



University
of Glasgow

**Monday 10 May 2010
9.30 am – 11.30 am
(Duration: 2 hours)**

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

**COMPUTING SCIENCE M:
REAL TIME AND EMBEDDED SYSTEMS**

(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 hour or the last half-hour of the examination.

Section A

1. (a) Consider a system of three independent preemptable periodic tasks, with no precedence or resource constraints, running on a single processor:

$$T_1 = (6, 2), T_2 = (12, 3), \text{ and } T_3 = (4, 1)$$

All jobs have phase equal to zero, and relative deadline equal to their period. Construct a cyclic schedule for these tasks, suitable for implementation on a clock-driven scheduler, and demonstrate that the system meets all its deadlines.

[2]

- (b) When implementing a clock-driven scheduler, it is common to use a schedule based around a fixed frame size, rather than one with arbitrary job durations. Explain why a schedule using a fixed frame size is desirable.

[2]

- (c) The tasks from part (a) are to be re-implemented on an operating system that supports priority-based scheduling. Which priority-based scheduling algorithm would you prefer: rate monotonic or earliest deadline first? Justify your answer.

[5]

- (d) It is not always possible to determine schedulability for a fixed priority system using maximum schedulable utilisation tests. An alternative schedulability test can be used in some cases, based on the concepts of time demand analysis and critical instants. Explain what is meant by the term “critical instant”, and outline how time demand analysis can be used to provide a schedulability test for fixed priority systems.

[6]

2. (a) Aperiodic jobs sometimes have to be scheduled as part of a system comprising mostly periodic tasks executed according to some fixed priority scheduling algorithm. These aperiodic jobs can be executed using *slack stealing*, or they can be executed by a *periodic server* task. Discuss how these two approaches to executing aperiodic jobs work, and highlight their advantages and disadvantages. [5]
- (b) Two simple types of periodic server are the polling server and the deferrable server. Describe how each of these algorithms works in terms of the parameters of the server task $T_s = (\phi, p, e, D)$, server release times, and the budget consumption and replenishment rules. [7]
- (c) The deferrable server is the simplest of a class of periodic servers known as *bandwidth preserving servers*. A key limitation of a deferrable server $T_{ds} = (p_{ds}, e_{ds})$ – removed in more sophisticated bandwidth preserving servers such as the sporadic server – is that it's not sufficient to treat the deferrable server as a regular task T_i with $p_i = p_{ds}$ and $e_i = e_{ds}$ when determining if the system is schedulable. With the aid of an example, illustrate why this limitation occurs. [3]

3. (a) It has been suggested that Internet Television (IPTV) services will never fully replace traditional cable, satellite, and broadcast TV services because the Internet does not provide the necessary quality of service guarantees. Discuss whether the lack of guaranteed quality of service is a significant issue for IPTV services, considering both live programmes (e.g. sporting events) and stored content (e.g. yet another re-run of *Friends*). [7]

(b) If you believe that lack of guaranteed quality of service is a significant issue for IPTV services, explain how quality of service might be added to IP networks, and what guarantees it might provide.

- *alternatively* -

If you do not believe lack of guaranteed service is a significant issue for IPTV services, explain how IPTV services can be made to work reliably over a network that does not guarantee real time performance. [8]

4. In his paper *Absolutely Positively on Time: What Would it Take?* (IEEE Computer, July 2005, pages 85-87), Edward A. Lee argues that “many recent computing advances do more harm than good when embedded computing systems absolutely must meet tight timing constraints”. Discuss this argument, comparing and contrasting the facilities provided by Real Time Specification for Java (and its associated virtual machine), with those in the traditional framework of C and a POSIX-compliant operating system. Outline how *you* believe programming models for real time and embedded systems should evolve in future.

[15]