Thursday, May 10, 2007
2:30p.m. – 4:15p.m.

University of Glasgow

DEGREES OF M.Sc., P.G. Dip., M.Sci., M.Eng., B.Eng., B.Sc., M.A. and M.A. (Social Sciences)

COMPUTING SCIENCE M:
REAL TIME AND EMBEDDED SYSTEMS

(Answer 3 out of 4 questions)

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

$$T_1 = (8, 1),\ T_2 = (6, 3),\ \text{and}\ T_3 = (12, 1)$$

All jobs have phase equal to zero, and relative deadline equal to their period. Is this system schedulable using the Earliest Deadline First algorithm? Explain your answer.

[2]

(b) The system from part (a) must be modified such that the relative deadline of T_3 is shorter than its period. What is the shortest relative deadline for T_3 that still allows the system to be scheduled? Explain your answer.

[4]

(c) The set of independent preemptable periodic tasks described in part (a) are to be re-implemented on a system that supports only fixed priority scheduling. Which scheduling algorithm would you prefer: rate monotonic or deadline monotonic? Justify your answer.

[1]

(d) The system from part (c) is to be implemented on a real time operating system that provides a scheduling interface conforming to POSIX 1003.1b. Outline the features of this interface, describe the scheduling policies that it provides, and explain the differences between those policies.

[9]

(e) Discuss how would you use the features of POSIX 1003.1b to implement the scheduling algorithm chosen in part (c).

[4]

2. **(a)** One technique for scheduling aperiodic jobs in a fixed priority system involves the use of a bandwidth-preserving server. Explain how such a server is used to schedule aperiodic jobs, and discuss why it is preferable to use a bandwidth-preserving server instead of running the aperiodic jobs in a background task.

[4]

(b) Two simple types of bandwidth-preserving 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) You are designing a system that must support a mix of periodic tasks and aperiodic jobs. You have the choice of using a polling server or a deferrable server to support the aperiodic jobs. Which would you use? Explain why.

[2]

(d) 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 new sporadic jobs works, and explain how you can prove that the periodic tasks meet their deadlines.

[7]

3. **(a)** You have been given the task of implementing a voice-over-IP application, to support interactive telephone calls over the Internet. The specification calls for the application to use UDP/IP as the transport protocol for the voice data, rather than TCP/IP. Explain why this is the correct choice of transport.

[4]

(b) On deploying the voice-over-IP application, you find that IP phone calls to some users fail, because they have a firewall that prevents UDP/IP traffic, but allows TCP/IP. You implement an alternative system, using TCP/IP for the voice traffic, to allow those users to communicate. Under what conditions do you expect the TCP/IP-based alternative to have acceptable real-time behaviour? Explain why.

[5]

(c) Some time later, you are involved in the implementation of a streaming Internet radio station. Will a TCP/IP-based transport work better for the radio station than for the interactive voice application? Justify your answer.

[3]

(d) Your answers to parts (a)–(c) of this question have touched on some problems that can occur when running real time applications over IP networks. These, and the other problems with such systems, occur because IP was not architected to provide timing guarantees to networked applications. Given this seemingly fundamental limitation, explain why so many companies are adopting voice-over-IP solutions, and discuss how IP networks could evolve to better support real time applications.

[8]

4. **(a)** 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 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.

[20]