



University  
of Glasgow

**Friday, 16 May 2014**  
**9:30 am – 11:30 am**  
**(2 hours)**

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

# **ADVANCED OPERATING SYSTEMS**

**Answer 3 out of 4 questions**

**This examination paper is worth a total of 60 marks.**

**The use of calculators is not permitted in this examination.**

**For examinations of at least 2 hours duration, no candidate shall be allowed to leave the examination room within the first hour or the last half-hour of the examination.**

**INSTRUCTIONS TO INVIGILATORS: Please collect all exam question papers and exam answer scripts and retain for school to collect. Candidates must not remove exam question papers.**

1. (a) Explain what is the difference between periodic, aperiodic, and sporadic tasks. Your answer should discuss when the jobs comprising the task will execute, and what is their deadline. [4]
- (b) Two ways of scheduling jobs from an aperiodic task are to use a background server or to use a periodic polling server. Describe these two approaches, highlighting their relative advantages and disadvantages. [6]
- (c) Systems that incorporate sporadic jobs generally include an acceptance test that is executed before accepting such jobs. What is the purpose of this acceptance test? [2]
- (d) A system comprises a mixture of periodic tasks and sporadic jobs. These are to be scheduled in a preemptive manner on a single processor, using the earliest deadline first (EDF) algorithm for the periodic tasks, with a sporadic server to handle execution of the sporadic jobs. Explain the operation of the acceptance test for newly arrived sporadic jobs in such a system. Your answer should clearly state the condition that must be true to guarantee that both the periodic tasks and sporadic jobs can be scheduled, and outline how the acceptance test that checks that condition can be implemented. Comment on the optimality of the acceptance test. [8]
2. (a) Java requires objects to be allocated on the heap, but C and C++ allow objects and complex data structures to be stored on the stack. The use of stack allocation can make automatic memory management much simpler for certain classes of objects. What are these objects, and why is their management simplified? [5]
- (b) We discussed the paper by Bacon et al., on “A real-time garbage collector with low overhead and consistent utilization” (Proc. ACM PoPL, 2003). This paper proposes a real-time garbage collector that executes as a periodic task, scheduled alongside the other tasks in the system. Discuss the trade-off between predictable timing and collection efficiency for this garbage collector. Outline any constraints it imposes on the memory allocation behaviour of the system to ensure correctness. [5]
- (c) Garbage collected languages, such as Java, generally implement their garbage collector as part of a virtual machine, or other run-time support library, running as a user-space application. It has been argued that this is sub-optimal, and that the garbage collector should be provided as an operating system service that is available to all applications. Explain the problems with user-space implementations, and highlight the advantages and disadvantages of providing a garbage collector as an operating system service. [10]
3. (a) Many modern multicore processor systems use a non-uniform memory access (NUMA) architecture. Explain in what way the memory is non-uniform, and describe how the non-uniform nature of the memory is visible to software running on such processors. [4]
- (b) We discussed the multi-kernel model of operating systems for multicore processors, where each core runs its own kernel, and all communication is by explicit message passing. Outline the advantages and disadvantages of such an operating systems architecture, and discuss how it relates to the structure of modern multicore systems. [6]

- (c) Synchronous message passing systems force the sender and receiver of a message to rendezvous in time to exchange the message. In asynchronous message passing systems, the sender continues after sending the message, which is buffered for later delivery; the buffer may be unbounded and able to grow arbitrarily large, or may have a finite capacity. Discuss how the different approaches affect the concurrency and reliability of a system. [10]
4. (a) Message passing systems are often said to avoid many of the problems inherent in lock-based concurrency. However, message passing systems can still deadlock, and race conditions can still occur. Explain how deadlocks and race conditions can occur in message passing systems. Discuss whether you think they are more or less likely to occur than in systems using lock-based synchronisation, justifying your answer. [12]
- (b) An alternative to using message passing is to use atomic transactions to control and manage concurrency. The code running in an atomic transaction must be *referentially transparent* and *revocable*. Explain what these properties mean, and why they are necessary for a transaction. [8]