University of Glasgow

Thursday, 10 May 2012
2.00 pm - 4.00 pm
(2 hours)

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

ADVANCED OPERATING SYSTEMS (M)

Answer 3 out of 4 questions

This examination paper is worth a total of 60 marks.

You must not 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 return to School together with exam answer scripts

1. **(a)** Digital process control systems generally operate in a control loop, periodically sampling the state of the system, comparing that with a reference input, performing a control law calculation based on the difference, and then driving the state of the system using some form of actuator. Effective control depends on the reference input, correct implementation of the control law, and on the accuracy of the sampled measurements of the system state. Discuss the trade-off between the sampling period of the measurements, processor overhead, and stability of the system. [4]

(b) The execution of process control systems of the type outlined in part (a) can be modelled as a periodic task with four timing parameters (two of which are optional, and have assumed default values in many tasks). What are those parameters? [2]

(c) The timing constraints of a system can variously be described as *hard* or *soft* real-time. Explain what is meant by these terms. Discuss why most real systems should be considered to be on the spectrum between fully hard real-time or completely soft real-time, rather than at one of the extremes. [4]

(d) A range of algorithms exist that can be used to schedule periodic tasks in a priority-driven system with a single processor. These include the *rate monotonic* algorithm and the *deadline monotonic* algorithm. Explain how these algorithms assign priorities to tasks. Discuss under what circumstances you would prefer one algorithm to the other, and when it makes no difference which algorithm is chosen. [10]

2. **(a)** The jobs in a real-time system may execute as part of a *periodic* task, or in an *aperiodic* or *sporadic* manner. Describe the behaviour of these three types of task in terms of the schedule of their release times and their deadlines. [3]

(b) Aperiodic jobs sometimes need to be scheduled as part of a system comprising mostly periodic tasks executed according to a fixed priority scheduling algorithm. These aperiodic jobs can be executed using slack stealing, or they can be executed by a periodic server task. Explain how these two approaches to executing aperiodic jobs work, and highlight their advantages and disadvantages. [5]

(c) The *deferrable server* algorithm can be used to implement a periodic server task for the purpose of scheduling aperiodic or sporadic jobs. With reference to the underlying periodic task, and the budget consumption and replenishment rules, explain how a deferrable server works. Discuss how you can determine the schedulability of fixed- and dynamic-priority systems containing a deferrable server. [7]

(d) The deferrable server is one example of a bandwidth-preserving server algorithm. Another example is the sporadic server algorithm, which trades algorithm complexity for ease of proving schedulability. Do you believe the complexity of these servers is worthwhile for periodic systems that include some aperiodic jobs, or should those jobs be scheduled as simple, low-priority, background jobs? What about systems of periodic tasks that also contain some sporadic jobs? Justify your answer. [5]

3. **(a)** Some programming languages, for example C++, support allocation of objects both on the stack and on the heap. Other languages, such as Java, allocate all objects on the heap. Discuss the implications of this choice for memory management. [5]

(b) One widely used approach to automatic memory management is to add reference counts to objects, and to update the runtime system to maintain these counts, and reclaim the memory allocated to an object when the reference count for that object is decremented to zero. Outline the main advantages and disadvantages of reference counting as a means of automatic memory management. [5]

(c) Garbage collection and virtual memory can interact in ways that cause extremely poor system performance. Explain what is happening when this occurs, and why. [6]

(d) Explain why it is difficult to build a garbage collector for the C programming language, but relatively straight-forward for Java. [4]

4. **(a)** Modern computing devices increasingly include multiple processor cores, and these cores typically operate with a non-uniform memory access (NUMA) model. Outline what is meant by a NUMA model, and explain to what extent a NUMA system hides the underlying memory heterogeneity, and how the effects of the non-uniformity are visible to the programmer. With reference to the organisation of the underlying hardware, explain why such systems exhibit non-uniform memory access. [6]

(b) Some computing systems are designed to have separate memory regions available to each processor, with either no shared memory, or only limited cache coherency. Using a message-passing communications protocol, it *is* possible to implement the traditional abstraction of threads with shared memory across the heterogeneous cores of a processor that does not have hardware-based cache-coherent shared memory. Discuss the advantages and disadvantages of faking shared memory in this way, explaining your reasoning. [4]

(c) Rather than using hardware memory protection to isolate processes, some research operating systems (e.g., Singularity) employ the concept of software isolated processes. Explain what is a software isolated process, and describe how the isolation between processes is enforced. Discuss the extent to which you believe software isolation is sufficient to enforce separation between processes. [10]

END OF QUESTION PAPER