University of Glasgow

Tuesday, 28 April 2015
09:30 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)** Uncontrolled resource access is problematic in real-time systems, since the schedule can be disrupted by tasks that block waiting to access a resource. One way to manage access to resources and control the blocking time is to use the priority ceiling protocol. Describe this protocol, explaining how the scheduling, resource allocation, priority ceiling, and priority inheritance rules work. [7]

   (b) Discuss under what conditions the priority ceiling protocol can be used with dynamic priority scheduling algorithms. Describe what changes need to be made to the protocol to support dynamic priority scheduling. [3]

   (c) What is the maximum amount of time for which a job may block when the priority ceiling protocol is in use? [2]

   (d) Access to hardware resources is controlled by device drivers. Discuss the advantages and disadvantages of using a higher-level programming language, for example OCaml, Rust, or C++, to implement device drivers, compared to the more traditional, C-based, device driver framework implemented in most common operating systems. [8]

2. **(a)** A common form of automatic memory management is to use reference counts on data objects. Explain how reference counting works, and outline under what circumstances it can lead to memory leaks. Discuss the relative advantages and disadvantages of reference counting compared to garbage collection. [10]

   (b) 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]

   (c) Virtual memory can also interact poorly with real-time systems, because it can introduce unpredictability into process execution times. Given this, and the interactions with garbage collection outlined in part (b), discuss whether you think the benefits of virtual memory outweigh the problems it causes. [4]

3. **(a)** Two alternative abstractions for concurrency are transactional memory with automatic roll-back and retry, and communication via message passing, where copies of immutable data are passed between shared-nothing processes. Which of these abstractions do you prefer as a basis for systems programming? Justify your answer, and explain your rationale for making this design choice. [10]

   (b) One popular implementation of message passing is in the Erlang programming language. Erlang adopts a dynamically typed model for message passing, where any message can be sent to any process. The Singularity operating system also uses message passing, but requires channels to be statically typed, with the compiler ensuring that a process will only receive messages corresponding to some contract. Discuss the advantages and disadvantages of the two approaches. [6]

   (c) One of the key advantages claimed for transactional memory over lock-based concurrency control is that transactions maintain correctness when composed into larger units. With the aid of an example, show how composition of correct lock-based code can lead to race conditions that might affect correctness. [4]

4. **(a)** The Accelerator system extends C# with a data parallel array type that can be used to transparently offload array-based computations to a GPU. Describe how Accelerator uses the data parallel array type to manage the GPU, perform calculations, and safely transfer data between the GPU and the main processor. [10]

(b) Compare and contrast the approach to offloading computations to the GPU chosen by the Accelerator, with the approach to GPU computations adopted in OpenCL. Discuss the extent to which an Accelerator-like model can replace languages like OpenCL. [10]