University of Glasgow

Monday, 14 May 2018
9:30am - 11:30am
(2 hours)

DEGREES OF 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.

The use of calculators is not permitted in this 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)** Briefly describe what is Moore's Law and what is Dennard Scaling. Explain why the combination of Moore's Law and Dennard Scaling allowed processor clock rates to grow significantly between the late 1970s and early 2000s. [10]

(b) The growth in processor clock rates dramatically slowed in the mid-2000s. Explain why this occurred. [6]

(c) While processor speeds stalled in the mid-2000s, the bit-rate of local area network links, such as Ethernet, has continued to increase. Explain why this increase in network bit-rate, relative the processor clock rate, makes it difficult to implement a high-performance network protocol stack. [4]

2. **(a)** Most commonly used operating systems arrange the virtual address space of a process so that the memory used by the stack is separate to the memory used by the heap. Outline what data is stored on the stack and what is stored on the heap. Explain why the stack and the heap need to be separate regions of memory. [6]

(b) Automatic management of heap allocated memory can be supported using either reference counting or garbage collection. Give two advantages and two disadvantages that reference counting has in comparison to using garbage collection. [4]

(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)** We discussed the actor framework for programming concurrent systems. In this framework, programs comprise a set of shared-nothing processes that can communicate by the exchange of immutable messages. Many actor-based systems adopt a let-it-crash approach to error handling, whereby responsibility for handling failures in a task is pushed out to a separate supervisor task. Discuss whether the let-it-crash model is an appropriate way of providing robustness in a massively concurrent system, or if in-process error detection and recovery is more suitable. Your answer should include an explicit discussion of the trade-offs between the two approaches. [8]

(b) 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]

4. **(a)** We discussed the paper by Kamp and Watson on "Jails: Confining the omnipotent root." (Proc. SANE, 2000). This describes a container-based approach to virtualisation, as was originally developed for FreeBSD, that has since been ported to several other operating systems. Describe what type of virtualisation is provided, and outline the key properties of the *jails* environment. [7]

(b) Linux implements a container system that is a generalisation of the ideas in FreeBSD *jails*, and provides similar light-weight virtualisation. Products such as *Docker* build on this container system, and add new features that make Linux containers easier to use than FreeBSD jails. Describe what are the key features of Docker that simplify creation of Linux containers, and how these differ from the features provided by FreeBSD jails. [6]

(c) Compare and contrast the light-weight virtualisation provided by Linux containers or FreeBSD jails with that provided by a hypervisor such as Xen. When would you choose a container-based environment, and when would a hypervisor-based virtualisation be more suitable? [7]

END OF QUESTION PAPER