Monday, 27 April 2020, 09:00 BST
(24 hour open online assessment – Indicative duration 2 hours)

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

ADVANCED SYSTEMS PROGRAMMING (M)
COMPSCI 5083

Answer 3 out of 4 questions

This examination paper is worth a total of 60 marks.

1. **(a)** The Rust programming language uses a region-based approach for memory management. This introduces complexity into the type system, since it must track ownership and borrowing of data, and makes implementation of certain data structures problematic. In return, it ensures predictable and safe memory management. Other programming languages make a different trade-off, offering a simpler type system but providing fewer guarantees around memory safety or less predictable memory management. Do you think Rust makes the right trade-off for its intended uses cases and given the expected skills and experience of developers? Justify your answer. [8]

   (b) State machines are an important aspect of systems programming. They encode the state of a resource, and describe what operations can be performed on that resource in what context. Using a combination of pseudo-code to illustrate key concepts and English prose, describe *one* way in which a state machine can be represented in Rust. Discuss the advantages and disadvantages of your chosen approach. [12]

2. **(a)** Systems programs have generally been written in memory unsafe languages, such as C. It is generally accepted that this is problematic, and that it's desirable to move to memory safe languages where possible. Discuss whether it is possible to write an operating system entirely in a memory safe language, highlighting areas where this would be difficult. [5]

   (b) Is it feasible to write an entire operating system in a garbage collected language? Discuss the advantages and disadvantages of such an approach. [5]

   (c) Studies have shown that the majority of security vulnerabilities in deployed systems relate to a lack of memory safety. To what extent do you think it is currently feasible to use only memory safe languages for systems programming? Explain what are the costs and benefits inherent in making such a change to the implementation language. Discuss what is stopping the industry from moving entirely to memory safe languages for systems programming. [10]

3. **(a)** Programming languages, run-times, and operating systems can allow I/O operations to be performed synchronously or asynchronously. Explain what is the difference between these two models. Discuss why support for asynchronous I/O is frequently considered to be beneficial. [5]

   (b) One way of implementing asynchronous I/O operations is to structure the code as a set of *coroutines*. In this model, operations that may block return a `Future<T>` representing a value of type `T` that will become available at some later time. The runtime system for such languages includes an `await` operation, that allows a function to wait for a result to become available without blocking the thread, by passing control to other coroutines while waiting. Discuss whether you think this is a good approach to writing asynchronous code. Explain what are the strengths and weaknesses of this approach to structuring code. [7]

   (c) 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. [8]

4. (a) The recommended reading for the course included Shapiro's paper entitled "Programming language challenges in systems codes: why systems programmers still use C, and what to do about it" (Proc. ACM Workshop on Programming Languages and Operating Systems, San Jose, CA, USA, October 2006). This suggests that the C programming language is not appropriate for writing systems code, and outlines some new programming language features that would improve systems programming. The lectures covered other topics in this space, and introduced you to the Rust programming language, as an example of a modern language that attempts to innovate in the field of systems programming.

Do you agree with the thesis of this paper and the lecture discussion? Discuss the extent to which you believe that changing the programming language will help to address the challenges in building secure, high performance, and robust systems programs. Outline what features of a programming language or runtime you consider important for supporting systems programming, and what are harmful. Illustrate your answer using examples from systems programming, including features that might help or hinder their implementation, to help make your argument. [20]