University of Glasgow

Monday, 9 May 2022
14:00-15:30 BST
Duration: 1 hour 30 minutes
Additional time: 30 minutes
Timed exam – fixed start time

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

ADVANCED SYSTEMS PROGRAMMING (H)
COMPSCI 4089

Answer 3 out of 4 questions

This examination paper is an open book, online assessment and is worth a total of 60 marks.

1. **(a)** Modern programming languages and runtime environments place a strong emphasis on support for concurrent and distributed programming. With reference to trends in computer systems design, and the environment in which systems programs operate, discuss why this might be the case. [7]

   (b) The lectures presented statistics showing that around two-thirds of reported software security vulnerabilities relate to the lack of memory safety in certain programming languages. Briefly explain what is memory safety, and discuss why lack of memory safety can result in security vulnerabilities. [5]

   (c) 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]

2. **(a)** It's common for programming languages to support concurrency by providing multiple threads of execution within a single address space, along with locks to control access to shared mutable data. This is the model adopted by the C programming language with the *pthreads* library, and by Java with synchronised methods, for example. Discuss what are the problems inherent in this programming model, considering in particular correctness of the resulting code and composition of operations. Use pseudocode fragments to provide examples that illustrate key points. [5]

   (b) As an alternative to the thread-based programming model, some languages offer support for concurrency via transactions with automatic roll-back and retry, or through message passing. Systems that use message passing can be further subdivided into those that avoid concerns about shared mutable state by making messages immutable, and those that avoid such problems by tracking ownership to prevent shared access of mutable data. Such languages and systems were discussed in the lectures, and in the recommended reading for the course. Outline the advantages and disadvantages of each of these three approaches for systems programming. Discuss, with justification, which approach you consider to be the most promising approach for improving systems programming; if you think none are promising, explain why. [15]

3. **(a)** A number of programming languages have invested significant effort to provide language and runtime support for asynchronous programming. Describe what is asynchronous I/O and how it differs from synchronous I/O. Discuss the advantages and disadvantages of providing language and runtime support for asynchronous programming. [10 marks]

   (b) We discussed the concept of *type-driven development* in the lectures. In this approach to software development the initial focus is on defining appropriate types to model the solution, then on writing the function prototypes for key type transformations and major operations of the program, and only then refining and completing the implementation. That is, the structure of the design and implementation is determined by the type-level model of the solution space. This is at odds with certain other implementation strategies, where the focus is first on the actions and operations, treating the types as more of an implementation detail. Discuss whether you believe type-driven development using a strongly typed language is a good approach to software development or not. Justify your answer, highlighting *both* strengths and weaknesses of the approach. [10]

4. **(a)** The recommended reading included Shapiro's paper entitled "Programming language challenges in systems codes: why systems programmers still use C, and what to do about it" (ACM Workshop on Programming Languages and Operating Systems, San Jose, CA, USA, October 2006) and "The bugs we have to kill" by Bratus, Patterson, and Shubina (USENIX ;login:, August 2015). These papers suggest that the way we write systems programs has to change if we are to successfully develop secure, highly concurrent, and networked systems in the future. In particular, the authors suggest that we should use strongly typed, memory safe, programming languages to improve overall robustness of the code, that those languages need to give control over data layout and memory access, and that we need to pay special attention to input parsing and handling of untrusted data.

Do you agree with the thesis of these papers? Discuss the extent to which you believe that changing the programming language, and using better tools to parse and process untrusted input, will help to address the challenges inherent in building secure, highly concurrent, networked systems. Discuss what programming language and runtime features, and what features of parsing tools, you consider important to support secure systems programming, and what are harmful. Give examples to illustrate key points. [20]