University of Glasgow | School of Computing Science

Assessed Coursework

Course Name	**Advanced Operating Systems (M)**			
Coursework Number	**Exercise 3**			
Deadline	**Time:**	**4:30pm**	**Date:**	**29 February 2016**
% Contribution to final course mark	**12%**		**This should take this many hours:**	**16**
Solo or Group ✓	**Solo**	✓	**Group**	
Submission Instructions	**Submit via the dropbox outside the Teaching Office in Lilybank Gardens**			
Who Will Mark This? ✓	**Lecturer** ✓	**Tutor**	**Other**	
Feedback Type? ✓	**Written** ✓	**Oral**	**Both**	
Individual or Generic? ✓	**Generic**	**Individual** ✓	**Both**	
Other Feedback Notes				
Discussion in Class? ✓	**Yes**	**No** ✓		
Please Note: This Coursework cannot be Re-Done				

Code of Assessment Rules for Coursework Submission

Deadlines for the submission of coursework which is to be formally assessed will be published in course documentation, and work which is submitted later than the deadline will be subject to penalty as set out below. The primary grade and secondary band awarded for coursework which is submitted after the published deadline will be calculated as follows:

(i) in respect of work submitted not more than five working days after the deadline
   a. the work will be assessed in the usual way;
   b. the primary grade and secondary band so determined will then be reduced by two secondary bands for each working day (or part of a working day) the work was submitted late.

(ii) work submitted more than five working days after the deadline will be awarded Grade H.

Penalties for late submission of coursework will not be imposed if good cause is established for the late submission. You should submit documents supporting good cause via MyCampus.

Penalty for non-adherence to Submission Instructions is 2 bands

You must complete an "Own Work" form via

http://www.dcs.gla.ac.uk/socs-online for all coursework

UNLESS submitted via Moodle

Advanced Operating Systems (M): Exercise 3

Dr Colin Perkins

16 February 2016

There have traditionally been three approaches to memory management in programming languages and systems: either the system provides a tracing garbage collector, or it provides reference counting, or it relies on the programmer to manually allocate and free memory. There are some advantages and disadvantages to each approach, but the fact that all three are widely used suggests that none are suitable for all problem domains.

An alternative automatic memory management scheme that is gaining attention is *region-based memory management*, as is used in the recently released Rust programming language (`http://rust-lang.org/`). Region-based memory management tries to achieve effective automatic memory management, without the indeterminism and overheads of a garbage collector, by tracking ownership of data, and using this to automatically deallocate objects when they go out of scope. In this final summative exercise, you will study region-based memory management, and its advantages and disadvantages compared to other approaches.

There are three parts to this exercise. Firstly, research and write a report describing how region-based memory management works in the Rust programming language, and how Rust manages ownership of data, including different pointer types, and borrowed references to data. Your report should describe when memory allocations occur, how Rust tracks ownership of data, and when it deallocates memory. *[10 marks]*

Then, compare and contrast region-based memory management, as used by Rust, with the more traditional behaviour of garbage collected languages, and highlight the relative advantages and disadvantages of the two approaches. With the aid of sample code fragments, discuss the types of program that Rust makes easy to write, and the types that are difficult, or impossible, to write, explaining what language design decisions make them easy or difficult. *[10 marks]*

Finally, the developers of Rust claim that it is suitable for systems programming. We earlier discussed some of the challenges of systems programming (e.g., Shapiro's paper, referenced in lecture 7). Define what *you* consider to be systems programming, and discuss whether Rust succeeds in being a good choice in that area. Discuss also whether Rust is suitable for use as a general purpose programming language. Justify your answers, with reference primarily to the way Rust approaches automatic memory management, but also highlighting other features of the language if appropriate. *[10 marks]*

You should submit a single printed report covering the three topics outlined above. A mark out of 30 will be assigned to each submission, weighted as noted, and will be converted to a percentage and used to assign a band on the University's 22 point scale. There is no single correct answer for this exercise, and answers can legitimately argue for or against the suitability of Rust, or the appropriateness of the features discussed. Marks will be assigned for technically correct descriptions of the various features, for properly researched arguments that have well-reasoned justification, and for coherent and clear argumentation.

Print your report on A4 paper, formatted in two columns, using the Times Roman font in 10pt, with 1 inch margins. If you use LaTeX to prepare your document, the following preamble will format your submission appropriately:

```
\documentclass[10pt,a4paper,twocolumn]{article}
\usepackage{fullpage}
\usepackage{newtxtext}
\usepackage{newtxmath}
\begin{document}
...
```

(your are not required to use LaTeX). Your report must not exceed six pages in length, including all figures, tables, code samples, and any references. Length is not an indication of merit: if you can cover the required material in less than six pages, then please do so.

You must submit your report before 4:30pm on 29 February 2016. Following the code of assessment, late submissions will be accepted for up to 5 working days beyond this due date. Late submissions will receive a two band penalty for each working day, or part thereof, the submission is late. Submissions that are received more than five working days after the due date will be awarded a band of H.

A drop box will be available for submissions outside the Teaching Office in Lilybank Gardens. Submissions are only accepted via that drop box. This problem set is worth 12% of the mark for this course. Make sure you include your matriculation number in your report, and submit a statement of originality. Submissions that do not follow these submission instructions will be given a two band penalty.