Challenges in Systems Programming: Energy Efficiency

Advanced Systems Programming (H) 2022-2023 – Laboratory Exercise 2
Dr Anna Lito Michala & Dr Colin Perkins, School of Computing Science, University of Glasgow

1 Introduction

The Advanced Systems Programming (H) course uses the Rust programming language (`https://rust-lang.org/`) to illustrate several topics in systems programming. However, as programmers you will need to develop critical thinking skills that challenge when a language is suitable for the software being developed. To assist this learning, this exercise is introducing the concept of energy efficiency of a language. **This is a formative exercise and is not assessed.**

The use of a programming language can affect the energy efficiency of the final software solution. This is due to the often unseen components of the developed software, like the compiler. These affect the final body of assembly instructions generated, the number of data transitions in an out of CPU, as well as the basic memory footprint. This exercise is exploring these issues through reading and discussion in groups.

The expected timeline for this laboratory exercise is that you complete the all the tasks in Section 2 by the end of the timetabled lab session on Monday 16 January 2023.

2 Formative Exercises

To improve your critical reasoning about programming languages such as C and Rust, complete the following exercises. These exercises are not assessed, and you do not need to submit your solutions.

1. Organise yourselves into groups of 5 or 6 (10 mins).
2. Use google or the library's resources to answer the question: "why is C more energy efficient compared to other languages?" (15 mins - individual).
3. Discuss your findings and link them to the main arguments of the paper that was presented in the lecture today on: "why do we still use C?" here (15 mins - group).
4. Each group should present their main argument as a quick stand up presentation (10 mins).
5. The lab will end with an open discussion about the variety of language features and issues that the C language brings (5 mins).

Discuss your findings with the lecturer or lab demonstrators to make sure you are working towards the task and allow everyone in the group to speak and contribute.

3 Next Steps

At this point you should have a very basic introduction of programming in Rust. We will dive into the specifics in Lecture 3 at which point we will start working on coding examples. **However, you are expected to learn the basics of**

programming in Rust in a self-directed manner. The laboratory exercises and lectures will direct you to appropriate resources, but on onus is on you to study the material.

Lecture 3 will provide a basic understanding of the language. The later parts of the course will then consider the use of the Rust type system to help model the problem domain and help ensure correctness of the solution (Lecture 4), consider how it handles memory allocation and data ownership (Lectures 5 and 6), how it addresses the challenges of concurrency (Lecture 7), and how it supports asynchronous programming (Lecture 8). To prepare for this material, you should begin to review the remaining material in the online Rust book, and practice programming in Rust to ensure you start to have a good understanding of the language.