

COMPUTING HONOURS PROJECT (COMP10034)

Marking Sheet for Interim Reports

Please mark the Interim Reports out of 100%.

Assessment of the Interim Report

Student Name: Kyle Christie

Banner ID: B00415210

Campus: Paisley

The marking scheme for the Interim Report is as follows:

Background/Literature Review and Referencing	20/30
Preliminary Work/Creative Accomplishments	24/30
Progress and Further Planned Work	25/30
Structure, Presentation, Style of the Report	6/10

Agreed Mark with the Moderator: 75 (out of 100%)

Student Feedback (Supervisors only)

This is a well-written report. The technical ambition and achievements are especially impressive. The project is on track for an impressive conclusion. There is nevertheless still room for improvement.

Aim for more research on the memory safety topic in general. Perhaps consider both a modern and historical perspective; including garbage collection and reference counting. Some approaches will focus on new languages and language features, while other will focus on standalone libraries. The ACM SIGPLAN International Symposium on Memory Management (ISMM) may be a useful resource. You might consider this from a bilateral, pre and post Rust, perspective. Microsoft's Project Verona should not be missed; but also discuss the hardware approach of the ARM/Cambridge CHERI project.

Work-so-far is useful and highly readable, although somewhat episodic and diary-like. This is common in an interim report, but ultimately you should aim to tell the story of your work in the *best* order, not the chronological order you completed the work. Aim to make the writing less personal.

A table or diagram summarising upcoming tasks and their expected durations would have been useful (e.g. a Gantt chart).

Refer to each Figure within the text. At least figures 3, 5, 7, 8, 9 and 10 are missing

this.

Remove numbering from initial pages. Start numbering after the abstract.

Perhaps add an initial sentence or so to the abstract to describe memory safety. Also use the word "security" somewhere. The abstract also says that you are writing a "Linux kernel module". Perhaps you could say that it is (also) a driver?

Indent the first paragraph of each section or subsection.

"Next, Drivers have..." - perhaps "Furthermore" could be better here; or "Secondly". Also, "drivers" should not be capitalised.

When you say that there has been "little or no change" in driver ABI/architecture, it is implied that you are talking about Linux drivers. Perhaps either make that explicit; or include a similar reflection on Windows/Apple drivers.

"...Linux Device Drivers Training series by YouTube Channel 'Karthik M' (Karthik M, 2014)."

As you include the citation here, you can reduce the in-text referencing. Perhaps: "...Linux Device Drivers Training series (Karthik M, 2014)."

"It is clear that Drivers, ..."

->

"It is therefore clear that drivers, ..."

Perhaps reduce the size of the Figure 3. logo.

"(Rust Community, n.d.)" - it's not clear to me what n.d. means.

When you start to discuss Rust in Section 2.1, I feel that the package manager is introduced too soon. As it is somewhat auxiliary to the core offering, perhaps it could come a little later?

Oatman (2022) is cited on 3/4 consecutive sentences on page 9. Perhaps one can be enough.

"rust toolchain, code of all ages..."

->

"rust toolchain. Code of all ages"

"...no defined memory model thus has..."

->

"...no defined memory model, thus has..."

"implements a Stack and Dynamic Heap within..."

->

"implements a stack and dynamic heap within..." (and so too for other uses of stack and heap throughout)

"exceptions; A manually created box and when..."

->

"exceptions: a manually created box; and when..." (i.e. use semicolons for "lists" within your prose, rather than commas.)

"are applied both the stack and heap;"

->

"are applied both the stack and heap:" (present using a colon)

"with the `move` function..." - could another font be used for inline code names rather than backticks?

"A variables lifetime begins at initialisation..."

->

"A variable's lifetime begins at initialisation..." (apostrophe for ownership, or a missing letter)

"which is supposed to outside the..." - the grammar here seems broken.

"...it is possible for the programs data structure..."

->

"...it is possible for the program's data structure..."

"..."known as a 'user-after-free' vulnerability."

->

"..."known as a 'use-after-free' vulnerability."

"...make it available to others and to act as a backup in the case that my workstation broke down." Perhaps elide such minor details.

Figure 5: feel free to edit this code. For example, the comments in `hello_init` could perhaps fit on two lines. You might also consider whether the text that explains Figure 5 might mean that comments in the (actual) code are not necessary within a report.

I think the input from Jonathan Blow and Dave Plummer could be better presented. Currently, it is simply slipped in to a paragraph listing arbitrary activities you have completed. Perhaps there will be somewhere else in your final dissertation, which is focused on the topics they are addressing. Even the literature review could accommodate a short quote, in context, from each of them.

"...a Hello World driver..." - Perhaps put "Hello World" in quotes in such scenarios.

The large figures on pages 15, 16 and 17 slightly interrupt the flow of text.

"'Scull' (an example contained within Linux Device Drivers 3)." - please cite the book as with other citations.

Aim for a consistent width in appendix code example frames.