COMP3151/9154 Assignment 2
Due: 9:00am Thus Aug 3, 2023
20 Marks

This assignment is to be done in pairs. Individual submissions will be accepted but are strongly discouraged.

This assignment is an open-ended research task where you are asked to independently investigate modern concurrent programming styles in a modern programming language, solve a problem using that language, and present your solution in a pedagogical video.

1 Language Selection

Choose one of the following languages:

- Scala (with Akka)
- Kotlin (with Akka)
- Elixir
- Erlang
- Go
- Dart
- Rust
- Swift
- Clojure (with STM)
- Haskell (with STM)

The above list deliberately excludes legacy languages such as JavaScript, Python, C and C++ since they do not support concurrency beyond very basic locking constructs, or they suffer from other significant limitations in concurrent programming. (Akka is also available with Java, but this has been excluded in order to encourage you to look beyond what you already know.)

2 Task Description

You must select a well-known concurrency program or algorithm that demonstrates clearly the style of concurrent programming you have chosen to investigate.

The problem you select may be one of the ones covered in class or in homework, or some other problem in concurrent programming.

Thoroughly research the given problem and examine solutions presented in academic literature and elsewhere. Also thoroughly research the concurrency model you have selected, and implement this algorithm or problem in your chosen language. While you do your research, maintain a list of sources (using a standard referencing format such as ACM or IEEE).

Then, create a video (maximum length 15 minutes) that covers:

- The concurrency problem you are solving.
- The concurrency constructs in the language you have selected.
- An explanation of how the concurrency constructs of the selected language help with this problem.
- A demonstration of your solution.

The purpose of the video is to be **pedagogical**. Assume that the audience for your video is other members of this class, and you are presenting in order to provide additional course content. You should therefore assume that content already covered in class need not be covered in detail (although brief reminders may help). The presentation should focus on introducing some new content not already covered in class or previous homework or assignments.

You will be evaluated on how well your video uses the concurrency problem to motivate and explain the style of concurrent programming you are investigating.

You do not need to spend a significant amount of time discussing syntax of the given programming language, or giving a tutorial in programming in the given language in general. Rather, focus on how the language provides support for concurrent programming. In particular, you should identify what the language does to support concurrent programming, with emphasis on ideas that have not already been covered in class. (In particular, *exclude* semaphores and monitors from discussion!)

It is probable that your selected language will support a number of different concurrent programming styles. You only need to select the style that best demonstrates your problem solution.

The video should be posted as an unlisted video on YouTube (if there is a problem with this service, email Frank Jia), and a hyperlink provided in your Assignment 2 submission. The duration of the video should be no longer than 15 minutes. Longer submissions will only be evaluated on the first 15 minutes.

You may not use artificial means (speeding up the video) to fit your video content into the allotted time. If your video is too long, cut stuff out.

You must present your content in a clear, methodical, structured way.

Both team members should be heard in the video. For example, one member could introduce the problem, and another could discuss the solution.

You may use slides, screen captures, whiteboard videos or any other teaching or video props. You may submit supplementary files along with your video and your problem solution, and this data will be made available to viewers of your video.

You must ensure that the technical quality of your video is sufficient to make it watchable and enjoyable. Professional quality recording is not expected, but please try to make sure all audio is recorded in a quiet environment and that volume levels are normalised across the video.

3 Peer Assessment

After the due date for the assignment, all students will be required to view a number of other student videos, and provide feedback comments. These peer assessments will be **qualitative** – you will not award marks, but instead offer comments and and answer some evaluation questions for each video. These assessments will be used to assist in determining the final grade.

This peer assessment will be due on Friday of Week 10. It should not take more than an hour of your time. 10% of your mark is awarded for completing this peer assessment.

4 Assessment

Marks will be allocated with the following weighting:

- (1 mark) Your explanation of the problem.
- (5 marks) Your explanation of the concurrency constructs in the chosen language.
- (5 marks) Your explanation of how these constructs help to solve the chosen problem.
- (5 marks) Your submitted code and the demonstration in the video.
- (2 marks) The overall quality of presentation (from a pedagogical and technical perspective) of your video.
- (2 marks) Completion of the peer assessment task.

5 Deliverables

assignment2.txt A text file containing:

- A hyperlink to the video on YouTube.
- A list of your sources in a standard citation format.

assignment2.zip A ZIP format archive containing the code for your problem solution and any other supplementary material you wish to include for viewers of your video.

partner.txt A file with a single line on the format `z<digits>`. This must be the zID of your group partner. For individual submissions, write your own zID.

6 Submission

Only one member of the pair should submit. Work out who will submit ahead of time. Duplicate submissions are extremely annoying to sort out!

Submit your work by typing:

```
give cs3151 Assgt2 assignment2.txt assignment2.zip partner.txt
```