Assignment

Dr Carey Pridgeon

January 21, 2019

1 Module Assessment Preamble

2 Module Learning Outcomes Met by this assesment:

- 1. Describe the issue of data consistency in Multi-Threaded applications.
- 2. Apply Concurrency to common tasks
- 3. Design and Implement a basic Concurrent application.

3 Scenario

The Mars Curiosity Rover has been instructed to autonomously vector across a flat region for two meters. It has to either deal with the problems it encounters itself or ask for help from Earth.

4 The task to solve is as follows

- The Mars Rover, being so distant, must be able to react autonomously to problems with its locomotion, such as obstacles or changes in the surface it is traversing (rocks, loose sand, holes etc).
- Using a subsumption architecture implemented with Condition Variables, implement a wheel control system for six wheels that monitors the state of each wheel (blocked, freewheeling, sinking), and activates a sequence of events to counter that problem.
- For Example:
  - Stopping and choosing a new direction.
  - Lift a wheel.
  - Lower a wheel.

- You system must be able to differentiate between two levels of difficulty:-
  - Rover can solve this problem itself.
  - Rover needs to send a status update describing the problem and await instructions.

4.1 Sample Problems:

- A wheel encounters a rock to large to bump over.
- Loose sand causes a wheel to spin, making it loose track of how far it has travelled.
- A wheel drops, indicating the presence of a hole or possibly a cliff edge.
- More than one wheel encounters a problem.
- The first attempt at an autonomous solution fails.
- Pick a solution, implement it, test if it worked, and if it did carry on. Testing will be via random number generation, so this is not required to be a complex aspect of your solution. Since there is this randome element, your program can and will fail to find succesful paths on it's own.

5 Required elements for the program

- The program must run on the modules Linux server. You must submit a program that can be easily compiled and run without alteration.
- Submitting a non compiled source code file and expecting us to do any work figuring out how to compile it will mean your marks will go down. If we cannot compile it your marks will suffer, so include a compilation solution. If you don't understand this, talk to a member of the module teaching team.
- If your program is not fully automatic, you need to include some form of easy to use, preferably menu based, problem selection method.
  - A threaded menu system where problems can be selected for the Rover to encounter is good.
  - Automatic problem selection would be better, if it is properly implemented as a random selection.
    - If the automatic problem selection is also threaded that would be even better.
- Sample problems for your rover simulation to encounter:
  - Single rock.

  - Wheel/s sinking/freewheeling in loose sand.
  - Cliff edge.
  - Multiple wheels encounterig the same problem.

- Condition Variables **must** be used for passing control between threads within the subsumption architecture. Failure to use them will impact your marks significantly.

- The Rover **must** maintain a log of problems and the solutions it chose. This log to be recorded in a text file.

6 How aspects can be implemented

- You must use a Subsumption model for the wheel problem solving. This subsumption model **must** be threaded, and those threads must pass control between themselves via Condition Variables.

- Within the subsumption model portion of your code, no thread is allowed to do more than one task (for example raise a wheel).

- Outside of that portion this restriction does not apply.

- If you have a thread that does more than one job, you need to make it clear where in the threading heirarchy it belongs if you don't want to risk losing marks.

  - Do this by writing comments in your code.

- Solutions as they are attempted, and the outcome of the attempt must be recorded in the log.

- A request for help from controllers on Earth can be considered a succesful outcome, and recorded as such in the log.

- Success for the two meter vectoring task will be determined by vectoring the two meters or by not becoming stuck.

- During vectoring *at least* **five** problems must be encountered and dealt with so distance measuring *can* be determined by five problems occuring.

- Of these five problems I will expect it to resolve at least two itself, although not on every run. I will want to see this happening, therefore submitting a non functional program will impact your grade.

7 This assignment is marked out of 100 and represents 30% of the module mark

- for each percentage shown in the following breakdown, that is the maximum you can get, not a flat value that will be assigned.

- These **100** grades are distributed as follows

7.1 20 Percent for in class progress demonstration to a mamber of the teaching team

- **20%** is available, in **5%** increments, for you demonstrating progress on your assignment in the classroom.

- For each **5%** you must, each in a seperate week (this is non negotiable), demonstrate incremental development of the assignment.

- Failing to do this will disqualify you from this portion of the grade.

- Asking to do all your observations in one go in the final week will not count, you will at most get one of these one 5% grade increments.

- In short then, they must see progress in your work on four separate occasions over the module to have access to the full grades available.

- Note that we don't expect for this portion of the mark that you get things right, just that you are seen to be trying and making progress. You can use these sessions to get help, and get help at any other time you ask for it.

- There is a potential cut off point for this aspect of the grade. Once we start doing Viva's, there will be less time for graded feedback sessions, so don't leave them too late.

- By this I mean if people are waiting to do their Viva's and you are still trying to get a feedback grade, it's probably not going to happen, so be aware of this and manage your time appropriately.

7.2 30 Percent for the accompanying Report

The **30%** is subdivided as follows

- Up to **15%** if all concurrent Programming elements of your program are listed, with their reason for inclusion, and a brief description of the role they play in your code.

- Up to **15%** for readability (layout, clarity, clear demonstration that you have understood the task).

7.3 Viva of the finished assignment

- to make available a further **40%** of the grade you must, on completion of your software, prior to submision, demonstrate your code to an assessing staff member in class and be prepared to answer questions about the codebase.
- The purpose of the Viva is to demonstrate your level of understanding, both of the task at hand, and the tools with which you are attempting to solve that task.
- Faliure to complete this Viva will mean **40%** of the assignment grade will **not** be available to you when assignment marking is done.
- You will not be given a grade at the end of the Viva, although you will see your feedback, which will be entered on Moodle. You may not have immediate access to the place on moodle where it is written, so ask to see it when it is entered, or to have it emailed to you.
- Questions will be on the following aspects:
  - Program structure/design explanation in detail
  - Justification of program flow selection choices.
  - Your understanding of the concurrency componants. (this may include a whiteboard demonstration by you)
- The program will be run enough times to demonstrate functionality.
- If the program is imcomplete or does not work, some understanding of why this is the case will be expected. If you then go on to submit a non functional program, your answers here will help boost your grade.

7.4 10 Percent for the code portion

This **10%** is subdivided as follows

- Up to **5%** How well does the program answer the assignment task question?
- Up to **5%** How well written the code is (design, commenting, clarity)

7.5 Engagement

- This marking scheme is designed to reward engagement as well as the finished product.
- Higher grades are available to those who work on their assignment once it is handed out than for those who leave it till the last minute.
- Feedback is also more easily available if you are working on your assignment in the classroom.

8 Expectations

- We don't expect everyone to complete this assignment fully. We expect you to make a good attempt.
- We expect you to work on this assignment throughout the module, not cram it at the end. The marking scheme activelly rewards spreading out the work.
- There is room in the marking scheme to get marks for doing something interesting.

9 Grading quick guide

Four Graded Feedback Sessions	5% each
Viva	Up to 40%
Report	Up to 30%
Source Code	Up to 10%