48450 Real Time Operating Systems

Assignment 2 (20 marks) V4

Deadline for submission: 6.00 PM, 1 May 2017

1. Introduction

This assignment will involve some application program developments with real time file reading and writing. You are required to create a program that applies several key concepts of Real Time Operating Systems subject. A submission will be marked based on its merits and may be awarded a mark that is less than the total Mark 20 if it's of modest quality. You are required to include a reflective self-assessment in the conclusion and submit it by the due date.

All programs are implemented in C language.

This assignment is marked out of 20 and comprises 20% of the total score for this course.

2. Assignment details

Topic: Mutex/Semaphore and Pipe for real time file reading/writing -- Mark 20

The program implementation will involve using the concept of threads and pipes. Your program is required to create three threads (A, B and C) for reading data from one file and writing the data to another file through the pipe-line concept. This is similar to an internet file transmission.

Your program is required to define a pipe.

1. Thread A writes one line of data from a given "data.txt" file to the pipe. Note: The "data.txt" text file contains several lines
2. Thread B reads data from the pipe. There are several ways to keep the data for the operation of Thread C. You are required to try your best to find a solution
3. Thread C reads the line from B and detects whether the line is from the file header (see Figure 1). If the line is not the file header, it writes the data as a line into "src.txt" file, otherwise the line will be discarded.

The three threads run sequentially (only one thread will not be blocked) and iteratively (one line for each time). When Thread A reaches to the end of the source file, Threads A, B and C will finish.

Your program is required to include mutex/semaphore, thread and pipe whereby necessary. Please note: The threads A, B and C must run in mutual exclusively based on the three semaphore statuses -- 'write', 'read' and 'justify'. This is to avoid the contents under/overflow within the pipe.

Your program is also need to use a 'struct' to pass the parameters to the threads. The source file can be downloaded from UTSOnline. This text file contains two parts which are shown in Figure 1. We only need to write the data of content region to "src.txt" and the file header region does not need to be written to "src.txt".

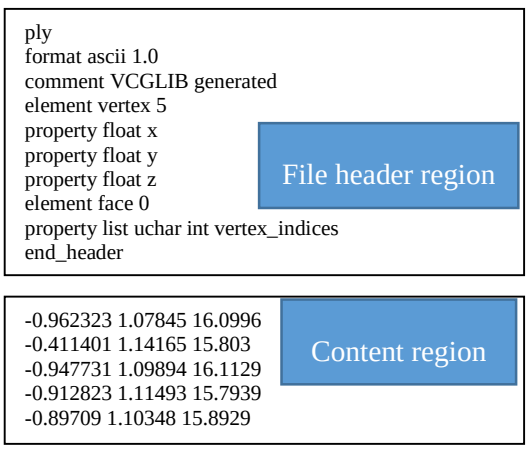

Figure 1. The overall structure the given text file

Experiment: If you do not apply mutex/semaphore, what the result will be by only using the three threads and the pipe? **You are required to write a brief report to summarise your observation**.

3. Assignment Deadline and Submission

The deadline to submit this assignment is 6.00 PM, 1 May 2016

You are required to submit two formats of the assignment:

1. Upload your full assignment report.
2. Upload your ‘C’ code file

If you use makefile for compiling your program, you are required to upload the makefile files as well