Costa Rica Institute of Technology

Computer Engineering

Operating Systems Principles

Prof.: Diego Vargas-Alejandra Bolaños

09/12/2017

Project #1:

Animated ThreadBridge

1. Motivation

With the development of this project the students will be able to integrate in a single assignment the knowledge of concurrency, scheduling, real time scheduling. The students will learn to develop continuous learning techniques to integrate a small bridge circuit simulation to the work.

2. Objective

As part of this project the students will develop a bridge simulation using a circuit, they will apply concurrency concepts with the creation and implementation of a customized pthreads library, scheduling concepts with the implementation of different scheduling algorithms and RT scheduling concepts with the incorporation of RT scheduling algorithms in the project development process.

3. General information

- Value of this assignment: 25%
- Program name: **threadbridge**.
- This project can be implemented in groups of 3 people. However it will be evaluated individually, during the review session random questions will be made to any of the students.
- Any fraud will be scored with 0 and will be processed according to TEC regulations.

4. Functional requirements

A. MyPthreads

You will have to re-implement the Pthreads library with the name **mypthreads.** It will include the following functions from the Pthreads library:

- mythread_create
- mythread_end
- mythread_yield
- mythread_join

- mythread_detach
- mymutex_init
- mymutex_destroy
- mymutex_lock
- mymutex_unlock
- mymutex_trylock

It will also include the following method

- mythread_setsched: This method will take care of changing the scheduling type of the thread.

B. Schedulers

The schedulers should be determined at the moment of the threads creation which means it will be required to include a parameter to the mythread_create in order to establish what scheduler is going to be used by a specific thread. Also, this method should be compatible with a Pthreads standard definition, which means it has to be possible to compile any code that uses Pthreads.

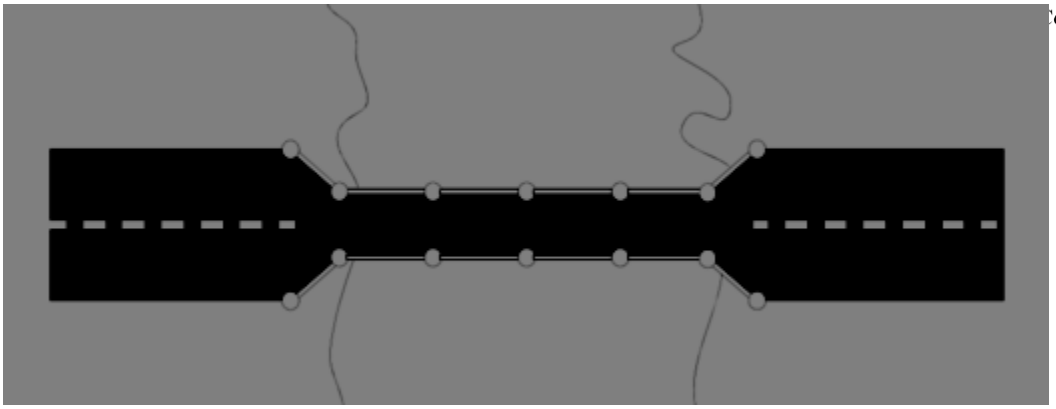
The default scheduler (in case it's not specified) will be Round Robin. You can find the schedulers that will be supported by this custom library as follows:

- Round Robin Scheduler: You should develop the implementation of the scheduler following the Round Robin algorithm.
- Priority Scheduler: You should develop the implementation of the scheduler following the priority scheduling algorithm. The threads created using this scheduler may need extra parameters, for example: the initial priority.
- Shortest Job First: You should develop the implementation of the scheduler following the SJF scheduling algorithm. The threads created using this scheduler may need extra parameters, for example: the amount of time it needs to cross the bridge(Speed of the car).
- FCFS Scheduler: You should develop the implementation of the scheduler following the FCFS scheduling algorithm.
- Real time scheduler: You should develop the implementation of the scheduler following a Real Time algorithm. The threads created using this scheduler may need extra parameters, like for example: time limit.

C. The ThreadBridge

We will have four different bridges located in one of our cities, they had to be temporary installed due to issues with floods that destroyed the original bridges.

These bridges are one way bridges, which means the cars can transit over any of these bridges one way at a time. Because of this reason, we will need a method to take care of the traffic control of the bridges.



Each of those bridges can work with any of the following three methods to control the traffic.

- Access to the bridge is **controlled by a transit officer**. The transit officer is going to determine when a car from any of the sides of the bridge can access the bridge allowing **K** cars from one side of the bridge and then **K** cars from the other side of the bridge (repeating this behavior). If at some moment the queue of cars in one of the sides of a bridge is empty but the other one has cars waiting to access the bridge, then the access to the side of the bridge with queued cars will be granted even when the amount of **K** cars has not been reached.
- Access to the bridge is **controlled by a semaphore** (red and green lights). We will have a semaphore in each side of the bridge, when it's green in one side, it will be red in the other one. The semaphore will be green during a certain amount of time in seconds which will be customized by the user at the beginning of the program execution.
- Access to the bridge is **not controlled at all (jungle law)**. The drivers (well educated and respectful of the law) can decide when to access the bridge following these conditions:
    - A car can only access the bridge if the bridge it's empty or if there are no cars circulating in the opposite way (to avoid collisions).
    - A car can only access the bridge if the amount of cars allowed to circulate over the bridge is not exceeded (according to the bridge length).

In all the cases the cars from one side cannot access the bridge while there are still cars from the other way transiting over it and also if the limits of cars over the bridge is exceeded. Also, It has been determined that in cases where there are ambulances or radioactive cars, they should be allowed to pass the bridge as soon as possible.

You need to generate a configuration file where this parameter (traffic control method) can be specified and manipulated at the beginning of the program execution by the user.

Other custom parameters to be included in the configuration file are as follows:

- The length of the bridge
- Exponential distribution median to be used by the system for the cars generation.
- Car's average speed

- The traffic control method
- The semaphore time in seconds (how much time the semaphore will be green) if using the semaphore method in each of the sides (this time can be different in each of the sides of a bridge)
- The amount of **K** cars in case the traffic control method is set to transit officer
- What percentage of cars will be ambulances
- What percentage of cards will be radioactive cars.
- All these parameters mentioned above are independent of the bridge side (they can be different in each of the sides of the bridge) and independent of the bridge (remember we will have 4 different bridges)

The bridge management main objectives are going to be: maximum bridge usage, minimum number of cars waiting and a fair assignment of the bridge.

D. The Cars

We will have 3 different types of cars, the cars will be generated automatically by the system following an exponential distribution (the creation time of each of the threads representing the cars will follow an exponential distribution).

They will be created in a random way (in each of the bridges) following a percentage that will be set by the user to determine what amount of cars will be ambulances and what amount of cars will be radioactive cars.

The cars (of all kinds) will be created with slightly different speeds. The speed is going to impact the time they will need to cross the bridge.

- **Normal cars:** This type of cars will not have any special treatment to get its access to the bridge.
- **Ambulances:** this type of cars will use a real time algorithm (soft) to schedule its access to the bridge with some kind of priority.
- **Radioactive cars:** this type of cars will use a real time algorithm (hard) to schedule its access to the bridge. If this type of cars don't arrive to their destination quick, it will be a catastrophic.

E. The Bridge simulation

You should write a simulation of the described problem. Each entity (cars, semaphores, ambulances, transit officers, etc) will be represented by a thread that will be taking care of updating the corresponding data structures used in the solution of this project. The user should be able to easily customize which traffic control method will be used to manage each of the bridges.

Messages should be displayed at console level reporting each of the events that are happening in each of the 4 bridges. These messages should be very detailed in order to understand what's exactly happening in each of the bridges.

Optionally a graphical simulation of the system can be created.

F. The Physical Bridge Simulation

You will create a circuit, using any desired embedded device of your preference that is going to interact with the threadbridge program to produce a physical simulation of the different bridges.

The simulation should include the following:

- You should create a physical representation of the bridges
- When a car is passing through a bridge, you should create a representation that reflects the car is passing (remember we may have cars with different speeds, which should be visually represented as well).
- If the bridge traffic is controlled using a semaphore, you need to represent those semaphores using red and green LEDs at each of the sides of the bridge.
- Differentiate normal cars from ambulances and from radioactive cars.

5. Technical requirements

- This project has to be implemented using the C programming language.
- This program will have to be implemented on GNU/Linux (gcc)
  - Make sure a make file is created.
- The circuit piece for the physical simulation can be implemented in any desired language and using any desired embedded device.

6. Documentation

Follow the instructions below in order to document this project. **You don't have to print the documentation**, it will have to be delivered in digital format (PDF).

In any of the sections, you could re-use some of the information in this document if required.

- Introduction
  - Do an overview of the threads and scheduling
  - How do they work?
  - How are they implemented?
  - Provide a brief specification of the project, what it does, how it works. You are free to re-use some of the information provided in this document.
- Development environment
  - Provide all the details of all the tools that have been used during the development of this project.
- Continuous learning attribute analysis

- Document the continuous learning soft attribute required for the career certification program, create a section explaining how this project helped with the continues learning process, contemplating the following:
- You should demonstrate you are capable of integrating information or knowledge to reach the learning objectives
- You should propose solutions to problems being innovative, creative.

- Provide the details of your program design
  - Provide a description of your code.
  - Provide the details of the design of your project using UML.
  - Provide the design details of your circuit created to perform the physical simulation.
- Instructions of how to use the program.
  - Provide detailed instructions of to setup and use the program and the circuit.
- Student activity log:
  - Include here the details of the activities performed by each of the students in this assignment.
  - Include one line per each activity, providing details like:
  - Description of the activity.
  - Amount of time in hours/minutes spent on each activity.
  - Total amount of hours spent in the development of this project per student.
  - **Use a time-sheet format** (or table) to present this activity log.
- Project final status
  - Provide a detailed status of your project.
  - Provide details of any issues, limitations or challenges you may have faced during the development of the project.
  - Document any known issues or bugs (if any).
- Conclusions
- Suggestions, recommendations.
- References
  - Provide all the references used during the development of this project. Use APA format and make sure all the references documented in that section are actually used in your documentation. Failure to create the references properly will be scored with negative points in the documentation review.
- Make sure your **source code** is well **documented**.

- Explain what your functions/procedures do, all their parameters and document your expressions and data structures.

- Digital documentation:
  - Include your documentation inside the .tar.gz file that will be created with all the source code and executables of this project.
  - Upload the .tar.gz file to your shared directory in Google Drive along with the corresponding timestamp, sha1 hash and your digital signature of the document.
  - Follow the instructions in the next section with the details of the structures of the directories you must use.

7. Deliverables

- Source code and executables of your programs. The programs should be in compliance with the specifications in the **Technical requirements** section.
- Documentation (sources and pdf).
- Use the following structure inside your “Proyectos” directory in Google Drive:
  - Proyectos:
  - <carne>-proyecto2.tar.gz: compressed file with the following contents:
    - Documentation: this should be a directory with both the PDF and source files (.tex or .md) for the documentation.
    - Program: this should be a directory with your program source code and executable.
    - Additional_files: this is an optional directory, it should have any additional file you think it would be required.
  - <carne>-proyecto2.tar.gz.asc
  - <carne>-proyecto2.tar.gz.tsr
  - <carne>-proyecto2.tar.gz.sha1

8. Evaluation

- ThreadBridge: 60%
  - Bridges and cars implementation: 10%
  - MyPthreads: 20%
  - Schedulers: 20%
  - Threadbridge simulation (console simulation): 10%
  - Keep in mind this simulation is mandatory, in here all the events will be documented at console level which is important during the review process.

- Physical simulation: 20%
- Documentation using Latex or Markdown: 20%
    - Internal documentation: 3%
    - Continuous learning attribute documentation: 5%
    - The rest of the score will be determined by the professor during the review process.

9. Additional considerations

The programs and documentation are in separated items in the evaluation but the following restrictions apply:

- If documentation is not delivered, you will have automatically a score of 0 in this project.
- If the source code is not compiling, you will get a score of 0. Make sure you provide a functional source code.
- The program has to be programmed in gcc for GNU/Linux. Failure to do this will give you a score of 0 in this assignment.

Also the take the following under consideration:

- The professor will be reviewing the documentation out of the revision session.
- During the review session, the program deliverables could be downloaded from any of the students shared directories chosen by the professor.
- Each group will have up to 20 minutes to present the project during the review session and perform the technical defense of the work. The responsibility of presenting and defending all the work relies on each of the students, so it is recommended to have everything ready and handy prior the review session.
- Every error or warning displayed during the review session considered to be part of your technical validations during the development of the programs, will be punished on your final project score with -2 points.
- Each group will be responsible of the equipment that will be used during the review session. In case of issues to take your own equipment, you will have to notify the professor with 2 days in advance (prior the review session) so the required equipment can be properly coordinated.
- The following people could participate during the review session:
    - ↘ Other professors.
    - ↘ Coordinator of Computer Engineering.

10. Delivery Date

October 5 of 2017 before 23:59:59 GMT-6.

11. Revision Date

The review date of this project will be October 7th at 10:00. During the review you will have to download the work from your shared directory and execute your work in the H laboratoy.