Assignment P2-1

Jakob Roberts - CSC 360

1. How many threads are you going to use? Specify the work that you intend each thread to perform

I am going to use one thread for every train read from the file on top of the main thread. Each thread created per train will count the loading time for the train, put it into a queue then wait until it can be put into the main track. The number of threads used will vary depending on the input file.

2. Do the threads work independently? Or, is there an overall "controller" thread?

There is a controller thread which the main thread performs this task. The main thread keeps track of all the queues and makes sure that the signal goes to the correct head of a queue when the main track is empty.

3. How many mutexes are you going to use? Specify the operation that each mutex will guard.

I will use five mutex. One mutex will guard the placement and running of a train on the main track at one time. The second will ensure that only one train is being put into the Low priority station Westbound queue at a time. The third will ensure that only one train is being put into the Low priority station Eastbound queue at a time. The fourth will ensure that only one train is being put into the High priority station Westbound queue at a time. The fifth will ensure that only one train is being put into the High priority station Eastbound queue at a time.

Essentially, one per queue(of which there are four) and one for the main track.

4. Will the main thread be idle? If not, what will it be doing?

It will be somewhat idle and waiting to receive another file with more trains and values if necessary. It will also be checking if all the queues are empty, and the main track is empty, then the program has nothing to do.

5. How are you going to represent stations (which are collections of loaded trains ready to depart)? That is, what type of data structure will you use?

Four queues, two for high priority and two for low priority, each having one for westbound and eastbound trains. Once the high priority queues is empty, the low priority queues can be pulled from. A train is only added to a queue once it has been loaded.

6. How are you going to ensure that data structures in your program will not be modified concurrently?

By the use of each station mutex (low and high priority/eastbound and westbound), a loaded train can only be added to one type of station at a time.

7. How many convars (condition variables) are you going to use? For each convar:

One convar

(a) Describe the condition that the convar will represent.

1- signal for when train is on/off of main track

(b) Which mutex is associated with the convar? Why?

1- the fifth mutex that ensures that only one train is on the main track at one time

(c) What operation should be performed once pthread cond wait() has been unblocked and re-acquired the mutex?

1- it dequeues the head of the queue that has been selected by the controller thread and puts the thread onto the main track.

8. **In 15 lines or less, briefly sketch the overall algorithm you will use. You may use sentences such as: If train is loaded, get station mutex, put into queue, release station mutex. The marker will not read beyond 15 lines.**

1. Get input file, parse, create threads for trains and begin “loading” trains.
2. If train loaded, get station mutex, put into queue, release station mutex. <repeat for all input trains>
3. Choose train from a high priority queue, if queues empty, then choose train from low priority queues.
4. If chosen low priority train and high priority train becomes available, switch choice.
5. Ensure that next choice(if two identical priority trains available) is opposite of last main track train.
6. Get main track mutex, dequeue chosen train, run train on main track, release main track mutex.
7. If all queues are empty and no train on main track, all mutex are free and alive threads are idle.