

1)

I am going to use N threads, N is the number of input trains. One thread for each train. It will sleep till it has finished loading, then it is ready to “cross” on the main track. Then it will sleep till it has finished crossing the main track.

2)

No, the threads will work concurrently. No, there will not be an overall controller thread in my design.

3)

I am going to use 2 mutex.

One mutex will guard the “main track” from input trains. The mutex locks the main track when it is being used and unlocks it when it is available to use.

Another one which guards the shared queue where the trains go into when they have finished loading. So, if two trains finished loading at the same time only one of them can access the queue.

4)

No, the main thread will not be idle. It will schedule and dispatch the input trains.

5)

I am going to use 4 queues to represent the stations.

6) By using mutex to guard the shared resource between threads.

7)

I am going to use 2 condition variables,

- a) One represents the condition of the main track, whether it is being used or not. Another one represents when a train has finished loading.
- b) One mutex which I am going to use which guards the main track, so this mutex is associated with the main track condition variable. The condition variable tells the mutex whether to lock or unlock the main track. Another mutex which guards the queues, associates with the finished loading condition variable, unlocks for loaded train or lock when it is being used.

c)

After pthread cond wait() unblocked and re-acquired the mutex:

Main track condition variable: main thread dispatches the next loaded train onto the main track.

Station queue condition variable: put loaded train into the queue

8)

Main()

Read input trains

Sort trains by their properties to determine which train to cross first, into an array called order[]

Broadcast to start loading and start timer

Check and wait to look for a loaded train to cross, dispatch next train to cross once been signaled

Wait for signal to remove the crossed trains in the order[] and queue

End if order[] is empty

Thread()

Wait for a signal to start loading

Sleep for Loading time

Get station mutex, put into queue, release station mutex

Wait to see if it is allowed to go

Get main track mutex, access the main track, release main track mutex

Sleep for crossing time

Signal to unlock mutex and finished crossing

Exit