Alain-Joseph Po-Deziel
University of Victoria – CSC 360

p2a (Design Document)
Monday February 19th, 2018

1) I will be using 3 threads, where each thread will represent a train. Each thread will therefore provide the train's status, as well as making it easy to determine if the train is loading or crossing the central track. For example, if the train is being loaded, the thread will be set to sleep for the amount of time specified to load it in the input file.

2) The threads will be working independently, as no thread depends on the function/behaviour of the other threads. In other words, it does not rely on another thread for it to function correctly (e.g. if there was a thread that monitored all of the trains' statuses, it would not need to wait on a dispatcher thread). As a result, there is no overall controller thread.

3) There will be 2 mutexes.

   Mutex 1: TrackControl
   The first will be used for controlling access to the whole single strip of track. If one train enters the track, that thread will be locked and no other train will be allowed to enter.

   Mutex 2: StationControl
   The second mutex will be used for controlling access to train stations. If a train is currently present in the respective station and is loading, that thread will be locked to prevent other trains from entering.

4) If a train thread is not able to enter the track, it will idle. Otherwise, the train thread must either be loading or crossing the track.

5) To represent stations, a priority queue will be used. This will allow loaded trains to be collected into it, and then ordered by priority. However, if two trains have same priority, then the specifications set out in assignment will prevail.

6) To prevent data structures from being modified concurrently, a mutex will be used to prevent train threads from modifying the stations' priority queues while a train is crossing the track.

7) There will be 2 condition variables, as shown below:
   a) **Convar 1**: If train present in central track, lock mutex.
      Associated mutex: TrackControl
      Operation to perform: Lock TrackControl mutex to prevent other trains from accessing central track while it is in use by another train.

   b) **Convar 2**: If train present in station, sleep train thread.
      Associated mutex: StationControl
      Operation to perform: Lock StationControl mutex to prevent other trains from accessing respective station while it is in use by another train that is loading at that moment.

8) **Algorithm design assumes rules laid out in assignment spec.**
   - Create all necessary train threads.
   - Verify if there is a train in priority queue of station. If not, begin loading (i.e. wait on respective thread). Otherwise, sleep thread.
   - Once loaded, put train into priority queue.
   - If no train on track, allow train with highest priority to go forward.
   - If there is a train on the track, wait train with highest priority.
   - Once allow to cross, pop train out of priority queue.
   - Once train crosses, close respective thread and increase train_count.
   - Once train_count reaches number of trains read from file, exit program. Otherwise, rerun main method.