Flight Scheduling

(Pranjal Singh, Kartik Singh)

Aim

The project aims to build an automated flight scheduler which takes the input as multiple flights (either leaving or entering the runway). The outcome will be to define the order in which the runway is used by the flights.

Implementation Details

In any airstrip, there is only one available resource i.e the runway. The number of users of the resource are multiple. Hence, we must ensure that while the runway is used by one of the flights, no other flight interferes with it. This is done in order to avoid any crashes. So, we must sort the flights according to the priorities first (taken as input by the user). Next, the flight that requires the runway is given the access to it and simultaneously, it is ensured that no other flight uses the resource until the resource is freed up the flight that used it. In order to implement this, we can try out multiple things. First, we can run the flights on different threads and then based on their priority, we can choose the one that should run first. This is a fairly simple version of the implementation. Next, we can do it using semaphores. We can make the runway as a shared resource. Whichever user (in this case flights) wants to access it can acquire the resource and then set it's availability to zero. By this, we can ensure that no two users are using the resource simultaneously. Another way of implementing this is using mutex locks. We can assign a mutex lock to the runway. Whichever resource wants to access the resource can acquire the lock and upon the completion of the usage, release the lock so that it can further be used by other users. The final way of doing it is using monitor but for the sake of efficiency, we are going to implement it using Threads and Semaphores.

Algorithm

The runway is treated as a resource and the flights as the users. Based on the requirement to use the resource, the users are granted the access. The permission to use the resource depends upon a number of factors.

- Flight wants to take-off and access the runway.

  1. If the runways is clear, grant access.
  2. If the runway is in use, make the flight wait.

- Flight wants to land and access the runway.

  1. If the runway is clear, grant access.
  2. If the runway is occupied then irrespective of the priority of the next flight, immediately grant access to the current flight as soon as the runway is available.

- Multiple flights want to use the runway.

  1. The flights want to take-off
     (a) Grant access to the flight with higher priority
  2. Flights want to land
  3. Grant access to the flight with the higher priority and immediately after its landing, grant the access of the runway to the next flight that wants to land.

  1. One flight wants to land while one wants to take-off
     (a) Irrespective of their priority, grant access to flight that wants to land.
     (b) After its landing has been completed, grant access to the flights again based on the algorithm.

Submission -

1. At the time of first submission, we'll just show the implementation of flights taking-off

2. During the final submission, we'll show the completed project that'll include both the flights taking off and landing.

Coding Language - Java