INDIVIDUAL ASSIGNMENT

TECHNOLOGY PARK MALAYSIA

CT074-3-2

CONCURRENT PROGRAMMING

APU2F2206SE / APU2F2206CS(DA) / APU2F2206CS / APD2F2206CS(DA) / APD2F2206SE / APD2F2206CS

HAND OUT DATE : 1 DECEMBER 2022

HAND IN DATE : 13 JANUARY 2023

WEIGHTAGE : 20%

INSTRUCTIONS TO CANDIDATES:

1. **Submit your assignment at the administrative counter.**
2. **Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing).**
3. **Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.**
4. **Cases of plagiarism will be penalized.**
5. **The assignment should be bound in an appropriate style (comb bound or stapled).**
6. **Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**
7. **You must obtain 50% overall to pass this module.**
8. **This report is Part 1 of 2 for the assignment.**

Name : **Yam Chen Xi**

TP Number : **TP061635**

Intake Code : **APD2F2206CS**

Asia Pacific Airport's Air Traffic Controller Simulation

Due to the easing of Covid-19 regulations, demand for aviation services such as passenger traffic, cargo demand, and airport staff is gradually increasing. Hence, the Asia Pacific Airport is currently managing more flights and has lengthier wait times for landing authorization as there is only one runway that is available for all the airplanes to land and depart. To assess the situation at the airport and improve the effectiveness of the aviation service, the airport management decided to simulate Air Traffic Controller to curb the issue. The simulation is aimed to create a faultless air traffic controller system for Asia Pacific Airport while minimising the occurrence of accidents. Thus, it is necessary to visualise the events that occur in the airport and solve the problem that happened during the simulation.

The assumptions and activity flow are stated as following. There will only be 6 airplanes arriving at Asia Pacific Airport in a day due to limited capacity available at the airport. Once there are the sixth airplane arriving, the airplane generator method will set airport full to true. This will allow the airport to stop receiving more arriving airplanes and announce the airport is at full capacity as the airport is unable to accommodate more incoming airplanes to be land and resupplied.

Airplane generator will randomly create the instances airplane thread every 0 to 2 seconds. By using a clock thread, the generator will be set to sleep(rand.nextInt(3000)) to ensure that there will be new airplane arrives every 0 to 2 seconds. The new airplane is placed into Linked List to queue the airplane in first out method for landing. The method is implemented by adding the new element at the tail of the queue and remove the oldest element from the head of the queue while prioritising the airplanes that required emergency landing. This will ensure the fairness for the airplanes in the list and avoid collision happening at the runway. To minimize the occurrence of concurrency problems, multiple threads may insert and take elements simultaneously from the Linked List.

Once the runway is available, the next airplane in the queue will be permitted to land and dock and the rest of the airplanes in list will be waiting for the runway is free to be used. A clock thread will be used in this to allow the thread to sleep for 1 second in between before the next airplane utilizes the runway. This is to make sure the runway to be cleared before departure and arrival of airplanes aiming to reduce the risks of accidents that may happen on the runway.

Each of the process to be carried out are required to take some time and take turn for the airplanes to utilise the shared resources. Hence, wait() method and notify() method are used accordingly to control the activities of the airplane and actions to be taken on the airplane. Wait() method will cause the selected thread to wait until the other thread that invokes the next process to continue with notify() method. These methods can be used in a scenario where an airplane wait for the airplane that has higher priority in the queue to land. Once the airplane has landed, it will notify the system that the runway is cleared to be used. Therefore, the processes can be executed with a smooth flow as each of the process are responsive to one another. With wait and notify method, the airplanes are able to carry out the tasks in a more organized manner and minimise miscommunication that may arise within the system.

The airplanes will coast to the assigned gate and docked at the designated gate to disembark passengers and allow the staffs to prepare for the next flight. By using the clock thread, the airplane that have acquired the runway in 3 seconds. It will also be using 2 to 3 seconds to dock at the assigned gate and using at least 2 to 5 seconds to unload passengers and load the passengers of the next flight into the airplane. Concurrently, refilling of supplies and cleaning of aircraft will be carried out within the same timeframe. In the meantime, there the airplane will also be refuelled and prepare for departure after all the procedure are done.

There is only one refuelling truck available at Asia Pacific Airport, the refuelling of aircraft will be implemented according to the availability of the refuelling truck in first come first serve priority at the airport. With this condition, deadlock may happen in this thread as the airplanes are required to share the same refuelling truck and wait for the process to be done until its turn. To prevent the occurrence of permanent blocking within the process, synchronisation allows the resources to lock and unlock before any thread access the resource. By doing this, multiple threads will be able to take turn to use mutual resources without causing starvation in the simulation due to greedy threads. This is due to the reason a synchronised method can only be accessed by one thread at a time and block access from other threads from accessing this method. Thus, the first airplane that arrived at the gate will be granted the permission to be refilled first and release the method to the next airplane after the process is completed. In this process, the priority is given to any airplanes that landed in emergency with fuel shortages. After all the required events has been completed at the airport, the air traffic and runway are checked to ensure it is clear to permit airplanes to depart.

If there are special situations such as medical emergency, malfunctions and fuel shortage occurring on an airplane, the airplane will be the first priority to be landed. By utilising setPriority() method and set to MAX_PRIORITY, the airplane priority will be move to the top of the queue to allowing it to use the runway to land as soon as possible. Other airplanes that have lower priority in the list are required to wait in the queue for the plane with emergency to land first. Therefore, the airplane with emergency can be landed safely and able to get the respond that the airplane needed immediately. One of the special situations, airplane that landed due to fuel shortage will be prioritised to be refuelled at the airport. If the refuelling truck is currently refuelling an aircraft, it will complete the task before approach the airplane that has fuel shortage and carry out its duty.

The airplanes that are ready for departure will be the second priority in the queue to use the runway when the airport gates are fully occupied. This is because there are only 2 airport gates available at Asia Pacific Airport, the airplanes that are ready for departure will be prioritised in the queue to depart when there is no more place to allow more planes to lane. This is to allow the airport to vacate more gates for airplanes to land and ensuring that the airport continuously to permit arriving airplanes to be landed. If there are airplanes that required emergency landing, airplanes that are ready for departure will be the second priority to use the runway.

By planning out the assumptions and activity flow of Asia Pacific Airport's Air Traffic Controller Simulation, it provides a clear guideline on what should be done after each of the procedure. Additionally, the adaption of concurrency concepts and safety aspects of the system will allow the simulation to run multiple threads simultaneously and recreate the process accurately with minimum error.

References

Alejandro Ugarte. (2019). *What is Thread-Safety and How to Achieve it*. Baeldung. https://www.baeldung.com/java-thread-safety

Barua, S. (2022). *Synchronization in Java*. GeeksforGeeks. https://www.geeksforgeeks.org/synchronization-in-java/

Oracle. (2020). *BlockingQueue (Java Platform SE 7)*. Java™ Platform, Standard Edition 7 API Specification. https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/BlockingQueue.html

Pankai. (2022). *Thread Safety in Java | DigitalOcean*. DigitalOcean. https://www.digitalocean.com/community/tutorials/thread-safety-in-java

Scientech Easy. (2020). *Thread Priority in Java with Example*. Scientech Easy. https://www.scientecheasy.com/2020/08/thread-priority-in-java.html/

Wilson, C. (2019). *Multithreading and Concurrency Fundamentals*. Educative. https://www.educative.io/blog/multithreading-and-concurrency-fundamentals

Yadav, I. (2022). *Deadlock in Operating System*. Scaler. https://www.scaler.com/topics/operating-system/deadlock-in-os/