Damla Övek – 31587
Ahmet Can Turgut – 40616

COMP 304 : OPERATNG SYSTEMS
PROJECT 2 : BLACK SEA TOUR

In this project, a simulation of a travel agency, which provides bus tours to Black Sea for its passengers, is implemented by using C/C++. In the simulation, there are passenger and agent threads that are capable of buying, reserving, canceling and viewing buss tickets.

Pthreads are used for this project and mutexes and conditional variables are utilized in order to mutually exclusively perform buy, reserve and cancel operations. View operation is performed simultaneously by multiple threads.

Passenger are capable of cancel and/or view their bought or reserved tickets whereas agents view all tickets and buy, reserve, cancel tickets for their passengers.

When project source code is executed, it takes input from user and generates simulation based on inputs. There is also help method to demonstrate commands to user. Figure below shows the output of help command.

```
*********************************************************
*********************************************************
*   COMP 304 - OS Project 2 - BLACK SEA TOUR            *
*   Koç University - 2017 Spring                        *
*   Type -h to see capabilities                         *
*   Created by Ahmet Can Turgut & Damla Övek            *
*   CopyRight.. University Code of Honesty              *
*********************************************************
*********************************************************
./black_sea_tour: option requires an argument -- 'h'
usage:
  project [options]
options:
  -d                The total simulation time in days
  -p                Number of passengers
  -a                Number of agents
  -t                Number of tours
  -s                Number of seats
  -r                Random number generator seed
  -h                Show this help message
```

Simulation lasts D days given by user as input or one day as default. Real-time is used in the simulation. Main thread gets start time and simulation runs until current time equals start time plus simulation time. Each day lasts 5 seconds in the simulation. At the end of each day, daily reports of buying, reserving events and wait lists of passengers are shown to user.

Passengers are not allowed to reserve more than 2 seats in a day. Therefore, number of reservations belonging to the passenger is checked before a reservation request. Also, reservations that are not bought within a day are canceled.

PART I:

In the first part of the project, passenger threads were created. If user gives number of passengers as an input with -p command, main thread creates that many passengers. Otherwise, 5 passengers are created by default. Each passenger thread runs until simulation stops. During their run, they choose random seats on random tours and choose one of events (buy, reserve, cancel, and view) randomly. Since they perform their operations by themselves, their agentID is set to 0.

PART II:

In second part, agency threads were created. If user gives number of agents as an input with -a command, main thread creates that many agents. Otherwise, 5 agents are created by default. Each agent thread runs until simulation stops. During their run, they choose random passengers, random seats on random tours and choose one of events (buy, reserve, cancel, and view) randomly.

PART III:

In third part, a new feature is added to passenger threads. Before passengers try to buy or reserve a ticket, they first check if there is any available seat on the tour. If tour is full, they sleep until a notification from tour comes to say that tour becomes available. Sleeping passengers are kept in wait list. These sleep and notify operations are handled by using condition variables. There is a condition variable for each tour in the simulation. When a passenger send a buy or reserve request to a tour, it first wait for availability signal from the tour. If simulation ends, all sleeping threads are terminated before exit.

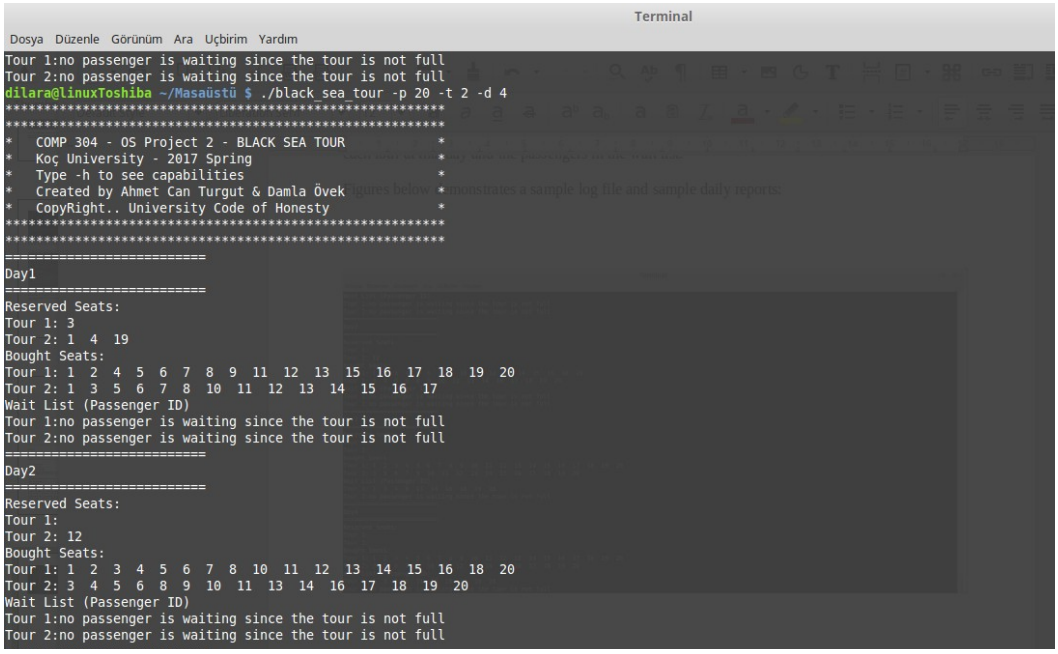
PART IV:

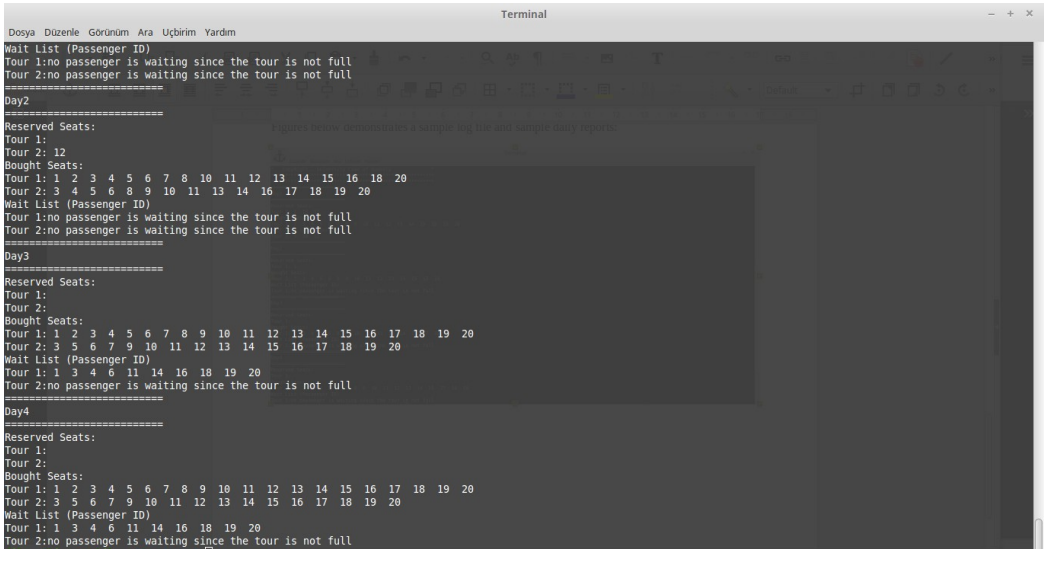
In this part, simulation is modified in a way that it handles multiple tours departing one bus associated with each tour.

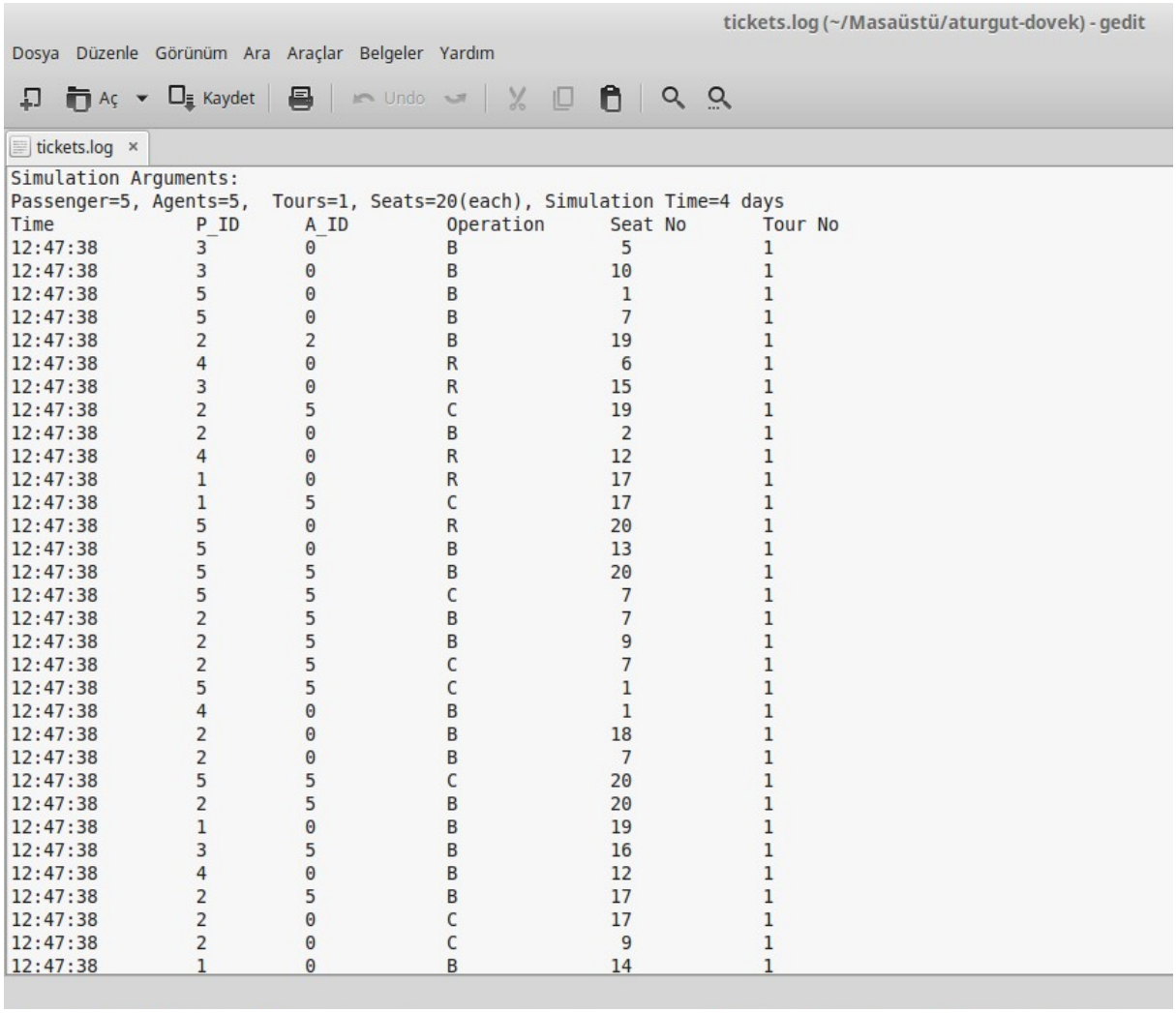
PART V:

Finally, “tickets.log” file is generated in order to keep record of events. Real-time of events are kept with the passenger id, agent id, tour number, seat number, and operation type. Also, daily reports are demonstrated to user. For daily reports, current time is periodically checked. When a day ends, a method named as dailyReport(int day) is called. It prints the bought and reserved seats on each tour at that day and the passengers in the wait list.

Figures below demonstrates a sample log file and sample daily reports:






tickets.log (~/Masaüstü/aturgut-dovek) - gedit
Dosya Düzenle Görünüm Ara Araçlar Belgeler Yardım
Aç
Kaydet
Undo
tickets.log
Simulation Arguments:
Passenger=5, Agents=5, Tours=1, Seats=20(each), Simulation Time=4 days
Time P_ID A_ID Operation Seat No Tour No
12:47:38 3 0 B 5 1
12:47:38 3 0 B 10 1
12:47:38 5 0 B 1 1
12:47:38 5 0 B 7 1
12:47:38 2 2 B 19 1
12:47:38 4 0 R 6 1
12:47:38 3 0 R 15 1
12:47:38 2 5 C 19 1
12:47:38 2 0 B 2 1
12:47:38 4 0 R 12 1
12:47:38 1 0 R 17 1
12:47:38 1 5 C 17 1
12:47:38 5 0 R 20 1
12:47:38 5 0 B 13 1
12:47:38 5 5 B 20 1
12:47:38 5 5 C 7 1
12:47:38 2 5 B 7 1
12:47:38 2 5 B 9 1
12:47:38 2 5 C 7 1
12:47:38 5 5 C 1 1
12:47:38 4 0 B 1 1
12:47:38 2 0 B 18 1
12:47:38 2 0 B 7 1
12:47:38 5 5 C 20 1
12:47:38 2 5 B 20 1
12:47:38 1 0 B 19 1
12:47:38 3 5 B 16 1
12:47:38 4 0 B 12 1
12:47:38 2 5 B 17 1
12:47:38 2 0 C 17 1
12:47:38 2 0 C 9 1
12:47:38 1 0 B 14 1