Assignment # 5

Student Name: Varadharaja Perumal

Student Id : 1008475675

Due: Tuesday, March 15, 2016

Milestone Document

1. Requirement/Specifications

(a) Problem statement – The problem involves the scenario of flights arriving into an airport, being allotted gates for the aircraft, and scheduling departures. Due to the limited number of gates available in an airport, the availability of gates for every flight arrival is at premium and hence the problem involves simulating situations of arrivals and departures with different number of gates, and arriving at the minimum number of gates that would be required to accommodate all flights so that "Airport full" situation never occurs.

(b) Input to the program – The user enters just one positive integer number that would represent the number of gates in the airport to be considered for the simulation.

(c) Output description – The program output is based on the result of simulation. With the user entered number of gates, if the program is able to allot a gate to every flight arrival throughout the 24 hour period, then the schedule of flights arrival, departure, gate number allotted and other details of the flight is chronologically printed for the entire day, else if the program encounters a situation where there is no gate available to any flight arrival, the program gets aborted with a message "Airport Full".

(d) Assumptions – some key assumptions made for the program to run without any errors are – the input gate number entered by user is a valid positive integer, the hourly schedule of flights arrival into the airport is considered as already available or can be generated from within the program. The departure of flight is schedule one hour after arrival but may be delayed. Once the flight departs, the particular gate is available to be allotted to the next arrival flight.

2. Problem analysis -

(a) The basic program involves simulating an event – which can be an arrival of a flight into the airport or else departure of a flight from the airport. The critical component in this simulation is the time of arrival or departure, and the allotment of gate numbers to every arrival flight.

(b) Since we need to find the event nearest in the future, it is appropriate that the set of arrivals and departure events waiting to happen be organized in a Priority queue (Binary heap), based on the time component of the event. The next event is thus the next arrival or next departure (whichever is sooner), both are easily available. Once the initial heap is built, the

minimum element (nearest event arrival/departure) in the heap is found, processed and deleted. The processing involves checking whether the event is an arrival or departure.

(c) If the event is departure, the processing would involve release of a gate that can be utilized for the next arrival and deleting the event from the list of events waiting to happen.

(d) If the event is arrival, then the process would involve allotting a gate number, and computing the departure time of the flight, and adding the new departure event to the set of events waiting to happen. If the arrival event does not find a gate number allotment, then it means the airport is full to its capacity and that aircraft could not be serviced at that time of arrival.

(e) The simulation would end when there are no more arrivals or departures in the heap to be processed and the heap size reduces to zero. This would mean that the number of gates initially entered was sufficient enough for handling all the arrivals throughout the day.

OR

The simulation may end when a particular arrival event could not be allotted any gate numbers due to non availability of gates at that time of occurrence of the arrival event. This would mean that the number of gates initially entered was not adequate for handling all the arrivals throughout the day.

(f) The basic data structures used in the program implementation is

(i) Binary heap class object – for the arrival and departure events which by very nature of the heap property organizes the data in a sorted order. The binary Heap-order property places the smallest or earliest event at the root of the heap, i.e. arranges the events as they occur in time of the day.

(ii) Stack class object – to have the available gate numbers in a stack object, thereby making it easy to push an available gate number(released on a departure event) into the stack or pop a gate number from the stack to allot to an arrival event.

(iii) An event structure is defined that has fields to hold the arrival/departure times, arrival/departure code, flight number, originating location and destination of each flight.

(g) Algorithm/Pseudo code in brief -

Ask gate numbers from user

Create an event structure/record

Create a stack object with gate numbers

A double for loop to create (24 * 14 = 336) 336 arrival times with flight numbers starting from 1000 and insert all arrival event records into a vector

Create the binary heap object using the arrival event time vector just created
Start a while loop to execute till binary heap size becomes zero

-------Start while loop

Get the minimum from heap, check the event for arrival or departure record, store in a temp variable, delete minimum from heap

If event is departure then store back its gate number into gate stack, and store the event in the schedules vector as a departure event

If arrival flight

Check for a gate number availability

If gate number available

Store the event in the schedules vector as an arrival event

Allot a delay time and re insert into the heap as a departure flight

Else airport full - abort

---- End while loop

If all arrivals have been successfully allotted gate numbers then the new vector thus formed will have all arrival and departure information

Create a new heap using the schedules vector which will sort all flights in proper order of time which can be printed

3. Program implementation performance analysis -

(a) The initial construction of heap-ordered tree object is done with N items (N arrivals) using the build heap function call from within the parameter constructor of the Binary heap class. The order of this build heap is linear i.e. O (N). There after there would be an insert operation for every new departure event.

(b) The binary Heap-order property organizes the data in a sorted order, places the smallest or earliest event at the root of the heap, and i.e. arranges the events as they occur in chronological order of time of the day. Hence finding the minimum just takes O (1) and there by contributes to the efficiency.

4. Results of the program simulation and analysis

(a) Number of gates less than or equal to 20 (<= 20) – the airport will always have an overcrowded situation and the Airport will not be able to allot gates for all arrivals.

(b) Number of gates between 21 and 22 – there is likely chance of running successfully and being able to allot gates for all arrivals.

(c) Number of gates above or equal to 23 (>=23) – the airport will always be able to service all aircraft arrivals throughout the day.

5. Machine details:-

Dell Laptop – year 2009

Processor Intel(R) Core(TM) 2 Duo CPU T6500 @ 2.10 GHz, 2100 MHz, 2 Core(s), 2 Logical Processor(s)

System Manufacturer Dell Inc.

System Model Studio 1737

System Type x64-based PC

Installed Physical Memory (RAM) 4.00 GB