

Programming Project #4

EGRE246 Fall 2021

Airport Landings

1 Overview

For this project you will write a *time-driven* simulation of planes landing at an airport. A time-driven simulation is where events occur with every advancement of an increment of time. Our clock will be simulated by a simple integer variable that is initialized to 0 and is incremented by one every time you wish to advance the clock one “clock tick”.

2 Input

Your input must come from a text file given on the command line. The contents of this file, in this order, are:

- the duration d of the simulation in clock ticks where $d > 0$
- the number of runways r available where $r > 0$
- a list of planes that will try to land, where each plane is composed of 4 items:
 1. scheduled landing time $s > 0$
 2. type of plane (as a single non-empty string)
 3. number of passengers on board $p \geq 0$
 4. time in clock ticks t required to completely land where $t > 0$
- a -1 to indicate the end of the list of planes

Example data file `planes.dat`:

```
10
3
1 Boeing727 112 4
2 AirbusA320 208 5
3 Cessna172 2 2
3 Beechcraft 3 2
3 EmbraerE190 86 3
3 Boeing767 214 5
4 Boeing767 159 5
5 AirbusA320 169 5
-1
```

You may assume that all data for the planes is correct and legal.

3 Output

Your output must be formatted exactly as the sample runs given below.

3.1 Error Output

There are two possible fatal errors that can occur when you program runs. If either of these situations occur **you are to output the error messages given and immediately exit the program.**

no file name on command line

message: [ERROR] no input file on command line.

input file does not exist

message, for an incorrect input file `badfile.dat`: [ERROR] file '`badfile.dat`' not found.

3.2 Statistics

With event reporting turned off (more on this later) you are to output only simulation statistics, i.e., in this order:

- length of simulation (in clock ticks)
- total passengers landed
- total runways
- runway usage (as a percentage)
- number of planes delayed
- total time planes were delayed

3.3 Event Reporting

When the preprocessor identifier `ECHO` is defined you are to report the following at the appropriate time:

- the current time
- plane is waiting to land
- plane lands on runway n
- plane has landed and has moved off of runway n
- at end of clock tick report the busy runways

The identifier `ECHO` is not defined by default; it is expected to be defined on the compilation command line. The statistics are outputted when event reporting is turned on. See the sample runs for the expected form of these messages.

4 Limits

maximum_string_length \leftarrow 80
maximum_number_runways \leftarrow 8
input_sentinel_value \leftarrow -1
maximum_duration: none
maximum_number_planes: none
maximum_number_passengers: none

5 Implementation

You should define data structure for both airplanes and runways; represent you runways as an array of runway structures. You are to input planes *only* when necessary, i.e. do not input all of your planes and store them in an array or list!

Note that it is very possible that a plane is scheduled to land at a certain time and there are no runways free at that time! When this happens the plane remains in the waiting area and we say it “circles” waiting until a runway is free. Planes always land in the order that they are scheduled in the data file, one per clock tick. Note that planes are issued an id number beginning with 0.

A plane is to be considered as landed when it moves off of a runway after it takes its designated amount of time to land. Consider the runways numbered from 1 to n where n is specified in the data file. Planes always land on the lowest numbered runway that is available, if any. Only a single plane can land per clock tick.

You should subdivide your code into appropriate functions as reasonable and needed. Most all of your variables should be defined in `main` and passed into and out of functions as needed.

6 Algorithm

Here is a suggested general algorithm that can be used as a starting point.

```
open data file
initialize all variables
input simulation duration & number of runways
set clock to initial value
set plane waiting area to empty
while there's still time left (loops once per clock tick) do
    if there's input & the plane waiting area is empty then
        load next plane into waiting area
    endif
    if waiting area is not empty then
        if plane waiting's arrival time <= clock then
            land a plane onto a runway
            set waiting area to empty
```

```

        endif
    endif
    for all the runways that are busy do
        decrement time left for landing by one
        if time left for landing = 0 then
            move the plane off of this runway
        endif
    endfor
    increment clock by 1
endwhile
output stats

```

7 Hints and Suggestions

- Be sure to get your input working 100% correctly before moving on to other parts of your code.
- The ‘waiting area’ is just a variable (of plane **struct** type) that you ‘move’ planes into and out of.
- Your runways are an array of runway type (probably a **struct** type). You land a plane onto a runway by assigning it from the waiting area to appropriate cell in the array of runways.
- Remember that you delimit code that is to be conditionally compiled (included) if a preprocessor identifier is defined in the following manner:

```

#ifdef ECHO
    // code to be conditionally included goes here
#endif

```

Then you “turn on” or include this code using the following compilation command line:

```
gcc proj4.c -DECHO
```

(See the sample runs below.)

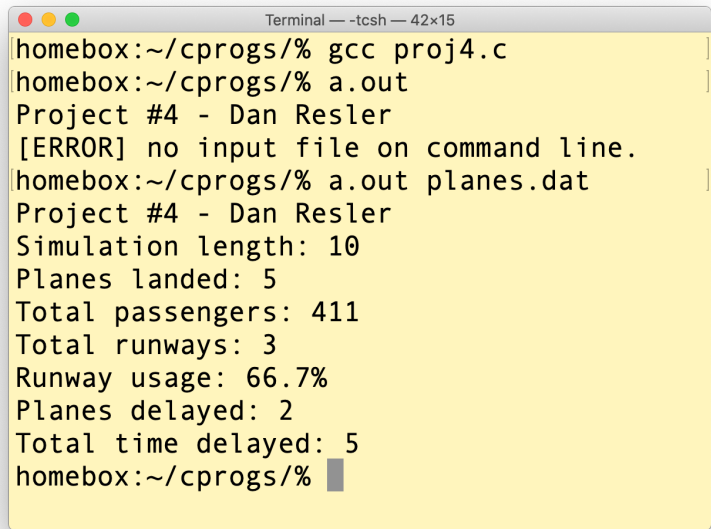
- Suggested data to include in definition of planes; this list may or may not be comprehensive or complete.
 - id number
 - name
 - scheduled landing time
 - needed time to land
 - number of people on board
 - time spent circling

- Suggested data to include in definition of a runway; this list may or may not be comprehensive or complete.
 - whether it's busy or not
 - time left for plane to exit runway
 - the plane on the runway

8 Sample Runs

Both runs are using the data file `planes.dat` presented above.

8.1 Statistics Only



```
Terminal — -tcsh — 42x15
homebox:~/cprogs/% gcc proj4.c
homebox:~/cprogs/% a.out
Project #4 - Dan Resler
[ERROR] no input file on command line.
homebox:~/cprogs/% a.out planes.dat
Project #4 - Dan Resler
Simulation length: 10
Planes landed: 5
Total passengers: 411
Total runways: 3
Runway usage: 66.7%
Planes delayed: 2
Total time delayed: 5
homebox:~/cprogs/%
```

8.2 Statistics and Event Reporting

```
homebox:~/cprogs/% a.out planes.dat
Project #4 - Dan Resler
**** time = 1 ****
-- waiting to land: [#0: Boeing727,schedule:1,needed-to-land:4,#onboard:112,circled:0]
-- landing runway 1: [#0: Boeing727,schedule:1,needed-to-land:4,#onboard:112,circled:0]
-- runway busy: #1
**** time = 2 ****
-- waiting to land: [#1: AirbusA320,schedule:2,needed-to-land:5,#onboard:208,circled:0]
-- landing runway 2: [#1: AirbusA320,schedule:2,needed-to-land:5,#onboard:208,circled:0]
-- runways busy: #1,#2
**** time = 3 ****
-- waiting to land: [#2: Cessna172,schedule:3,needed-to-land:2,#onboard:2,circled:0]
-- landing runway 3: [#2: Cessna172,schedule:3,needed-to-land:2,#onboard:2,circled:0]
-- runways busy: #1,#2,#3
**** time = 4 ****
-- waiting to land: [#3: Beechcraft,schedule:3,needed-to-land:2,#onboard:3,circled:0]
-- moving off runway #1: [#0: Boeing727,schedule:1,needed-to-land:4,#onboard:112,circled:0]
-- moving off runway #3: [#2: Cessna172,schedule:3,needed-to-land:2,#onboard:2,circled:0]
-- runway busy: #2
**** time = 5 ****
-- waiting to land: [#3: Beechcraft,schedule:3,needed-to-land:2,#onboard:3,circled:1]
-- landing runway 1: [#3: Beechcraft,schedule:3,needed-to-land:2,#onboard:3,circled:2]
-- runways busy: #1,#2
**** time = 6 ****
-- waiting to land: [#4: EmbraerE190,schedule:3,needed-to-land:3,#onboard:86,circled:0]
-- landing runway 3: [#4: EmbraerE190,schedule:3,needed-to-land:3,#onboard:86,circled:3]
-- moving off runway #1: [#3: Beechcraft,schedule:3,needed-to-land:2,#onboard:3,circled:2]
-- moving off runway #2: [#1: AirbusA320,schedule:2,needed-to-land:5,#onboard:208,circled:0]
-- runway busy: #3
**** time = 7 ****
-- waiting to land: [#5: Boeing767,schedule:3,needed-to-land:5,#onboard:214,circled:0]
-- landing runway 1: [#5: Boeing767,schedule:3,needed-to-land:5,#onboard:214,circled:4]
-- runways busy: #1,#3
**** time = 8 ****
-- waiting to land: [#6: Boeing767,schedule:4,needed-to-land:5,#onboard:159,circled:0]
-- landing runway 2: [#6: Boeing767,schedule:4,needed-to-land:5,#onboard:159,circled:4]
-- moving off runway #3: [#4: EmbraerE190,schedule:3,needed-to-land:3,#onboard:86,circled:3]
-- runways busy: #1,#2
**** time = 9 ****
-- waiting to land: [#7: AirbusA320,schedule:5,needed-to-land:5,#onboard:169,circled:0]
-- landing runway 3: [#7: AirbusA320,schedule:5,needed-to-land:5,#onboard:169,circled:4]
-- runways busy: #1,#2,#3
**** time = 10 ****
-- runways busy: #1,#2,#3
Simulation length: 10
Planes landed: 5
Total passengers: 411
Total runways: 3
Runway usage: 66.7%
Planes delayed: 2
Total time delayed: 5
homebox:~/cprogs/%
```

9 Deliverables

Submit your single file solution via Gradescope in the usual fashion. You may work in pairs on this project; submit only a single copy making sure that both of your names are on the file if you work

with someone else. Be sure to document your code in the manner previously discussed in class.

Due date: Wednesday October 27