BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI, K. K. BIRLA GOA CAMPUS

I SEMESTER 2012-2013

Operating Systems (CS C372 & IS C362)

Assignment 2

Due date 01/10/2012 (9.00 A.M) **Marks [5 + 15 = 20]**

Instructions:

(1) Please upload the assignment using **http://cc.bits-goa.ac.in/file**
(The file name should be <your id number>.tar.gz Example: **2012A7PS999G.tar.gz**)
You are allowed to upload only once.

(2) This is an **individual** assignment. Please see section 4b of handout for Malpractice regulations.

(3) The Assignment implementation should follow ADT with header files, implementation files, driver file & make file. No other implementation shall be considered for evaluation.

(4) The programming assignments will be graded according to the following criteria

- Completeness; does your program implement the whole assignment?
- Correctness; does your program provide the right output?
- Efficiency; have you chosen appropriate algorithms and data structures for the problem?
- Programming style (including documentation and program organization); is the program well designed and easy to understand?
- Viva.

DO NOT FORGET to include a README file (text only) in your tar.gz file with the following contents.

General README instructions

In the directory you turn in (please upload the assignment as a tar.gz file), you must have a text-only file called README, in which you will cover AT LEAST the following:

1. Your name. If you interacted significantly with others indicate this as well.
2. A list of all files in the directory and a short description of each.
3. HOW TO COMPILE your program and HOW TO USE (execute) your program.
4. A description of the structure of your program.
5. In case you have not completed the assignment, you should mention in significant detail:
   - What you have and have not done
   - Why you did not manage to complete your assignment (greatest difficulties)

   This will allow us to give you partial credit for the things you have completed.
6. Document any bugs of your program that you know of. Run-time errors will cost you fewer points if you document them and you show that you know their cause. Also describe what you would have done to correct them, if you had more time to work on your project.

Please refer section 4b of the handout to know more about Malpractice regulations of the course.

QUESTION 1: *Spiderman to the rescue!!!*

There is an NxN dimensional maze with each entry either 0 or 1.

Spiderman is at (0, 0). Mary Jane (MJ) is held up by the Green Goblin at ((N-1), (N-1)). Spiderman wants to rescue MJ as fast as possible. i.e.

Source: Top left corner of the matrix whose entry is 0. (Spiderman)

Destination: Bottom right corner of the matrix whose entry is 2. (MJ)

To get a quick way out, Spiderman asks his spies to help him for finding MJ. Each spy represents a process and Spiderman is the main process. Initially Spiderman who is at (0, 0), calls two spies which will check (0, 1) and (1, 0) entries. Depending on those entries, each will decide whether to call one or two other spies. If the spy is on the last row or column of the maze, it will call only one spy otherwise it will call two spies. This continues till a spy reaches MJ or no spy can find a way further.

Which means: Each process at (i, j) will have either one or two child processes which can find the possible path in the maze. These child processes read the entries that are to the right- (i, j+1) of and below- (i+1, j) the current entry- (i, j).

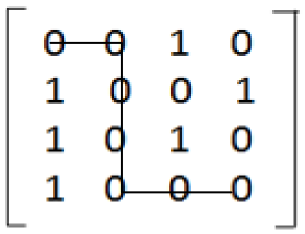
The right child in the tree (shown in Figure 2) corresponds to the child reading the entry to the right of the current process whereas the left child in the tree (shown in Figure 2) corresponds to the child reading the entry below the current entry.

In the maze, the path is created by 0's whereas 1's are the obstructions and 2 is the final destination where MJ is. So, if you have a 0 as the entry for current process, you will have a path further from that node by creating either one or two child processes. If the entry for current process is 1, you cannot have a path from that process and it dies.

The input will be such that there is only one possible path from the source to the destination.

You have to print the path in **reverse order** where each process that participated in the actual path must print the corresponding entry as an **(i, j) pair**.

The following is the input matrix.



$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Figure 1 : Input Matrix

The corresponding process tree for the above given input matrix will be:

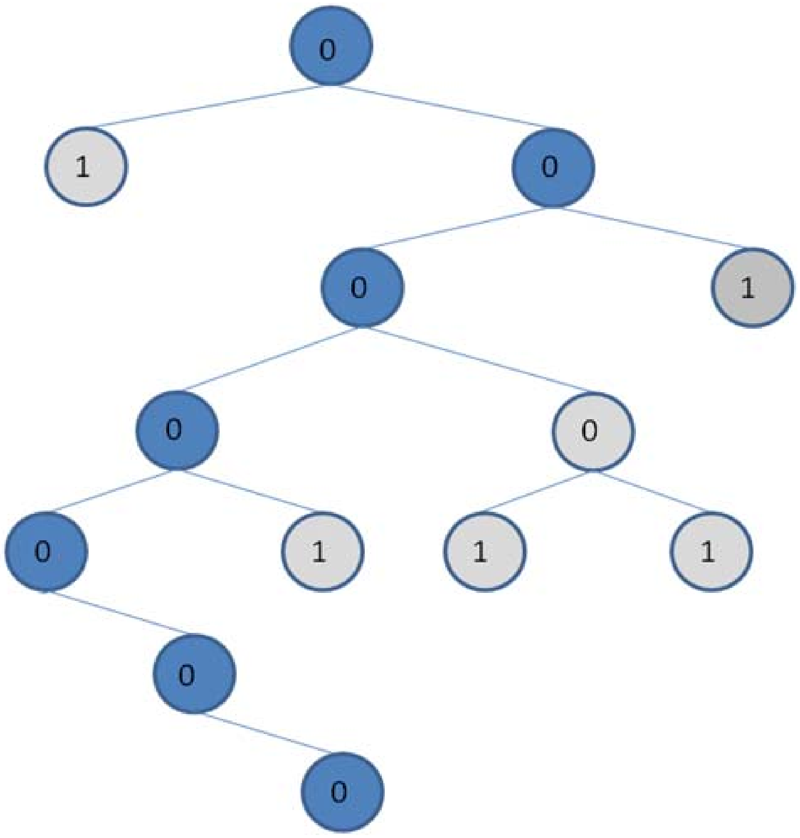


Figure 2 : The process tree

Sample Input 1:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Corresponding Output 1: (3,3) -> (3,2) -> (3,1) -> (2,1) -> (1,1) -> (0,1) -> (0,0)
MJ successfully rescued.

Sample Input 2:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Corresponding Output 2:
Spiderman cannot rescue MJ.

QUESTION 2: *Airport Departure System*

You are asked to simulate the departure process at an airport.

There are multiple processes at the airport. These are the following:

1. **Airport**
   This is the main process which reads all the input fields from input file before creating any children.
   This main process will fork the following child processes:
   - Baggage_Counter
   - Immigration
   - Security check
   - Waiting lounge
   - Boarding

   The description of each child process is given further in the question.
   The airport will now send the passenger to the Baggage_Counter if and only if the arrival time of the passenger is within 4 hours of the flight,(if a passenger is entering before 4 hour you should neglect that entry, because for such passengers there will be another entry with new Entry Time. So the number of entries in the input file is not equal to the number of passenger entering the airport).
2. **Baggage Counter**
   This process will basically have three child processes:
   (i) Luggage Security, (ii) Weight, (iii) Boarding_Pass.

   This process will signal the airport to check whether there are passenger in airport and waiting to proceed to baggage_counter (the airport will give start signal to Baggage_counter only if there are passenger that have entered the airport else wait till the next passenger enter the airport).

   Termination: The process should terminate only after all its children terminate. All the children terminate only after processing last passenger.

2.1. Luggage Security

This process will check the passenger's baggage for content which is prohibited to take into the flight.

(i) Liquids correspond to the "Colorful Numbers*"
(ii) Explosives correspond to the "Palindrome Numbers"**

***Given a number, say 263, is said to be colorful if the product of all its substrings is unique. 2, 6, 3, 2*6, 6*3, 2*6*3. [Consider substrings only.]**

****Palindrome numbers are those numbers which when read backwards give the same number. e.g. 121, 9009, etc.**

Refer Sample input section for the content of Baggage.

The luggage security process will eliminate such numbers from the baggage content and will print all the Liquids and Explosives on to the terminal after it receives signal from Weight Process, then it will append the Baggage ID (generated sequentially) in the boarding pass file generated by Boarding_Pass Process.

Now this will signal the Baggage_counter to send this passenger to immigration and send new passenger for luggage security.

Termination: The process should terminate after processing the last passenger.

2.2. Weight

The weight process for each passenger will resume only after his luggage security is done (not printing just checking of liquid and explosive).
A passenger is allowed to carry only 2 bags as luggage and 1 bag as a handbag. For each additional bag, a passenger will be charged $40.
Also, the weight of a person's bag must be below 20kgs. Each additional kg will be charged $2.
So this process will calculate the price due to extra luggage. If there is excessive luggage then this will ask for the Money from the passenger by printing the amount in the terminal.
This process will first calculate the price and then wait for signal from the Boarding_Pass process and print the amount after receiving signal from the Boarding_pass process. After this it will append the price that the passenger had paid in the boarding pass file generated by Boarding_pass process.
Termination: The process should terminate after processing the last passenger.

2.3. Boarding_Pass

The Boarding_Pass process for each passenger will resume only after his extra Baggage price is calculated.
This process should generate a Seat Number sequentially and then create a boarding pass file with the name same as that of ticket number. In this file it will print the passenger Name , Age , Sex,
Seat Number.
This process should also generate the meal that is to be served to that particular passenger at random. The meal can be of three types: Vegetarian, Non-Vegetarian, Asian Vegetarian. Out of these three, a type should be chosen at random and should append the same to the above file.
The following format should be maintained while appending the details to the file.
<Ticket number> <Age > <Sex> <Seat Number><Meal to be served>
(in single line)
Now it will signal weight process (to print the price paid by passenger due to extra luggage)
Termination: The process should terminate after processing the last passenger.

3. Immigration

The Passenger will enter this process only after it is done with Baggage_Counter.
This process forks two processes namely VISA and Passport.
These two processes check for the validity of the VISA and Passport respectively and report to Immigration.
After the completion of this procedure, the passenger will now enter the Security Check process.
Termination: The process should terminate after processing the last passenger.

Passport

This process will compare the expiry date of the Passport with the Current Date. If the passenger has a return ticket then check the expiry Date of passport with that date.
If the passenger is rejected then append to its boarding pass that “Immigration Not Successful” and do not send the passenger to the VISA process and go back to Immigration process for next passengers.
Else send the passenger to VISA.
Termination: The process should terminate after processing the last passenger.

VISA

If the passenger is a citizen of destination country then check the VISA expiry date of the current country with today’s date.
Else if passenger has return ticket then check its Visa expiry with return Date and todays Date.
Else only check the Visa Expiry with current Date.
Example: Assume that Airport is in INDIA and flight destination is England so,
if passenger is citizen of England then check its India Visa expiry date with Current Date
Else if has return ticket check whether its Visa is valid till its return
Else check its VISA expiry with the current Date only.

If the passenger is rejected then append to its boarding pass that “Immigration Not Successful”
Else Append “Immigration Successful”
And go back to Immigration process for next passenger.
Termination: The process should terminate after processing the last passenger.

4. **Security Check**

This process will have 2 child processes:
Security_M
Security_F
The passenger will enter Security check after the immigration process.
Here it will first check whether its immigration was Successful or not.
If not then it will give signal to airport to throw this passenger out of airport and print to the terminal.
If Successful then check whether it is a Male or Female and send to corresponding child process:
Security_M
Security_F
After the last passenger Security will give Signal to its child processes to exit. Along with this it will also signal the Waiting_lounge saying that this was last passenger.

4.1 Security_M or Security_F :

This process will check the content of handbag and remove the explosive and liquid as describe in the Luggage_Security. Apart from these prime numbers are not allowed for Security Reasons.

So this process will print all the rejected numbers to the terminal along with the passenger ticket number.
This will signal the waiting_Lounge that one more passenger is finished security check.

5. **Waiting Lounge**

This process will contain all the passengers that have undergone the security check. The passengers will stay here till the last passenger comes out of the security check. Unless and until this happens, the boarding for the flight will not start.
In the meanwhile, the waiting lounge forks two child processes namely Café and restroom. A passenger can visit these processes randomly and spends maximum of 1 second in each one.
Termination: The process should terminate after final announcement from Boarding process is received.

5.1. Café

This process must print the ticket number of the entering passenger along with the café number on terminal.
Print format-
Passenger <Ticket Number> entered Café 1
After 1 second, the passenger will return to the Waiting_Lounge.
Termination: The process should terminate after processing the last passenger.

5.2. Restroom

This process works in the same way as the Café process.
Termination: The process should terminate after processing the last passenger.

6. **Boarding**

On receiving signal from Waiting_Lounge, this process will announce the first boarding call to all the passengers by printing-
"All passengers are requested to move to the Gate."

If still some passengers are not reporting the boarding counter i.e. some passengers still in café or restroom, then Boarding process will make another boarding call saying

"This is the last boarding call for passenger <Ticket Number>. Please report immediately to the gate"

This should be printed on the terminal.

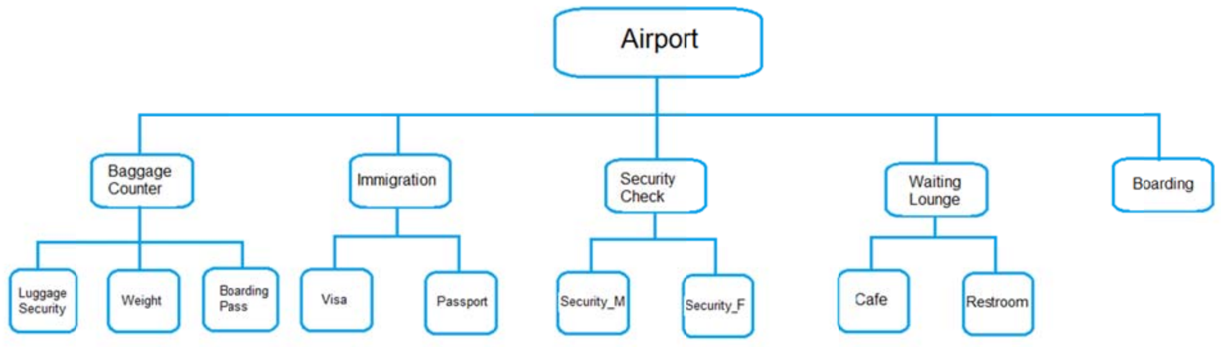
After waiting for 2 seconds, Boarding process will give signal to the Café, Restroom and Waiting_Lounge to exit.

Then it will give signal to airport and the Airport will print-
Total number of passengers arrived at the airport = <count1>
Total number of passengers boarded = <count2>

Flight Number #<PID of airport process> ready for departure with <count> passengers.
Plane Flew!

THE END…

Process Diagram:



Input format:

<Flag to denote return ticket there or not> <entry time> <entry date> <ticket no.> <flight no.> <flight time> <flight date> <destination> <return ticket no.> <return flight no.> <return time> <return date> <return to> <first name> <last name> <gender> <age> <no. of bags> <weight of bag1> <weight of bag2>.... <10 contents of bag1> <10 contents of bag2>...

If the passenger do not have the return ticket then there will be no entry corresponding to the return ticket. The last bag will be the handbag for the passenger and this above entry for one passenger will be in one line.

Example:

```
2

1         8:00 09/09/2012                    ABCD123 BA787 10:00 09/09/2012 England
          ABCD214 BA707 5:00 2/11/2012 India                 J0126541G 12/01/2017 India
          5/11/2012 England                  Jaykare Shinde M 21           3        15 20 10         11 6 12
3 24 9 37 8 54 19        21 61 22 33 4 59 27 18 4 29              15 62 32 63 24 79 17 58 51 9


0         8:00 09/09/2012                    XBCD151 BA787 09:00 09/10/2012 England
          H0122541G 12/01/2017 Australia                5/11/2012 England

     Maya Vati F 21 4  25 20 10 15                1 26 32 43 14 29 57 8 84 9                 11  41  22  3  24
79 7 98 14 99            11 22 32 43 54 69 77 88 91 0 01 02 12 03 04 69 77 88 91 0
```

First entry has the return ticket and second entry does not have return ticket.