

The Elevator Problem

Semaphores and Threads

In a medium-sized city somewhere in Pennsylvania stands a building with N floors, numbered from 0 to $N-1$, and an elevator. The P people who work in the building use the elevator to move from one floor to another. At exactly 8 o'clock every weekday morning, these people enter the building and take the elevator to the floors they work on. The elevator is waiting with open doors at 8 o'clock and is big enough to hold all P people at once. From 8 o'clock in the morning until 5 o'clock in the afternoon, the people wander around the building, using the elevator to move from floor to floor as needed. The wandering time is at least 1 second in any floor they visit and is less than *maxWanderTime*. At the end of the day at 5 o'clock they always come back to floor 0 and wander there and then go home.

Your assignment is to write a multithreaded program that simulates the movement of the people in the building and their use of the elevator. Use semaphores to synchronize the elevator thread and the people threads.

The program operates in a deterministic mode, where the itinerary of each person for the day is provided ahead of time.

In the deterministic mode, each person operates according to an input script, which determines which floor the person goes to and how long the person wanders around on that floor before using the elevator to go to another floor. Each person always start at floor 0, and the last floor they come to is also 0.

When a person wants to use the elevator to go to another floor, the person pushes the elevator call button, waits for the elevator to arrive and open its doors, enters the elevator, pushes the button of the destination floor, and waits for the elevator to arrive at that floor. Because this is an old building, each floor has just a single button to call the elevator, not separate "up" and "down" buttons we are more accustomed to seeing.

The elevator continually goes up and down, picking up and dropping off passengers. The elevator goes up until it reaches the highest floor $N-1$ and then goes down to the lowest floor 0. While going up and down like this, the elevator picks up and drops off passengers along the way. The important feature of the elevator algorithm is that the elevator does not reverse direction until it has gone all the way to the end in one direction as needed to pick up and/or drop off passengers.

The elevator takes one second to move from a floor to the next higher or lower floor. If the elevator needs to stop at a particular floor to pick up or discharge people, it takes one second to

do so, no matter how many people embark or disembark. The elevator stops at a floor only if there are any passengers that need to get off or get on at that floor.

After each person returns to floor 0 and wanders and leaves the system, the person exits. When the elevator travels up and down the last time and sees that it has no passengers were picked up or dropped off, it sleeps for `maxWanderTime` and then checks to see anyone is waiting at any of the floors. If no one is seen still, it exits the system

Your program will have an elevator thread is instantiated, and a person thread instantiated for each person. Then place all the semaphores and other state variables as data fields. The `main` method starts everything going. The person thread blocks on a semaphore until picked up by the elevator. When picked up, the person object indicates the desired destination floor. The person thread blocks on a semaphore until the elevator arrives at the floor. When arriving at a floor, the elevator thread naps for one second to let passengers get on and off.

An outline of person thread.. There is lot of code missing here.. This is one way you can think about what a person thread does in the program.

```
public void person(...) {
    while (notDone) {
        sleep(wanderTime);
        wait for the elevator to come to current floor
        get on the elevator;
        figure out the next floor to go to;
        wait for elevator to reach there;
        get off the elevator;
    }
    When done print good bye message and exit thread;
}
```

Input Data

The input data to your program are the number of people in the building, the maximum wander time (napping time) and the number of floors in the building. Place these numbers on the command line for your program to parse.

```
%elevator.o -p  $N_{people}$  -w maxWanderTime -f  $N_{floors}$  < file
```

-p <numberOfPeople> If -p is not given numOfPeople is set to 1 as default

-w <maxWanderTime> If -w is missing the maximum wander Time is set to 10 seconds

-f <NumberOfFloors> If -f is missing, the number of floors is set to 10

< file helps read the data from a file

Remember to give the corresponding variables default values in your program to handle missing command line options.

In the *file*, For each person, one or more lines of the form

$n \ f_1 \ t_1 \ f_2 \ t_2 \ f_3 \ t_3 \ \dots \ f_n \ t_n$

are present. Here, n is the number of elevator ride/floor wandering pairs for the person and the pair $f_i \ t_i$ means the person takes the elevator to floor f_i and wanders around on that floor for exactly t_i seconds before using the elevator to go to another floor.

Everybody must include the following three deterministic examples in their submitted sample runs.

```
% ./elevator.o -p 1 -f 11
3 2 5 10 5 0 5
% ./elevator.o -p 5 -f 11 -w 10
3 1 1 10 1 0 5
3 3 1 10 2 0 5
3 5 1 10 3 0 5
3 7 1 10 4 0 5
3 9 1 10 5 0 5
% ./elevator.o -p 5 -f 12 -w 10
10 2 1 11 1 1 5 2 1 3 1 4 1 5 1 6 1 7 1 0 2
10 4 1 11 2 1 5 3 1 4 1 5 1 6 1 7 1 8 1 0 2
10 6 1 11 3 1 5 4 1 5 1 6 1 7 1 8 1 9 1 0 2
10 8 1 11 4 1 5 5 1 6 1 7 1 8 1 9 1 10 1 0 2
10 10 1 11 5 1 5 6 1 7 1 8 1 9 1 10 1 11 1 0 3
```

Be sure to (stress) test your program with the following kinds of input data.

- One person in the building who wanders (naps) for a long time on each floor
- Many people wandering for short periods of time, keeping the elevator busy.

Output Requirements:

1. Output the input that is read in the following format
Person i: List of Floors To Visit:
Person i : Amount of Time to Spend
2. All outputs from elevator thread must be labeled “Elevator: “ It must also be pushed several spaces to the right to distinguish it from output from person threads.
Output actions of the elevator when
 - a) elevator stops at certain floor
 - b) when it starts its upward journey
 - c) when downward journey
 - d) when it leaves the system

3. All outputs from person thread must be labeled “Person <i>: “ It must be left aligned to distinguish it from output from elevator threads.

Output actions of the people with their id,

a) when person starts to wait to embark on the elevator

b) when they embark the elevator

c) when they wait to disembark the elevator

d) when they start to wander

e) when they leave the system.

4. Each time the elevator starts its upward journey, output the number of people waiting at each floor:

Getting Started with Threads

1. Use the library `pthread` which contains all the thread related functions. Make sure you have `#include <pthread.h>` in your include list. While compiling the source file use `-lpthread` at the end of compilation to include the thread library as follows

```
%gcc main.c -o main.out -lpthread
```

2. `pthread_create` routine will create a thread. Once a thread is created it starts running the code specified. The order of execution or speed of each thread is unknown. The following code creates 5 threads.

```
int main()
{
    pthread_t thread_id[5];

    // The following code creates 5 threads.
    for (int i =0; i <5; i++)
        pthread_create(&thread_id [i], NULL, thread_task, (void *) i);

    // The following code makes sure the main program waits until all threads have finished
    execution

    for (int i =0; i <5; i++)
        pthread_join(thread_id[i], NULL);

}

void thread_task(int i) {
    .....
    pthread_exit(0); // this code returns to the corresponding pthread_join issued in main()
}
```

Getting Started with Semaphores

You will be using the library <semaphore.h> for the semaphore functions. You do not need to do anything special to compile the code with semaphore.h functions. The following functions would be most useful.

```
sem_t mutex;  
sem_int(&mutex, 0, 1); // will initialize the value of the mutex to 1. You can initialize the mutex  
to any value of your choice.  
sem_wait(&mutex); // If the mutex value is 1 or more the thread does not block, and it reduces  
the mutex value by 1; If the mutex value is 0 or below the thread executing this code will block  
until another thread wakes it up executing sem_post  
sem_post(&mutex) // Increase the mutex value by 1 and unblock a thread that is blocked at  
sem_wait.
```

Internal Requirements

1. Use sleep for wander time and alighting time for elevator.
2. DO NOT USE ANY OTHER SYNCHRONIZATION like locks or anything else to synchronize
3. Note that you can have global variables and semaphores that are global, however NO GLOBAL VARIABLES CAN BE READ OR WRITTEN w/o semaphore protection.
4. Note that elevator thread simply goes all the way up and comes down all the way checking to see if it should let passengers in or out along the way. If it just left floor 3, then some person wanted to get on at floor 3, that person has to wait until the elevator goes all the way up and the elevator comes downward. On the way down the elevator will pick up the passenger at floor 3.
5. NOTE : Since people do not know which direction the elevator is headed, it is possible that a person takes elevator to Floor 10 when the elevator is headed down to Floor 0. However, next time the elevator goes upward and reaches Floor 10, this person gets off.

EXTRA CREDIT: If the person is aware of the direction of the elevator and takes the elevator only if the direction is correct you get extra credit. You must first submit a code for regular credit and also add separate code for extra credit. The extra credit will be graded only if the regular credit works correctly.

Sample Input-Output:

Input Being Printed Here from the Program:

Person 0: Floors To Visit 1,10,0

Person 0: Time To Spend 1,1,5

Person 1: Floors To Visit 3,10,0
Person 1: Time To Spend 1,2,5
Person 2: Floors To Visit 5,10,0
Person 2: Time To Spend 1,3,5
Person 3: Floors To Visit 7,10,0
Person 3: Time To Spend 1,4,5
Person 4 Floors To Visit 9,10,0
Person 4: Time To Spend 1,5,5

Person Number 2: Waiting for elevator at floor 0
Person Number 0: Waiting for elevator at floor 0
Person Number 4: Waiting for elevator at floor 0
Person Number 1: Waiting for elevator at floor 0

Printout From the Elevator:

Number of people waiting at floor

Floor Number	Number of People
0	4
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0

Elevator: At floor 0

Elevator: Heading to max Floor 10

Person Number 3: Waiting for elevator at floor 0

Elevator: Opening the doors at 0
Elevator: Opening the doors at 0
Elevator: Opening the doors at 0
Elevator: Opening the doors at 0

Person Number 1: Taking elevator to floor 3
Person Number 1: Waiting to get off at floor 3
Person Number 0: Taking elevator to floor 1
Person Number 4: Taking elevator to floor 9
Person Number 4: Waiting to get off at floor 9
Person Number 0: Waiting to get off at floor 1
Person Number 2: Taking elevator to floor 5
Person Number 2: Waiting to get off at floor 5

Person Number 0: Got off at floor 1
Person Number 0: Wandering for 1 seconds
Person Number 0: Waiting for elevator at floor 1

Person Number 1: Got off at floor 3
Person Number 1: Wandering for 1 seconds
Person Number 1: Waiting for elevator at floor 3

Person Number 2: Got off at floor 5
Person Number 2: Wandering for 1 seconds
Person Number 2: Waiting for elevator at floor 5

Person Number 4: Got off at floor 9
Person Number 4: Wandering for 1 seconds
Person Number 4: Waiting for elevator at floor 9

Person Number 4: Taking elevator to floor 10
Person Number 4: Waiting to get off at floor 10

Person Number 2: Taking elevator to floor 10
Person Number 2: Waiting to get off at floor 10

Person Number 1: Taking elevator to floor 10
Person Number 1: Waiting to get off at floor 10

Person Number 0: Taking elevator to floor 10
Person Number 0: Waiting to get off at floor 10

Person Number 3: Taking elevator to floor 7
Person Number 3: Waiting to get off at floor 7

Number of people waiting at floor

Floor Number	Number of People
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	1
8	0
9	0

Elevator Opening the doors at 1

Elevator: Opening the doors at 3

Elevator: Opening the doors at 5

Elevator: Opening the doors at 9

Elevator: At floor 10
Elevator: Heading to min Floor 0
Elevator: At floor 9

Elevator: At floor 5

Elevator: At floor 3

Elevator: At floor 1

Elevator: At floor 0

Person Number 3: Got off at floor 7
 Person Number 3: Wandering for 1 seconds
 Person Number 3: Waiting for elevator at floor 7

Person Number 0: Got off at floor 10
 Person Number 0: Wandering for 1 seconds

Person Number 2: Got off at floor 10
 Person Number 1: Got off at floor 10
 Person Number 1: Wandering for 2 seconds
 Person Number 2: Wandering for 3 seconds
 Person Number 4: Got off at floor 10
 Person Number 4: Wandering for 5 seconds
 Person Number 0: Waiting for elevator at floor 10

Person Number 1: Waiting for elevator at floor 10
 Person Number 2: Waiting for elevator at floor 10

Person Number 0: Taking elevator to floor 0
 Person Number 0: Waiting to get off at floor 0
 Person Number 1: Taking elevator to floor 0
 Person Number 1: Waiting to get off at floor 0
 Person Number 4: Waiting for elevator at floor 10

Person Number 3: Taking elevator to floor 10
 Person Number 3: Waiting to get off at floor 10

Person Number 0: Got off at floor 0
 Person Number 0: Wandering for 5 seconds
 Person Number 1: Got off at floor 0
 Person Number 1: Wandering for 5 seconds
 Number of people waiting at floor

Printout from Elevator

Floor Number	Number of People
0	0

Elevator: At floor 0
 Elevator: Heading to max Floor 10
 Elevator: Opening the doors at 7

Elevator: Opening the doors at 10
 Elevator: Opening the doors at 10
 Elevator: Opening the doors at 10

Elevator: Opening the doors at 10

Elevator: At floor 10
 Elevator: Heading to min Floor 0

Elevator: At floor 10
 Elevator: At floor 10

Elevator: At floor 7

Elevator: At floor 0
 Elevator: At floor 0

1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	3

Elevator: At floor 0

Elevator: Heading to max Floor 10

Person Number 0: Leaving the System Goodbye!

Person Number 1: Leaving the System Goodbye!

Elevator: Opening the doors at 10

Elevator: Opening the doors at 10

Elevator: Opening the doors at 10

Person Number 3: Got off at floor 10

Person Number 2: Taking elevator to floor 0

Person Number 2: Waiting to get off at floor 0

Person Number 3: Wandering for 4 seconds

Person Number 4: Taking elevator to floor 0

Person Number 4: Waiting to get off at floor 0

Elevator: At floor 10

Elevator: Heading to min Floor 0

Person Number 3: Waiting for elevator at floor 10

Elevator: At floor 0

Elevator: At floor 0

Person Number 2: Got off at floor 0

Person Number 4: Got off at floor 0

Person Number 4: Wandering for 5 seconds

Person Number 2: Wandering for 5 seconds

Printout From Elevator

Number of people waiting at floor

Floor Number	Number of People
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0

9 0
10 1

Elevator: At floor 0
Elevator: Heading to max Floor 10
Person Number 4:Leaving the System Goodbye!
Person Number 2:Leaving the System Goodbye!
Elevator: Opening the doors at 10
Person Number 3: Taking elevator to floor 0
Person Number 3: Waiting to get off at floor 0
Elevator: At floor 10
Elevator: Heading to min Floor 0
Elevator: At floor 0
Person Number 3: Got off at floor 0
Person Number 3: Wandering for 5 seconds

Printout From Elevator

Number of people waiting at floor
Floor Number Number of People
0 0
1 0
2 0
3 0
4 0
5 0
6 0
7 0
8 0
9 0
10 0

Person Number 3:Leaving the System Goodbye!

Elevator: At floor 0
Elevator: Heading to max Floor 10
Elevator: At floor 10
Elevator: Heading to min Floor 0

Waiting for max waiting time. No one is waiting for the elevator.
Elevator Leaving The System