CIS 657: Operating System
Lab 3
2019 Spring

Dazhuang Zhang
SUID:927645874
Engineering & Computer Science, Syracuse University

2019
March

1. Did you consult with anyone other than instructor or TA/grader on parts of this assignment? If Yes, please give the details.

2. Did you consult an outside source such as an Internet forum or a book on parts of this assignment? If Yes, please give the details.

I assert that, to the best of my knowledge, the information on this sheet is true.

Signature: **Date:**

Contents

1 Program Implematation

1.1 Run Configuration

This program started in ThreadTest(), so there is no need to add a new command configuration in main.cc for this Hotel Reservation System. We can just use **-K** as our command branch.

1.2 Introduction

For this program we need to keep track of guests' information and the availability of rooms. I use class List to store guests as elements, since they would be used in different functions we set them as global variables and also total simulation days and specific simulation day. Then I design a couple of print functions to help me output the information we need and since after 11 days we are asked to print summary of all past days, a function daily_work designed to record occupancy rate and granted rate every day, and I use vectors to store all this information like vector granted_rate[0] means granted rate at day 1.

For Concierge thread, I set a condition loop make sure this thread will run every day with no check information at first day(since no guest at that time) and no new guest thread generated at last day. Then I set a condition loop to make sure all guest threads sleeped or finished then back to concirege thread to run.

Also I design map to keep track of rooms' availability instead of using class Bitmap. Similarly I use map to store guest threads at sleep status with guest unique id as index(since each guest thread only deal with one guest request).

1.3 Modification for List

Similar with Lab 2, I did a modification for class List as follows:

```
template<class T>
ListIterator<T> List<T>::GetIterator() {
return ListIterator<T>(this);}
```

So we can use GetIterator to return the first element in the list.

1.4 Solutions for requirements

1.4.1 Simulation days and unique ID

We set the global variable to make sure simulation day is 11:

```
std::size_t day = 11;
```

For requirement that check out all staying guests, I just made a small change for check out date of guests, if one guest check out date is later than 11th day, then change check out date to 11th day and also number of nights get changed correspondingly:

```
size_t room_num = (rand() % 5) + 1;
new_req.setRooms_num(room_num);
size_t night_num = (rand() % 4) + 1;
size_t restday = day - today;
size_t check_in = (rand() % restday) + today;
size_t check_out = check_in + night_num;
if (check_out > day) {
    check_out = day;
    night_num = check_out - check_in;
}
```

```
if (today != day) {
  for (size_t i = 0; i < num; ++i) {
    size_t Id = 5 * (today - 1) + i + 1;
    Thread *gt = new Thread(”guest thread”);
    gt->Fork((VoidFunctionPtr) guest_thread, (void *) Id);
  }
}
```

From the code above we can see that at 11th day, no more new guest request could be generated.
For guest unique ID, I designed it based on first-come-first-serve basis, for example, five guests' ID on first day should be 1,2,3,4,5 and then 6,7,8,9,10 on second day and so on. Hence each guest will get his own unique ID which is very important for us to store and call guest threads.

I have mentioned that I use map to keep track of the availability of rooms:

```
vector<int> room_use(30); //check if room is available
map<int, vector<int >> room_available;

void map_initialize() {
    for (int i = 1; i <= day; ++i) {
         room_available[i] = room_use;
    }
}
```

According to the code above, i is simulation date as index. Then vector is the room availability of each day. If this room is available, then set it 0 otherwise it would be 1. Room ID could just set 1 to 30, just vector's index plus 1.

1.4.2 Rooms' Availability

Like I have mentioned above, map<int, vector<int >> room_available is used to keep track of the availability of rooms. I will give specific details for that:

```
    //check room available numbers
int total_count = 0;
for (size_t i = 0; i < room_available[check_in].size(); ++i) {
  int count = 0;
  for (int j = check_in; j < check_out; ++j) {
    if (room_available[j][i] == 0) {
      count += 1;
    }
  }
  if (count == night_num) { total_count += 1; }
}
```

```
//check if room is available
if (total_count >= new_req.getRooms_num()) {
  int c_total_count = 0;
  vector<size_t> room_id;
  for (size_t i = 0; i < room_available[check_in].size(); ++i) {
    int c_count = 0;
    for (int j = check_in; j < check_out; ++j) {
      if (room_available[j][i] == 0) {
        c_count += 1;
      }
    }
    if (c_count == night_num) {
      for (int j = check_in; j < check_out; ++j) {
        room_available[j][i] = 1;
      }
      c_total_count += 1;
      room_id.push_back(i + 1);
    }
    if (c_total_count == new_req.getRooms_num())break;
  }
  new_req.setRooms_Id(room_id);
```

In order to simulate system like real life, guests will live in the same rooms until they check out. There is no reason that guests always change their rooms, which brings a lot of inconvenience. So room is main subject. I checked each room's availability from guests' check in date to their check out date. All these night are vacant and then this room is available, then total counts plus 1.

If total counts large than rooms number guests need, it means that hotel has enough room for them, so request would be confirmed and these rooms would be set to 1 in vector from check in date to check out date which means they are not available at that time anymore. If total counts less than rooms number guests need, this request would be discarded and recorded by List discard.

1.4.3 Guest Class and Information

Class guest is designed in file guest.h and all the member functions are defined in file guest.cc. We can see that from source code at section **Code Snippets**.

I applied four list to keep track of guests' information.

```
std::unique_ptr<SortedList<guest>> staying = std::unique_ptr<SortedList<guest>>(new SortedList<guest>(&guest_check_out_compare));
List<guest> checkout;
std::unique_ptr<SortedList<guest>> confirmed = std::unique_ptr<SortedList<guest>>(new SortedList<guest>(&guest_check_in_compare));
List<guest> discard;
```

Since staying list and confirmed list should be sorted in specific order, I applied class SortedList to design these two list, each list sorted by different ways. There are two functions declared and defined in file guest.h/guest.cc used for sort list.

1.4.4 Thread

It's not very hard to set Concierge thread to create five new guest threads everyday(except 11^{th} day) after finish its daily work(call those guest thread wake up from sleep status and ready to run). The key is to make sure after call **Yield()**, the Concierge thread will run again after all other guest threads have finished or slept. I designed a conditional loop to solve it.

```
if (today != day) {
  while (1) {
    if (guest_thread_count != 5) {
      kernel->currentThread->Yield();
    } else { break; }
  }
} else { kernel->currentThread->Yield(); }
```

Each guest threads runs, the guest thread count will plus 1, the Concierge thread will run again until all other five guest threads given up the CPU.

How to avoid interrupt? Because scheduler->ReadyToRun assume that there is no interrupt. I learned how to set interrupt off in Yield() definition:

```
kernel->interrupt->SetLevel(IntOff);
```

As for solution of how concierge thread and guest thread work, We can see that from source code with comments at section **Code Snippets**.

2 Tests and Outputs

2.1 Test case

Since our program only need a simulation of 11 days. In order to test my system, I change it into a 30 days simulation. And the output file is in folder **output and text case**. We can test different cases by just changing global variable **day** to different value. Since output is so long, I just select a fragment of it.

```
day 23
Guest has checked out.
The guest's information is as follows:
Guest id: 33
Rooms number: 4
Rooms Id: 10 11 12 13
Check in date: 19
Check out date: 23
Nights number: 4

The available rooms number today is 10

day 23
Guest has checked out.
The guest's information is as follows:
Guest id: 44
Rooms number: 3
Rooms Id: 15 16 17
Check in date: 19
Check out date: 23
Nights number: 4

The available rooms number today is 10

day 23
Guest has checked out.
The guest’s information is as follows:
Guest id: 72
Rooms number: 3
Rooms Id: 9 27 28
Check in date: 21
Check out date: 23
Nights number: 2

The available rooms number today is 10

day 23
Guest has checked out.
The guest’s information is as follows:
Guest id: 88
Rooms number: 1
Rooms Id: 14
Check in date: 22
Check out date: 23
Nights number: 1

The available rooms number today is 10

day 23
Guest has checked out.
The guest’s information is as follows:
Guest id: 91
Rooms number: 1
Rooms Id: 25
Check in date: 22
Check out date: 23
Nights number: 1

The available rooms number today is 10

day 23
Guest has checked in.
The guest's information is as follows:
Guest id: 63
Rooms number: 5
Rooms Id: 13 14 26 27 28
Check in date: 23
Check out date: 25
Nights number: 2

day 23
Request has been discarded.
The guest's information is as follows:
Guest id: 111
Rooms number: 5
Rooms Id: No request granted.
Check in date: 27
Check out date: 30
Nights number: 3

day 23
Request has been discarded.
The guest's information is as follows:
Guest id: 113
Rooms number: 3
Rooms Id: No request granted.
Check in date: 29
Check out date: 30
Nights number: 1

day 23
```

```
Request has been discarded.
The guest's information is as follows:
Guest id: 114
Rooms number: 3
Rooms Id: No request granted.
Check in date: 27
Check out date: 30
Nights number: 3

day 23
Request has been made.
The guest's information is as follows:
Guest id: 115
Rooms number: 4
Rooms Id: 18 24 28 29
Check in date: 26
Check out date: 27
Nights number: 1

The available rooms number today is 10

day 23
Request has been confirmed.
The guest's information is as follows:
Guest id: 115
Rooms number: 4
Rooms Id: 18 24 28 29
Check in date: 26
Check out date: 27
Nights number: 1

day 23
Request has been discarded.
The guest's information is as follows:
Guest id: 112
Rooms number: 3
Rooms Id: No request granted.
Check in date: 28
Check out date: 30
Nights number: 2
```

2.2 Output

Here is the output text of our program system:

```
/home/big/Documents/NachOS_w_CMake-master/code/cmake-build-debug/NACHOS_ENTRANCE -K
day 1
Request has been made.
The guest’s information is as follows:
Guest id: 2
Rooms number: 4
Rooms Id: 1 2 3 4
Check in date: 8
Check out date: 11
Nights number: 3

The available rooms number today is 30

day 1
Request has been confirmed.
The guest’s information is as follows:
Guest id: 2
Rooms number: 4
Rooms Id: 1 2 3 4
Check in date: 8
Check out date: 11
Nights number: 3

day 1
```

```
Request has been made.
The guest's information is as follows:
Guest id: 3
Rooms number: 1
Rooms Id: 1
Check in date: 6
Check out date: 8
Nights number: 2

The available rooms number today is 30

day 1
Request has been confirmed.
The guest's information is as follows:
Guest id: 3
Rooms number: 1
Rooms Id: 1
Check in date: 6
Check out date: 8
Nights number: 2

day 1
Request has been made.
The guest's information is as follows:
Guest id: 4
Rooms number: 2
Rooms Id: 5 6
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 30

day 1
Request has been confirmed.
The guest's information is as follows:
```

```
Guest id: 4
Rooms number: 2
Rooms Id: 5 6
Check in date: 10
Check out date: 11
Nights number: 1

day 1
Request has been made.
The guest’s information is as follows:
Guest id: 5
Rooms number: 2
Rooms Id: 7 8
Check in date: 8
Check out date: 11
Nights number: 3

The available rooms number today is 30

day 1
Request has been confirmed.
The guest’s information is as follows:
Guest id: 5
Rooms number: 2
Rooms Id: 7 8
Check in date: 8
Check out date: 11
Nights number: 3

day 1
Request has been made.
The guest’s information is as follows:
Guest id: 1
Rooms number: 1
Rooms Id: 2
Check in date: 4
Check out date: 8
Nights number: 4

The available rooms number today is 30

day 1
Request has been confirmed.
The guest's information is as follows:
Guest id: 1
Rooms number: 1
Rooms Id: 2
Check in date: 4
Check out date: 8
Nights number: 4

day 2
Request has been made.
The guest's information is as follows:
Guest id: 6
Rooms number: 2
Rooms Id: 1 2
Check in date: 2
Check out date: 3
Nights number: 1

The available rooms number today is 28

day 2
Guest has checked in.
The guest's information is as follows:
Guest id: 6
Rooms number: 2
Rooms Id: 1 2
Check in date: 2
Check out date: 3
Nights number: 1
```

```
day 2
Request has been made.
The guest’s information is as follows:
Guest id: 7
Rooms number: 3
Rooms Id: 9 10 11
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 28

day 2
Request has been confirmed.
The guest’s information is as follows:
Guest id: 7
Rooms number: 3
Rooms Id: 9 10 11
Check in date: 10
Check out date: 11
Nights number: 1

day 2
Request has been made.
The guest’s information is as follows:
Guest id: 8
Rooms number: 4
Rooms Id: 12 13 14 15
Check in date: 8
Check out date: 11
Nights number: 3

The available rooms number today is 28

day 2
```

```
Request has been confirmed.
The guest’s information is as follows:
Guest id: 8
Rooms number: 4
Rooms Id: 12 13 14 15
Check in date: 8
Check out date: 11
Nights number: 3

day 2
Request has been made.
The guest’s information is as follows:
Guest id: 9
Rooms number: 3
Rooms Id: 16 17 18
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 28

day 2
Request has been confirmed.
The guest’s information is as follows:
Guest id: 9
Rooms number: 3
Rooms Id: 16 17 18
Check in date: 10
Check out date: 11
Nights number: 1

day 2
Request has been made.
The guest’s information is as follows:
Guest id: 10
Rooms number: 4
Rooms Id: 3 4 5 6
Check in date: 3
Check out date: 7
Nights number: 4

The available rooms number today is 28

day 2
Request has been confirmed.
The guest's information is as follows:
Guest id: 10
Rooms number: 4
Rooms Id: 3 4 5 6
Check in date: 3
Check out date: 7
Nights number: 4

day 3
Guest has checked out.
The guest's information is as follows:
Guest id: 6
Rooms number: 2
Rooms Id: 1 2
Check in date: 2
Check out date: 3
Nights number: 1

The available rooms number today is 26

day 3
Guest has checked in.
The guest's information is as follows:
Guest id: 10
Rooms number: 4
Rooms Id: 3 4 5 6
Check in date: 3
Check out date: 7
Nights number: 4

day 3
Request has been made.
The guest’s information is as follows:
Guest id: 12
Rooms number: 5
Rooms Id: 19 20 21 22 23
Check in date: 9
Check out date: 11
Nights number: 2

The available rooms number today is 26

day 3
Request has been confirmed.
The guest’s information is as follows:
Guest id: 12
Rooms number: 5
Rooms Id: 19 20 21 22 23
Check in date: 9
Check out date: 11
Nights number: 2

day 3
Request has been made.
The guest’s information is as follows:
Guest id: 13
Rooms number: 4
Rooms Id: 24 25 26 27
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 26
```

```
day 3
Request has been confirmed.
The guest's information is as follows:
Guest id: 13
Rooms number: 4
Rooms Id: 24 25 26 27
Check in date: 10
Check out date: 11
Nights number: 1

day 3
Request has been made.
The guest's information is as follows:
Guest id: 14
Rooms number: 4
Rooms Id: 7 8 9 10
Check in date: 6
Check out date: 7
Nights number: 1

The available rooms number today is 26

day 3
Request has been confirmed.
The guest's information is as follows:
Guest id: 14
Rooms number: 4
Rooms Id: 7 8 9 10
Check in date: 6
Check out date: 7
Nights number: 1

day 3
Request has been made.
The guest's information is as follows:
```

```
Guest id: 15
Rooms number: 3
Rooms Id: 5 6 9
Check in date: 8
Check out date: 10
Nights number: 2

The available rooms number today is 26

day 3
Request has been confirmed.
The guest’s information is as follows:
Guest id: 15
Rooms number: 3
Rooms Id: 5 6 9
Check in date: 8
Check out date: 10
Nights number: 2

day 3
Request has been made.
The guest’s information is as follows:
Guest id: 11
Rooms number: 2
Rooms Id: 11 12
Check in date: 3
Check out date: 7
Nights number: 4

The available rooms number today is 24

day 3
Guest has checked in.
The guest’s information is as follows:
Guest id: 11
Rooms number: 2
Rooms Id: 11 12
Check in date: 3
Check out date: 7
Nights number: 4

day 4
Guest has checked in.
The guest's information is as follows:
Guest id: 1
Rooms number: 1
Rooms Id: 2
Check in date: 4
Check out date: 8
Nights number: 4

day 4
Request has been made.
The guest's information is as follows:
Guest id: 16
Rooms number: 3
Rooms Id: 16 17 18
Check in date: 6
Check out date: 9
Nights number: 3

The available rooms number today is 23

day 4
Request has been confirmed.
The guest's information is as follows:
Guest id: 16
Rooms number: 3
Rooms Id: 16 17 18
Check in date: 6
Check out date: 9
Nights number: 3
```

```
day 4
Request has been made.
The guest's information is as follows:
Guest id: 17
Rooms number: 1
Rooms Id: 10
Check in date: 7
Check out date: 10
Nights number: 3

The available rooms number today is 23

day 4
Request has been confirmed.
The guest's information is as follows:
Guest id: 17
Rooms number: 1
Rooms Id: 10
Check in date: 7
Check out date: 10
Nights number: 3

day 4
Request has been made.
The guest's information is as follows:
Guest id: 18
Rooms number: 2
Rooms Id: 19 20
Check in date: 5
Check out date: 9
Nights number: 4

The available rooms number today is 23

day 4
```

```
Request has been confirmed.
The guest's information is as follows:
Guest id: 18
Rooms number: 2
Rooms Id: 19 20
Check in date: 5
Check out date: 9
Nights number: 4

day 4
Request has been made.
The guest's information is as follows:
Guest id: 19
Rooms number: 2
Rooms Id: 11 24
Check in date: 8
Check out date: 10
Nights number: 2

The available rooms number today is 23

day 4
Request has been confirmed.
The guest's information is as follows:
Guest id: 19
Rooms number: 2
Rooms Id: 11 24
Check in date: 8
Check out date: 10
Nights number: 2

day 4
Request has been made.
The guest's information is as follows:
Guest id: 20
Rooms number: 1
Rooms Id: 3
Check in date: 7
Check out date: 8
Nights number: 1

The available rooms number today is 23

day 4
Request has been confirmed.
The guest's information is as follows:
Guest id: 20
Rooms number: 1
Rooms Id: 3
Check in date: 7
Check out date: 8
Nights number: 1

day 5
Guest has checked in.
The guest's information is as follows:
Guest id: 18
Rooms number: 2
Rooms Id: 19 20
Check in date: 5
Check out date: 9
Nights number: 4

day 5
Request has been made.
The guest's information is as follows:
Guest id: 21
Rooms number: 1
Rooms Id: 21
Check in date: 7
Check out date: 9
Nights number: 2
```

```
The available rooms number today is 21

day 5
Request has been confirmed.
The guest's information is as follows:
Guest id: 21
Rooms number: 1
Rooms Id: 21
Check in date: 7
Check out date: 9
Nights number: 2

day 5
Request has been made.
The guest's information is as follows:
Guest id: 22
Rooms number: 3
Rooms Id: 28 29 30
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 21

day 5
Request has been confirmed.
The guest's information is as follows:
Guest id: 22
Rooms number: 3
Rooms Id: 28 29 30
Check in date: 10
Check out date: 11
Nights number: 1

day 5
```

```
Request has been made.
The guest's information is as follows:
Guest id: 23
Rooms number: 2
Rooms Id: 13 14
Check in date: 6
Check out date: 8
Nights number: 2

The available rooms number today is 21

day 5
Request has been confirmed.
The guest's information is as follows:
Guest id: 23
Rooms number: 2
Rooms Id: 13 14
Check in date: 6
Check out date: 8
Nights number: 2

day 5
Request has been made.
The guest's information is as follows:
Guest id: 24
Rooms number: 3
Rooms Id: 22 23 25
Check in date: 8
Check out date: 9
Nights number: 1

The available rooms number today is 21

day 5
Request has been confirmed.
The guest's information is as follows:
```

```
Guest id: 24
Rooms number: 3
Rooms Id: 22 23 25
Check in date: 8
Check out date: 9
Nights number: 1

day 5
Request has been made.
The guest's information is as follows:
Guest id: 25
Rooms number: 3
Rooms Id: 26 27 28
Check in date: 7
Check out date: 9
Nights number: 2

The available rooms number today is 21

day 5
Request has been confirmed.
The guest's information is as follows:
Guest id: 25
Rooms number: 3
Rooms Id: 26 27 28
Check in date: 7
Check out date: 9
Nights number: 2

day 6
Guest has checked in.
The guest's information is as follows:
Guest id: 3
Rooms number: 1
Rooms Id: 1
Check in date: 6
Check out date: 8
Nights number: 2

day 6
Guest has checked in.
The guest’s information is as follows:
Guest id: 14
Rooms number: 4
Rooms Id: 7 8 9 10
Check in date: 6
Check out date: 7
Nights number: 1

day 6
Guest has checked in.
The guest’s information is as follows:
Guest id: 16
Rooms number: 3
Rooms Id: 16 17 18
Check in date: 6
Check out date: 9
Nights number: 3

day 6
Guest has checked in.
The guest’s information is as follows:
Guest id: 23
Rooms number: 2
Rooms Id: 13 14
Check in date: 6
Check out date: 8
Nights number: 2

day 6
Request has been discarded.
The guest’s information is as follows:
```

```
Guest id: 26
Rooms number: 3
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 6
Request has been discarded.
The guest's information is as follows:
Guest id: 27
Rooms number: 4
Rooms Id: No request granted.
Check in date: 8
Check out date: 11
Nights number: 3

day 6
Request has been made.
The guest's information is as follows:
Guest id: 28
Rooms number: 4
Rooms Id: 16 17 18 25
Check in date: 9
Check out date: 10
Nights number: 1

The available rooms number today is 11

day 6
Request has been confirmed.
The guest's information is as follows:
Guest id: 28
Rooms number: 4
Rooms Id: 16 17 18 25
Check in date: 9
Check out date: 10
Nights number: 1

day 6
Request has been made.
The guest's information is as follows:
Guest id: 30
Rooms number: 4
Rooms Id: 26 27 28 29
Check in date: 9
Check out date: 10
Nights number: 1

The available rooms number today is 11

day 6
Request has been confirmed.
The guest's information is as follows:
Guest id: 30
Rooms number: 4
Rooms Id: 26 27 28 29
Check in date: 9
Check out date: 10
Nights number: 1

day 6
Request has been discarded.
The guest's information is as follows:
Guest id: 29
Rooms number: 2
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 7
```

```
Guest has checked out.
The guest's information is as follows:
Guest id: 10
Rooms number: 4
Rooms Id: 3 4 5 6
Check in date: 3
Check out date: 7
Nights number: 4

The available rooms number today is 15

day 7
Guest has checked out.
The guest's information is as follows:
Guest id: 11
Rooms number: 2
Rooms Id: 11 12
Check in date: 3
Check out date: 7
Nights number: 4

The available rooms number today is 15

day 7
Guest has checked out.
The guest's information is as follows:
Guest id: 14
Rooms number: 4
Rooms Id: 7 8 9 10
Check in date: 6
Check out date: 7
Nights number: 1

The available rooms number today is 15

day 7
```

```
Guest has checked in.
The guest’s information is as follows:
Guest id: 17
Rooms number: 1
Rooms Id: 10
Check in date: 7
Check out date: 10
Nights number: 3

day 7
Guest has checked in.
The guest’s information is as follows:
Guest id: 20
Rooms number: 1
Rooms Id: 3
Check in date: 7
Check out date: 8
Nights number: 1

day 7
Guest has checked in.
The guest’s information is as follows:
Guest id: 21
Rooms number: 1
Rooms Id: 21
Check in date: 7
Check out date: 9
Nights number: 2

day 7
Guest has checked in.
The guest’s information is as follows:
Guest id: 25
Rooms number: 3
Rooms Id: 26 27 28
Check in date: 7
Check out date: 9
Nights number: 2

day 7
Request has been made.
The guest’s information is as follows:
Guest id: 31
Rooms number: 3
Rooms Id: 4 5 6
Check in date: 7
Check out date: 8
Nights number: 1

The available rooms number today is 12

day 7
Guest has checked in.
The guest’s information is as follows:
Guest id: 31
Rooms number: 3
Rooms Id: 4 5 6
Check in date: 7
Check out date: 8
Nights number: 1

day 7
Request has been discarded.
The guest’s information is as follows:
Guest id: 32
Rooms number: 4
Rooms Id: No request granted.
Check in date: 7
Check out date: 11
Nights number: 4

day 7
```

```
Request has been discarded.
The guest's information is as follows:
Guest id: 33
Rooms number: 2
Rooms Id: No request granted.
Check in date: 9
Check out date: 11
Nights number: 2

day 7
Request has been discarded.
The guest's information is as follows:
Guest id: 34
Rooms number: 5
Rooms Id: No request granted.
Check in date: 9
Check out date: 11
Nights number: 2

day 7
Request has been discarded.
The guest's information is as follows:
Guest id: 35
Rooms number: 5
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 8
Guest has checked out.
The guest's information is as follows:
Guest id: 1
Rooms number: 1
Rooms Id: 2
Check in date: 4
Check out date: 8
Nights number: 4

The available rooms number today is 2

day 8
Guest has checked out.
The guest’s information is as follows:
Guest id: 3
Rooms number: 1
Rooms Id: 1
Check in date: 6
Check out date: 8
Nights number: 2

The available rooms number today is 2

day 8
Guest has checked out.
The guest’s information is as follows:
Guest id: 23
Rooms number: 2
Rooms Id: 13 14
Check in date: 6
Check out date: 8
Nights number: 2

The available rooms number today is 2

day 8
Guest has checked out.
The guest’s information is as follows:
Guest id: 20
Rooms number: 1
Rooms Id: 3
Check in date: 7
Check out date: 8
Nights number: 1

The available rooms number today is 2

day 8
Guest has checked out.
The guest’s information is as follows:
Guest id: 31
Rooms number: 3
Rooms Id: 4 5 6
Check in date: 7
Check out date: 8
Nights number: 1

The available rooms number today is 2

day 8
Guest has checked in.
The guest’s information is as follows:
Guest id: 2
Rooms number: 4
Rooms Id: 1 2 3 4
Check in date: 8
Check out date: 11
Nights number: 3

day 8
Guest has checked in.
The guest’s information is as follows:
Guest id: 5
Rooms number: 2
Rooms Id: 7 8
Check in date: 8
Check out date: 11
Nights number: 3
```

```
day 8
Guest has checked in.
The guest's information is as follows:
Guest id: 8
Rooms number: 4
Rooms Id: 12 13 14 15
Check in date: 8
Check out date: 11
Nights number: 3

day 8
Guest has checked in.
The guest's information is as follows:
Guest id: 15
Rooms number: 3
Rooms Id: 5 6 9
Check in date: 8
Check out date: 10
Nights number: 2

day 8
Guest has checked in.
The guest's information is as follows:
Guest id: 19
Rooms number: 2
Rooms Id: 11 24
Check in date: 8
Check out date: 10
Nights number: 2

day 8
Guest has checked in.
The guest's information is as follows:
Guest id: 24
Rooms number: 3
Rooms Id: 22 23 25
Check in date: 8
Check out date: 9
Nights number: 1

day 8
Request has been discarded.
The guest's information is as follows:
Guest id: 36
Rooms number: 2
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 8
Request has been discarded.
The guest's information is as follows:
Guest id: 37
Rooms number: 3
Rooms Id: No request granted.
Check in date: 9
Check out date: 10
Nights number: 1

day 8
Request has been discarded.
The guest's information is as follows:
Guest id: 39
Rooms number: 2
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 8
```

```
Request has been discarded.
The guest’s information is as follows:
Guest id: 40
Rooms number: 2
Rooms Id: No request granted.
Check in date: 8
Check out date: 10
Nights number: 2

day 8
Request has been discarded.
The guest’s information is as follows:
Guest id: 38
Rooms number: 5
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 9
Guest has checked out.
The guest’s information is as follows:
Guest id: 18
Rooms number: 2
Rooms Id: 19 20
Check in date: 5
Check out date: 9
Nights number: 4

The available rooms number today is 1

day 9
Guest has checked out.
The guest’s information is as follows:
Guest id: 16
Rooms number: 3
Rooms Id: 16 17 18
Check in date: 6
Check out date: 9
Nights number: 3

The available rooms number today is 1

day 9
Guest has checked out.
The guest’s information is as follows:
Guest id: 21
Rooms number: 1
Rooms Id: 21
Check in date: 7
Check out date: 9
Nights number: 2

The available rooms number today is 1

day 9
Guest has checked out.
The guest’s information is as follows:
Guest id: 25
Rooms number: 3
Rooms Id: 26 27 28
Check in date: 7
Check out date: 9
Nights number: 2

The available rooms number today is 1

day 9
Guest has checked out.
The guest’s information is as follows:
Guest id: 24
Rooms number: 3
Rooms Id: 22 23 25
Check in date: 8
Check out date: 9
Nights number: 1

The available rooms number today is 1

day 9
Guest has checked in.
The guest's information is as follows:
Guest id: 12
Rooms number: 5
Rooms Id: 19 20 21 22 23
Check in date: 9
Check out date: 11
Nights number: 2

day 9
Guest has checked in.
The guest's information is as follows:
Guest id: 28
Rooms number: 4
Rooms Id: 16 17 18 25
Check in date: 9
Check out date: 10
Nights number: 1

day 9
Guest has checked in.
The guest's information is as follows:
Guest id: 30
Rooms number: 4
Rooms Id: 26 27 28 29
Check in date: 9
Check out date: 10
Nights number: 1
```

```
day 9
Request has been discarded.
The guest’s information is as follows:
Guest id: 41
Rooms number: 1
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 9
Request has been discarded.
The guest’s information is as follows:
Guest id: 42
Rooms number: 3
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 9
Request has been discarded.
The guest’s information is as follows:
Guest id: 43
Rooms number: 2
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 9
Request has been discarded.
The guest’s information is as follows:
Guest id: 44
Rooms number: 3
Rooms Id: No request granted.
Check in date: 9
Check out date: 11
Nights number: 2

day 9
Request has been discarded.
The guest's information is as follows:
Guest id: 45
Rooms number: 3
Rooms Id: No request granted.
Check in date: 9
Check out date: 11
Nights number: 2

day 10
Guest has checked out.
The guest's information is as follows:
Guest id: 17
Rooms number: 1
Rooms Id: 10
Check in date: 7
Check out date: 10
Nights number: 3

The available rooms number today is 0

day 10
Guest has checked out.
The guest's information is as follows:
Guest id: 15
Rooms number: 3
Rooms Id: 5 6 9
Check in date: 8
Check out date: 10
Nights number: 2
```

```
The available rooms number today is 0

day 10
Guest has checked out.
The guest's information is as follows:
Guest id: 19
Rooms number: 2
Rooms Id: 11 24
Check in date: 8
Check out date: 10
Nights number: 2

The available rooms number today is 0

day 10
Guest has checked out.
The guest's information is as follows:
Guest id: 28
Rooms number: 4
Rooms Id: 16 17 18 25
Check in date: 9
Check out date: 10
Nights number: 1

The available rooms number today is 0

day 10
Guest has checked out.
The guest's information is as follows:
Guest id: 30
Rooms number: 4
Rooms Id: 26 27 28 29
Check in date: 9
Check out date: 10
Nights number: 1
```

```
The available rooms number today is 0

day 10
Guest has checked in.
The guest's information is as follows:
Guest id: 4
Rooms number: 2
Rooms Id: 5 6
Check in date: 10
Check out date: 11
Nights number: 1

day 10
Guest has checked in.
The guest's information is as follows:
Guest id: 7
Rooms number: 3
Rooms Id: 9 10 11
Check in date: 10
Check out date: 11
Nights number: 1

day 10
Guest has checked in.
The guest's information is as follows:
Guest id: 9
Rooms number: 3
Rooms Id: 16 17 18
Check in date: 10
Check out date: 11
Nights number: 1

day 10
Guest has checked in.
The guest's information is as follows:
```

```
Guest id: 13
Rooms number: 4
Rooms Id: 24 25 26 27
Check in date: 10
Check out date: 11
Nights number: 1

day 10
Guest has checked in.
The guest's information is as follows:
Guest id: 22
Rooms number: 3
Rooms Id: 28 29 30
Check in date: 10
Check out date: 11
Nights number: 1

day 10
Request has been discarded.
The guest's information is as follows:
Guest id: 46
Rooms number: 1
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 10
Request has been discarded.
The guest's information is as follows:
Guest id: 48
Rooms number: 5
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1
```

```
day 10
Request has been discarded.
The guest's information is as follows:
Guest id: 49
Rooms number: 2
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 10
Request has been discarded.
The guest's information is as follows:
Guest id: 50
Rooms number: 4
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 10
Request has been discarded.
The guest's information is as follows:
Guest id: 47
Rooms number: 4
Rooms Id: No request granted.
Check in date: 10
Check out date: 11
Nights number: 1

day 11
Guest has checked out.
The guest's information is as follows:
Guest id: 2
Rooms number: 4
Rooms Id: 1 2 3 4
Check in date: 8
Check out date: 11
Nights number: 3

The available rooms number today is 30

day 11
Guest has checked out.
The guest's information is as follows:
Guest id: 5
Rooms number: 2
Rooms Id: 7 8
Check in date: 8
Check out date: 11
Nights number: 3

The available rooms number today is 30

day 11
Guest has checked out.
The guest's information is as follows:
Guest id: 8
Rooms number: 4
Rooms Id: 12 13 14 15
Check in date: 8
Check out date: 11
Nights number: 3

The available rooms number today is 30

day 11
Guest has checked out.
The guest's information is as follows:
Guest id: 12
Rooms number: 5
Rooms Id: 19 20 21 22 23
Check in date: 9
Check out date: 11
Nights number: 2

The available rooms number today is 30

day 11
Guest has checked out.
The guest's information is as follows:
Guest id: 4
Rooms number: 2
Rooms Id: 5 6
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 30

day 11
Guest has checked out.
The guest's information is as follows:
Guest id: 7
Rooms number: 3
Rooms Id: 9 10 11
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 30

day 11
Guest has checked out.
The guest's information is as follows:
Guest id: 9
Rooms number: 3
Rooms Id: 16 17 18
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 30

day 11
Guest has checked out.
The guest’s information is as follows:
Guest id: 13
Rooms number: 4
Rooms Id: 24 25 26 27
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 30

day 11
Guest has checked out.
The guest’s information is as follows:
Guest id: 22
Rooms number: 3
Rooms Id: 28 29 30
Check in date: 10
Check out date: 11
Nights number: 1

The available rooms number today is 30

This is hotel’s last 11 days summary:
Day 1
Rooms’ Occupancy Rate: 0
Granted Rate: 1
Day 2
```

```
Rooms' Occupancy Rate: 0.0666667
Granted Rate: 1
Day 3
Rooms' Occupancy Rate: 0.2
Granted Rate: 1
Day 4
Rooms' Occupancy Rate: 0.233333
Granted Rate: 1
Day 5
Rooms' Occupancy Rate: 0.3
Granted Rate: 1
Day 6
Rooms' Occupancy Rate: 0.633333
Granted Rate: 0.4
Day 7
Rooms' Occupancy Rate: 0.6
Granted Rate: 0.2
Day 8
Rooms' Occupancy Rate: 0.933333
Granted Rate: 0
Day 9
Rooms' Occupancy Rate: 0.966667
Granted Rate: 0
Day 10
Rooms' Occupancy Rate: 1
Granted Rate: 0
```

3 Code Snippets

3.1 guest.h

```
//
// Created by big on 2/28/19.
//

```

```
#ifndef NACHOS_GUEST_H
#define NACHOS_GUEST_H


#include <cstdlib>
#include <iostream>
#include <math.h>
#include <vector>

class guest {
private:
    std::size_t Id;
    std::size_t rooms_num;
    std::size_t check_in_date;
    std::size_t nights_num;
    std::size_t check_out_date;
    std::vector<size_t> rooms_Id;
public:
    size_t getId() const;

    void setId(size_t Id);

    size_t getRooms_num() const;

    void setRooms_num(size_t rooms_num);

    size_t getCheck_in_date() const;

    void setCheck_in_date(size_t check_in_date);

    size_t getNights_num() const;

    void setNights_num(size_t nights_num);

    size_t getCheck_out_date() const;

    void setCheck_out_date(size_t check_out_date);

    const std::vector<size_t> &getRooms_Id() const;

    void setRooms_Id(const std::vector<size_t> &rooms_Id);

    bool operator==(const guest &other) const;

};

int guest_check_in_compare(guest l, guest r);
int guest_check_out_compare(guest l,guest r);

#endif //NACHOS_GUEST_H
```

guest.h

3.2 guest.cc

```
//
// Created by big on 2/28/19.
//

#include ”guest.h”

size_t guest::getId() const {
    return Id;
}

void guest::setId(size_t Id) {
    guest::Id = Id;
}

size_t guest::getRooms_num() const {
    return rooms_num;
}

void guest::setRooms_num(size_t rooms_num) {
    guest::rooms_num = rooms_num;
}

size_t guest::getCheck_in_date() const {
    return check_in_date;
}

void guest::setCheck_in_date(size_t check_in_date) {
    guest::check_in_date = check_in_date;
}

```

```
size_t guest::getNights_num() const {
    return nights_num;
}

void guest::setNights_num(size_t nights_num) {
    guest::nights_num = nights_num;
}

size_t guest::getCheck_out_date() const {
    return check_out_date;
}

void guest::setCheck_out_date(size_t check_out_date) {
    guest::check_out_date = check_out_date;
}

const std::vector<size_t> &guest::getRooms_Id() const {
    return rooms_Id;
}

void guest::setRooms_Id(const std::vector<size_t> &rooms_Id) {
    guest::rooms_Id = rooms_Id;
}

bool guest::operator==(const guest &other) const {
    return this->Id == other.Id;
}

int guest_check_in_compare(guest l, guest r) {
    if (l.getCheck_in_date() < r.getCheck_in_date()) {
        return -1;
    }
    if (l.getCheck_in_date() > r.getCheck_in_date()) {
        return 1;
    } else {
        return 0;
    }
}

int guest_check_out_compare(guest l, guest r) {
    if (l.getCheck_out_date() < r.getCheck_out_date()) {
        return -1;
    }
    if (l.getCheck_out_date() > r.getCheck_out_date()) {
        return 1;
    } else {
        return 0;
    }
}
```

guest.cc

3.3 threadtest.cc

```
#include "kernel.h"
#include "main.h"
#include "thread.h"
#include <iostream>
#include <vector>
#include <stdio.h>
#include <math.h>
#include <map>
#include "guest.h"
#include "scheduler.h"
#include <stdlib.h>
#include <string>
#include <memory>
#include <float.h>

/*
void
SimpleThread(int which) {
    int num;

    for (num = 0; num < 5; num++) {
        printf("*** thread %d looped %d times\n", which, num);
        kernel->currentThread->Yield();
    }
}

void
ThreadTest()
{
    Thread *t = new Thread("forked thread");
    t->Fork((VoidFunctionPtr) SimpleThread, (void *) 1);

    SimpleThread(0);
}
*/




//---------------------------------------------------------------------------
//Some global variable
//---------------------------------------------------------------------------
std::size_t day = 11;
std::size_t today = 1;
vector<int> room_use(30); //check if room is available
map<int, vector<int>> room_available; //keep track of the rooms' availability
vector<double> granted_rate(day - 1); //granted rate;
vector<double> occupancy_rate(day - 1); //occupancy rate;
std::unique_ptr<SortedList<guest>> staying = std::unique_ptr<SortedList<guest>>(
        new SortedList<guest>(&guest_check_out_compare));
List<guest> checkout;
std::unique_ptr<SortedList<guest>> confirmed = std::unique_ptr<SortedList<guest>>(
        new SortedList<guest>(&guest_check_in_compare));
List<guest> discard;
map<int, Thread *> thread_map;
int guest_thread_count = 0;
double grant = 0.0;


//---------------------------------------------------------------------------
//Initialize the map
//---------------------------------------------------------------------------
void map_initialize() {
    for (int i = 1; i <= day; ++i) {
        room_available[i] = room_use;
    }
}

//---------------------------------------------------------------------------
//Sort the list
//---------------------------------------------------------------------------
/*template<typename ListType, typename SortedListType, typename PredType>
```

```
std::unique_ptr<SortedListType> rebuild_list_with_order(ListType &original, PredType &&order){
    std::unique_ptr<SortedListType> result(new SortedListType(order));
    auto iter = original.GetIterator();
    while(!iter.IsDone()){
        result->Insert(iter.Item());
        iter.Next();
    }
    return result;
}

void sort_confirm_list(){
    auto newlist = rebuild_list_with_order<SortedList<guest>, SortedList<guest>,
            int (*)(guest,guest)>(*confirmed, &guest_check_in_compare);
    *confirmed = *newlist;
}

void sort_stay_list(){
    auto newlist = rebuild_list_with_order<SortedList<guest>, SortedList<guest>,
            int (*)(guest,guest)>(*staying, &guest_check_out_compare);
    *staying = *newlist;
}
*/



//--------------------------------------------------------------------------
//Print function
//--------------------------------------------------------------------------
void print_request(guest req, const std::string &hint) {
    std::cout << "day " << today << std::endl;
    std::cout << hint << std::endl;
    std::cout << "The guest's information is as follows: " << std::endl;
    std::cout << "Guest id: " << req.getId() << std::endl;
    std::cout << "Rooms number: " << req.getRooms_num() << std::endl;
    if (req.getRooms_Id().empty()) {
        std::cout << "Rooms Id: No request granted." << std::endl;
    } else {
        std::cout << "Rooms Id: ";
        for (size_t i = 0; i < req.getRooms_Id().size(); ++i) {
            std::cout << req.getRooms_Id()[i] << " ";
        }
        std::cout << std::endl;
    }
    std::cout << "Check in date: " << req.getCheck_in_date() << std::endl;
    std::cout << "Check out date: " << req.getCheck_out_date() << std::endl;
    std::cout << "Nights number: " << req.getNights_num() << std::endl;
    std::cout << std::endl;
}

void print_room_num() {
    int count = 0;
    for (size_t i = 0; i < room_available[today].size(); ++i) {
        if (room_available[today][i] == 0) { count += 1; }
    }
    std::cout << "The available rooms number today is " << count << std::endl;
    std::cout << std::endl;
}

void print_summary() {
    std::cout << "This is hotel's last "<<day<<" days summary: " << std::endl;
    for (int i = 0; i < day - 1; ++i) {
        int tday = i + 1;
        std::cout << "Day " << tday << std::endl;
        std::cout << "Rooms' Occupancy Rate: " << occupancy_rate[i] << std::::
   endl;
        std::cout << "Granted Rate: " << granted_rate[i] << std::endl;
    }
}


//--------------------------------------------------------------------------
//daily summary
//--------------------------------------------------------------------------
void daily_work() {
    double granted = grant / 5;
    granted_rate[today - 1] = granted;

    double counts = 0;
    for (size_t i = 0; i < room_available[today].size(); ++i) {

        if (room_available[today][i] == 1) {
            counts += 1;
        }
    }
    double count_rate = counts / 30;
    occupancy_rate[today - 1] = count_rate;
}


//-----------------------------------------------------------------------------
//Thread function
//-----------------------------------------------------------------------------
void guest_thread(size_t id) {
    guest_thread_count += 1;
    size_t guest_id = id;
    guest new_req;
    new_req.setId(guest_id);
    size_t room_num = (rand() % 5) + 1;
    new_req.setRooms_num(room_num);
    size_t night_num = (rand() % 4) + 1;
    size_t restday = day - today;
    size_t check_in = (rand() % restday) + today;
    size_t check_out = check_in + night_num;
    if (check_out > day) {
        check_out = day;
        night_num = check_out - check_in;
    }
    new_req.setCheck_in_date(check_in);
    new_req.setCheck_out_date(check_out);
    new_req.setNights_num(night_num);
    //check room available numbers
    int total_count = 0;
    for (size_t i = 0; i < room_available[check_in].size(); ++i) {
        int count = 0;
        for (int j = check_in; j < check_out; ++j) {
            if (room_available[j][i] == 0) {
                count += 1;
            }
        }
        if (count == night_num) { total_count += 1; }
    }
    //check if room is available
    if (total_count >= new_req.getRooms_num()) {
        int c_total_count = 0;
        vector<size_t> room_id;
        for (size_t i = 0; i < room_available[check_in].size(); ++i) {
            int c_count = 0;
            for (int j = check_in; j < check_out; ++j) {
                if (room_available[j][i] == 0) {
                    c_count += 1;
                }
            }
            if (c_count == night_num) {
                for (int j = check_in; j < check_out; ++j) {
                    room_available[j][i] = 1;
                }
                c_total_count += 1;
                room_id.push_back(i + 1);
            }
            if (c_total_count == new_req.getRooms_num())break;
        }
        new_req.setRooms_Id(room_id);
        /*
        if (total_count >= new_req.getRooms_num()) {
            int c_count = 0;
            vector<size_t> room_id;
            for (size_t i = 1; i <= room_use.size(); ++i) {
                if (room_use[i] == 0) {
                    room_use[i] = 1;
                    c_count += 1;
                    room_id.push_back(i);
                }
                if (c_count == new_req.getRooms_num())break;
            }
            new_req.setRooms_Id(room_id);
            */


        //count granted numbers
        grant += 1;
        print_request(new_req, "Request has been made.");
        print_room_num();
        if (new_req.getCheck_in_date() == today) {
            staying->Insert(new_req);
            print_request(new_req, "Guest has checked in.");
            thread_map[id] = kernel->currentThread;
            kernel->interrupt->SetLevel(IntOff);
            kernel->currentThread->Sleep(FALSE);
        } else {
            confirmed->Insert(new_req);
            print_request(new_req, "Request has been confirmed.");
            thread_map[id] = kernel->currentThread;
            kernel->interrupt->SetLevel(IntOff);
            kernel->currentThread->Sleep(FALSE);
        }
    } else {
        discard.Append(new_req);
        print_request(new_req, "Request has been discarded.");
        kernel->currentThread->Finish();
    }

    //after woken up
    //woken up by check in
    if (new_req.getCheck_in_date() == today) {
        staying->Insert(new_req);
        confirmed->Remove(new_req);
        print_request(new_req, "Guest has checked in.");
        kernel->interrupt->SetLevel(IntOff);
        kernel->currentThread->Sleep(FALSE);
    }

    //woken up by check out
    if (new_req.getCheck_out_date() == today) {
        checkout.Append(new_req);
        staying->Remove(new_req);
        vector<size_t> room_id;
        room_id = new_req.getRooms_Id();
        for (size_t i = 0; i < room_id.size(); ++i) {
            size_t it = room_id[i] - 1;
            for (int j = check_in; j < check_out; ++j) {
                room_available[j][it] = 0;
            }
        }
        /*
        for (size_t i = 0; i < room_id.size(); ++i) {
            size_t it = room_id[i];
            room_use[it] = 0;
        }
        */
        print_request(new_req, "Guest has checked out.");
        print_room_num();
        kernel->currentThread->Finish();
    }
    /*
    auto iter = confirm.GetIterator();
    auto iter2 = stay.GetIterator();
    while (!iter.IsDone()) {
        if (iter.Item().getId() == id) {
            stay.Append(iter.Item());
            confirm.Remove(iter.Item());
            print_request(iter.Item(), "Guest has checked in.");
            break;
        }
        iter.Next();
    }
    kernel->currentThread->Sleep(FALSE);

    while (!iter2.IsDone()) {
        if (iter2.Item().getId() == id) {
            checkout.Append(iter2.Item());
            stay.Remove(iter2.Item());
            vector<size_t> room_id;
            room_id = iter2.Item().getRooms_Id();
            for (size_t i = 0; i < room_id.size(); ++i) {
                size_t it = room_id[i];
                room_use[it] = 0;
            }

            print_request(iter2.Item(), "Guest has checked out.");
            print_room_num();
            break;
        }
        iter.Next();
    }
    kernel->currentThread->Finish();
     */

}


void concierge_thread(size_t num) {
  //set a while loop to simulate all 11 days.
    while (today <= day) {
        if (today > 1) {
      //call all qualified guest threads to check out
```

```
            if(!staying->IsEmpty()) {
                auto iter2 = staying->GetIterator();
                if (iter2.Item().getCheck_out_date() == today) {
                    while (iter2.Item().getCheck_out_date() == today) {
                        auto th_id2 = iter2.Item().getId();
                        kernel->interrupt->SetLevel(IntOff);
                        kernel->scheduler->ReadyToRun(thread_map[th_id2]);
                        iter2.Next();
                        if(iter2.IsDone()) break;
                    }
                }
            }
      //call all qualified guest threads to check in
            if(!confirmed->IsEmpty()) {
                auto iter = confirmed->GetIterator();
                if (iter.Item().getCheck_in_date() == today) {
                    while (iter.Item().getCheck_in_date() == today) {
                        auto th_id = iter.Item().getId();
                        kernel->interrupt->SetLevel(IntOff);
                        kernel->scheduler->ReadyToRun(thread_map[th_id]);
                        iter.Next();
                        if(iter.IsDone()) break;
                    }
                }
            }



            /*
            while (!iter2.IsDone()) {
                if (iter2.Item().getCheck_out_date() == today) {
                    auto th_id2 = iter2.Item().getId();
                    kernel->interrupt->SetLevel(IntOff);
                    kernel->scheduler->ReadyToRun(thread_map[th_id2]);
                }
                iter2.Next();
            }

            while (!iter.IsDone()) {
                if (iter.Item().getCheck_in_date() == today) {
                    auto th_id = iter.Item().getId();
                    kernel->interrupt->SetLevel(IntOff);
                    kernel->scheduler->ReadyToRun(thread_map[th_id]);
                }
                iter.Next();
            }
             */

        }
    //create 5 new guest threads
        if (today != day) {
            for (size_t i = 0; i < num; ++i) {
                size_t Id = 5 * (today - 1) + i + 1;
                Thread *gt = new Thread("guest thread");
                gt->Fork((VoidFunctionPtr) guest_thread, (void *) Id);
            }
        }
    //make sure all five guest threads finished or slept
        if (today != day) {
            while (1) {
                if (guest_thread_count != 5) {
                    kernel->currentThread->Yield();
                } else { break; }
            }
        } else { kernel->currentThread->Yield(); }
        daily_work();
        guest_thread_count = 0;
        grant = 0;
        today = today + 1;
    }
    print_summary();
}


void
ThreadTest() {
    map_initialize();
    Thread *ct = new Thread("Concierge thread");
    ct->Fork((VoidFunctionPtr) concierge_thread, (void *) 5);
}
```

threadtest.cc