ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ "ВЫСШАЯ ШКОЛА ЭКОНОМИКИ"»

МИКРОПРОЕКТ №2

Работа студента 2 курса бакалавриата группы БПИ-195 по предмету «Архитектура вычислительных систем»

Выполнил:

Карабаш Радимир Сейранович,

БПИ195-1

Преподаватель:

Доктор технических наук,

Профессор

Легалов А. И

Задание:

13. Задача о гостинице. В гостинице 30 номеров, клиенты гостиницы снимают номер на одну ночь, если в гостинице нет свободных номеров, клиенты устраиваются на ночлег рядом с гостиницей и ждут, пока любой номер не освободится. Создать многопоточное приложение, моделирующее работу гостиницы.

Решение:

Для решения задачи необходимо корректно описать взаимодействия клиентов с отелем, и поэтому было принято решение, что для реализации приложения требуется использовать условные переменные.

При запуске программы пользователь видит приветствуещее сообщение с информацией об исполнителе и задании, а затем программа просит ввести пользователя количество итераций для рассмотрения работы отеля. В случае передачи некорректных данных программа выдаст соответствующее информационное сообщение, где будет описано какие данные требуются программе на вход. Одна итерация равна одному дню работы отеля, через который ежедневно проходит 300 гостей.

После получения количеств итерации программа инициализирует поток-отель, который параллельно выполняет метод **enter_hotel**, в котором прописано логика работы отеля. Затем программа инициалилизирует 300 потоков гостей, выполняющие метод **try_rent**.

Взаимодействие отеля и клиента происходит следующим образом. Клиент пытается заселиться в отель при условии, что там есть свободные номера. Если же места есть, отель принимает сигнал от условной переменной и заселяет гостя в номер, иначе, пока мест нет, отель ожидает сигнал о свободных номерах, а гости засыпают и ждут своей очереди снаружи отеля.

Вывод данных осуществляется в консоль.

Входным параметром для количества итераций является целое число в диапазоне от 1 до 100 включительно.

Описание методов:

- void log(string message) логирование и вывод в консоль взаимодействия гостей с отелем
- **void enter_hotel**() метод, обратывающий логику отеля. Если место в отеле есть, то успешно заселяет гостя в номер.
- **void try_rent(int id)** метод, обратывающий логику гостя с идентификатором id. Если место в отеле есть, то успешно заселяет гостя в номер, иначе встает в очередь на заселение.

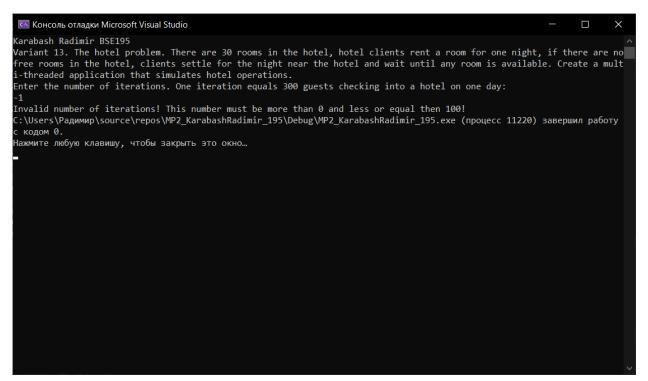
Текст программы приведен ниже:

```
#include <iostream>
      #include <thread>
      #include <mutex>
      #include <vector>
      #include <string>
      #include <fstream>
      #include <condition variable>
      #include <ctime>
     using namespace std;
     mutex check, rent, print;
     bool is notified = false;
     condition variable avaibility;
     int guest id, pr time, num guests = 0, ids = 1;
     // Логирование происходящего
     void log(string message) {
         print.lock();
         cout << "[Time " + to string((clock() - pr time) / CLOCKS PER SEC) +</pre>
"] " + message << endl;
         print.unlock();
      }
     // Метод приема гостей отеля
     void enter hotel() {
         while (true)
              unique lock<mutex> locker(check);
              while (!is notified)
                  avaibility.wait(locker);
              log("(+) Hotel meets Guest #" + to_string(guest_id));
              num guests++;
```

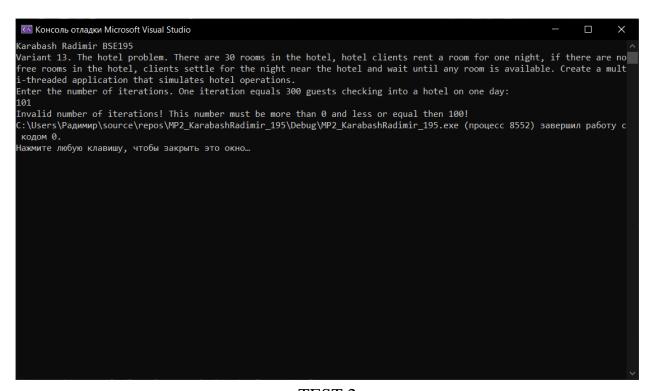
```
is notified = false;
         }
      }
      // Попытка гостя заселиться в отель
     void try rent(int id) {
          {
              unique lock<mutex> lock(rent);
              log("(?) Guest #" + to string(id) + " is trying to rent room...
");
              this thread::sleep for(chrono::milliseconds(100));
             bool flag = true;
              while (num guests \geq 30)
                  if (flag)
                  {
                      log("(!) Oh no! Hotel has no free rooms! Guest #" +
to string(id) + " must waiting outside... ");
                     flag = false;
                  }
              }
              guest id = id;
              avaibility.notify one();
              is notified = true;
          }
          {
              unique lock<mutex> lock();
              srand(time(nullptr));
              this thread::sleep for(chrono::milliseconds((rand() % 10 + 1) *
1000));
              log("(-) Guest #" + to string(id) + " leave hotel... ");
              num guests--;
          }
     int main()
      {
          cout << "Karabash Radimir BSE195" << endl;</pre>
          cout << "Variant 13. The hotel problem. There are 30 rooms in the
hotel, hotel clients rent a room for one night, if there are nofree rooms in
the hotel, clients settle for the night near the hotel and wait until any
room is available. Create a multi-threaded application that simulates hotel
operations." << endl;
```

```
cout << "Enter the number of iterations. One iteration equals 300
guests checking into a hotel on one day:" << endl;</pre>
          int n, k = 0;
          cin >> n;
          if (n \le 0 \mid \mid n > 100)
              cout << "Invalid number of iterations! This number must be more</pre>
than 0 and less or equal then 100!";
              return 0;
          }
          pr_time = clock();
          thread hotel (enter hotel);
          vector<thread> threads;
          while (k < n)
          {
              threads.clear();
              log("Day #" + to_string(k + 1) + " started...");
              for (size_t i = 0; i < 300; i++)
              {
                  threads.push_back(thread(try_rent, ids++));
              for (size t i = 0; i < threads.size(); i++)</pre>
                  threads[i].join();
              }
              k++;
              log("Day #" + to string(k) + " came to an end...");
          hotel.detach();
          cout << "The hotel served " + to_string(--ids) + " clients!" <<</pre>
endl;
         return 0;
      }
```

Результаты разработки:



TEST 1



TEST 2

TEST 3.1

```
[Time 51] (-) Guest #248 leave hotel...
[Time 51] (-) Guest #248 leave hotel...
[Time 53] (-) Guest #281 leave hotel...
[Time 53] (-) Guest #280 leave hotel...
[Time 53] (-) Guest #280 leave hotel...
[Time 53] (-) Guest #283 leave hotel...
[Time 53] (-) Guest #283 leave hotel...
[Time 53] (-) Guest #284 leave hotel...
[Time 53] (-) Guest #285 leave hotel...
[Time 53] (-) Guest #287 leave hotel...
[Time 53] (-) Guest #287 leave hotel...
[Time 53] (-) Guest #287 leave hotel...
[Time 53] (-) Guest #289 leave hotel...
[Time 53] (-) Guest #289 leave hotel...
[Time 54] (-) Guest #289 leave hotel...
[Time 57] (-) Guest #289 leave hotel...
[Time 57] (-) Guest #289 leave hotel...
[Time 57] (-) Guest #291 leave hotel...
[Time 57] (-) Guest #291 leave hotel...
[Time 57] (-) Guest #294 leave hotel...
[Time 57] (-) Guest #295 leave hotel...
[Time 57] (-) Guest #295 leave hotel...
[Time 57] (-) Guest #297 leave hotel...
[Time 57] (-) Guest #298 leave hotel...
[Time 58] (-) Guest #298 leave hotel...
```

TEST 3.2

```
© C\Users\Paдимир\source\repos\MP2_KarabashRadimir_195\Debug\MP2_KarabashRadimir_195.exe — Х Karabash Radimir BSE195

Variant 13. The hotel problem. There are 30 rooms in the hotel, hotel clients rent a room for one night, if there are no free rooms in the hotel, clients settle for the night near the hotel and wait until any room is available. Create a mult i-threaded application that simulates hotel operations. Enter the number of iterations. One iteration equals 300 guests checking into a hotel on one day:

5

C\Users\Paquiman \text{ArabashRadimir_195\Lebug\MP2_KarabashRadimir_195\Lebug\RPA \text{Papur} \text{ArabashRadimir_195\Lebug\RPA \text{Papur} \text{Papur} \text{ArabashRadimir_195\Lebug\RPA \text{Papur} \text{Papur
```

TEST 4.1

```
 Консоль отладки Microsoft Visual Studio
                                                                                                                                              [Time 266] (-) Guest #1479 leave hotel...
[Time 267] (-) Guest #1467 leave hotel...
[Time 267] (-) Guest #1451 leave hotel...
[Time 267]
[Time 267]
[Time 267]
             (-) Guest #1452 leave hotel...
Time 267]
             (-) Guest #1453 leave hotel...
             (-) Guest #1454 leave hotel...
[Time 267]
             (-) Guest #1471 leave hotel...
(-) Guest #1436 leave hotel...
[Time 267]
 Time 268]
[Time 268]
             (-) Guest #1457 leave hotel...
 Time 268]
             (-) Guest #1458 leave hotel...
[Time 268]
             (-) Guest #1440 leave hotel...
 Time 268]
             (-) Guest #1499 leave hotel...
[Time 268]
             (-) Guest #1500 leave hotel...
[Time 269]
             (-) Guest #1480 leave hotel...
[Time 269]
             (-) Guest #1481 leave hotel...
 Time 269]
                  Guest #1469 leave hotel...
[Time 269]
             (-) Guest #1483 leave hotel...
Time 269]
             (-) Guest #1472 leave hotel...
[Time 270] (-) Guest #1485 leave hotel...
 Time 270]
                  Guest #1486 leave hotel...
[Time 270] (-) Guest #1487 leave hotel...
[Time 270] (-) Guest #1496 leave hotel...
[Time 270] (-) Guest #1489 leave hotel...
[Time 270] Day #5 came to an end...
The hotel served 1500 clients!
C:\Users\Радимир\source\repos\MP2_KarabashRadimir_195\Debug\MP2_KarabashRadimir_195.exe (процесс 14440) завершил работу
с кодом 0.
 Нажмите любую клавишу, чтобы закрыть это окно…
```

TEST 4.2

Список использованных источников

• Потоки, блокировки и условные переменные в C++11 [Часть 2] [Электронный ресурс] / Хабр –URL: https://habr.com/ru/post/182626/. (Дата обращения: 09.12.2020).