

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

**ОТЧЁТ ПО МИКРОПРОЕКТУ № 2
ПО ДИСЦИПЛИНЕ
«АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ»
ВАРИАНТ 15**

Исполнитель
студент группы БПИ191
Н. К. Игумнов

Москва 2020

ЗАДАНИЕ

15. *Задача о гостинице - 3 (дамы и джентльмены)*. В гостинице 10 номеров рассчитаны на одного человека и 15 номеров рассчитаны на двух человек. В гостиницу приходят клиенты дамы и клиенты джентльмены, и конечно они могут провести ночь в номере только с представителем своего пола. Если для клиента не находится подходящего номера, он уходит искать ночлег в другое место. Создать многопоточное приложение, моделирующее работу гостиницы.

ИНТЕРПРЕТАЦИЯ

Вводится число *thread_num* – число клиентов. Затем генерируются *thread_num* случайных клиентов (дам или джентельменов), для каждого из которых выделен свой поток. Изначально заполняются двухместные номера, затем одноместные.

РЕШЕНИЕ

В данной задаче я применяю модель **взаимодействующих равных**.
Ниже представлены функции в программе:

Read()

```
bool Read(int *thread_number, std::string *file_name) { // NOLINT
#ifdef GENERATE
    std::mt19937 gen(std::chrono::high_resolution_clock::now().time_since_epoch().count());
    std::uniform_int_distribution<int64_t> dist_t(1, MAX_VALUE - MIN_VALUE + 1);
    *thread_number = dist_t(gen);
    *file_name = "answer.txt";
    std::cout << "Amount of people: " << *thread_number << '\n';
#else
    std::cout << "Input amount of people (in [1, 10000]): ";
    std::cin >> *thread_number;
    if (*thread_number < 1 || *thread_number > 10000) {
        std::cout << "Incorrect amount of people!\n";
        return false;
    }
#endif
#ifdef FILE_OUT
    std::cout << "Input name of output file: ";
    std::cin >> *file_name;
    std::ofstream out(*file_name);
    if (!out.is_open()) {
        std::cout << "Incorrect file name!\n";
        out.close();
        return false;
    }
    out.close();
#endif
    rk_sema_init(&write_semaphore, value: 1);
    return true;
}
```

Рисунок 1 – функция Read

Функция Read() считывает входные данные.

Если включён GENERATE (строка 23 раскомменчена), то число потоков генерируется случайно, а выходной файл задаётся по умолчанию (answer.txt).

Иначе пользователь вводит *thread_num* – число потоков (в [1, 1000]). Если число не попадает в указанные границы, то пользователю об этом сообщается. Если включён FILE_OUT, то пользователь также вводит название выходного файла.

Далее инициализируется семафор write_semaphore.

Compute()

```
int Compute(std::chrono::time_point<std::chrono::steady_clock> begin, int thread_number,
            std::vector<Customer> &single_rooms, std::vector<std::pair<Customer, Customer>> &double_rooms) {
    std::mt19937 gen(std::chrono::high_resolution_clock::now().time_since_epoch().count());
    std::uniform_int_distribution<int> dist_customer(a: 0, b: 1);
    int id = 0, could_not = 0;
    std::vector<std::thread> threads(thread_number);
    for (int i = 0; i < thread_number; ++i) {
        Customer customer = dist_customer( & gen) == 0 ? Customer::MAN : Customer::WOMAN;
        threads[i] = std::thread{SpendTheNight, begin, customer, id++, std::ref( & could_not),
                                std::ref( & single_rooms), std::ref( & double_rooms)};
        std::this_thread::sleep_for( d: std::chrono::milliseconds(100));
    }
    for (std::thread &thr : threads) {
        thr.join();
    }
    return could_not;
}
```

Рисунок 2 - функция Compute

Функция Compute() находит ответ на задачу, генерируя случайного человека и вызывая *thread_number* потоков, отвечающих за них. Каждый из потоков описывается функцией *SpendTheNight()*. Возвращает функция количество человек, которым не хватило места в отеле.

SpendTheNight()

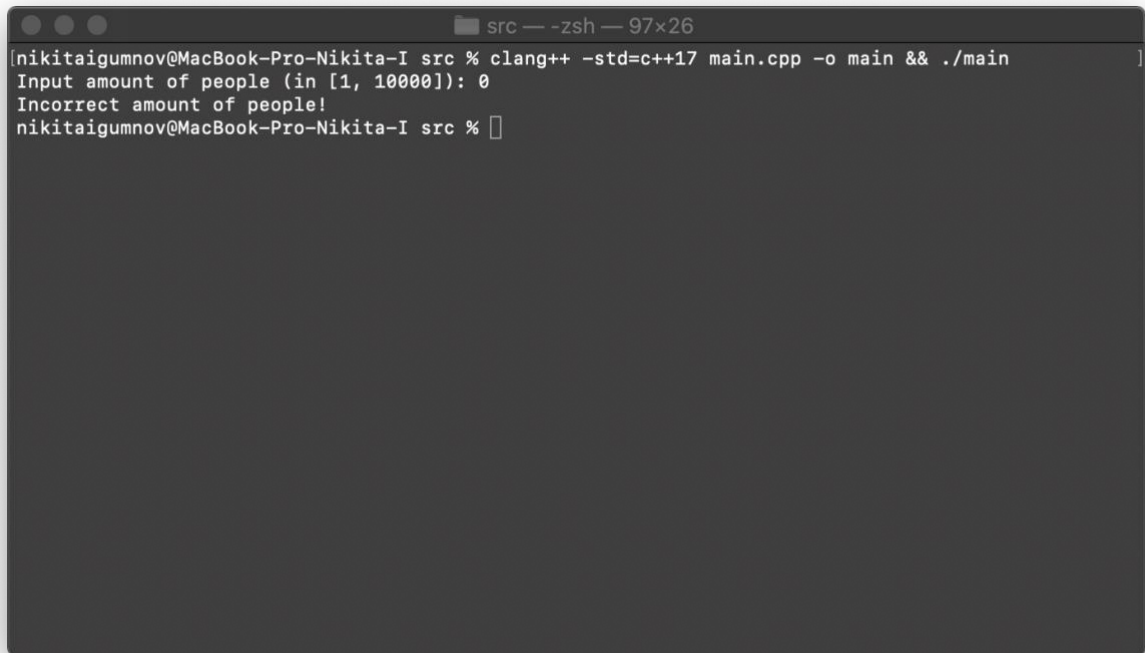
Функция *SpendTheNight()* обрабатывает каждого клиента. Выводится информация о том, что человек встал в очередь (*{name} #{id} came to your hotel*) и информация после обработки человека (либо, что он зарегистрировался в отеле (*{name} #{id} checked in to the {single / double} room #{id_room}, thread: {cur_thread.get_id()}*), либо, что все номера в отеле были заняты (*{name} #{id} could not find a room, thread: {cur_thread.get_id()})*).

Print()

Функция *Print()* выводит в консоль / выходной файл (в зависимости от *FILE_OUT*) полученное расположение людей в отеле и количество людей, которые не получили номер.

ТЕСТИРОВАНИЕ ПРОГРАММЫ

1. Некорректный ввод (*thread_number* = 0)



```
src — zsh — 97x26
nikitaigumnov@MacBook-Pro-Nikita-I src % clang++ -std=c++17 main.cpp -o main && ./main
Input amount of people (in [1, 10000]): 0
Incorrect amount of people!
nikitaigumnov@MacBook-Pro-Nikita-I src %
```

Программа отработала успешно.

2. *thread_number* = 5

```

src — zsh — 110x32
nikitaigumnov@MacBook-Pro-Nikita-I src % clang++ -std=c++17 main.cpp -o main && ./main
Input amount of people (in [1, 10000]): 5

Single rooms:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  |  |  |  |  |  |  |  |  |  |

Double rooms:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Number of people that could not find rooms: 0
Time: 152 ms, Gentleman #0 came to your hotel
Time: 307 ms, Gentleman #1 came to your hotel
Time: 307 ms, Gentleman #0 checked in to the double room #0, thread: 0x700003f61000
Time: 458 ms, Lady #2 came to your hotel
Time: 458 ms, Gentleman #1 checked in to the double room #0, thread: 0x700003fe4000
Time: 612 ms, Lady #3 came to your hotel
Time: 766 ms, Gentleman #4 came to your hotel
Time: 767 ms, Lady #2 checked in to the double room #1, thread: 0x700004067000
Time: 767 ms, Lady #3 checked in to the double room #1, thread: 0x7000040ea000
Time: 767 ms, Gentleman #4 checked in to the double room #2, thread: 0x70000416d000

Single rooms:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  |  |  |  |  |  |  |  |  |  |

Double rooms:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| M M | W W | M |  |  |  |  |  |  |  |  |  |  |  |

Number of people that could not find rooms: 0
Time (spend): 0.767 sec
nikitaigumnov@MacBook-Pro-Nikita-I src %

```

Программа отработала успешно.

3. *thread_number* = 100 (см. tests/answer1.txt)

```

src — zsh — 97x26

Double rooms:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Number of people that could not find rooms: 0
Time: 154 ms, Lady #0 came to your hotel
Time: 305 ms, Lady #1 came to your hotel
Time: 305 ms, Lady #0 checked in to the double room #0, thread: 0x70000ad7e000
Time: 456 ms, Lady #2 came to your hotel
Time: 456 ms, Lady #1 checked in to the double room #0, thread: 0x70000ae01000
Time: 607 ms, Lady #3 came to your hotel
Time: 761 ms, Lady #4 came to your hotel
Time: 761 ms, Lady #2 checked in to the double room #1, thread: 0x70000ae84000
Time: 914 ms, Gentleman #5 came to your hotel
Time: 914 ms, Lady #3 checked in to the double room #1, thread: 0x70000af07000
Time: 1068 ms, Gentleman #6 came to your hotel
Time: 1222 ms, Gentleman #7 came to your hotel
Time: 1222 ms, Lady #4 checked in to the double room #2, thread: 0x70000af8a000
Time: 1376 ms, Gentleman #8 came to your hotel
Time: 1376 ms, Gentleman #5 checked in to the double room #3, thread: 0x70000b00d000
Time: 1530 ms, Lady #9 came to your hotel
Time: 1681 ms, Lady #10 came to your hotel
Time: 1681 ms, Gentleman #6 checked in to the double room #3, thread: 0x70000b090000
Time: 1834 ms, Lady #11 came to your hotel
Time: 1835 ms, Gentleman #7 checked in to the double room #4, thread: 0x70000b113000

```

Программа отработала успешно.

4. *thread_number* = 10 (см. tests/answer2.txt)

```
src — -zsh — 97x26
[nikitaigumnov@MacBook-Pro-Nikita-I src % clang++ -std=c++17 main.cpp -o main && ./main
Input amount of people (in [1, 10000]): 10
Input name of output file: answer2.txt

Single rooms:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | | | | | | | | | |

Double rooms:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| | | | | | | | | | | | | | |

Number of people that could not find rooms: 0
Time: 154 ms, Gentleman #0 came to your hotel
Time: 306 ms, Lady #1 came to your hotel
Time: 306 ms, Gentleman #0 checked in to the double room #0, thread: 0x70000b43d000
Time: 460 ms, Gentleman #2 came to your hotel
Time: 610 ms, Lady #3 came to your hotel
Time: 610 ms, Lady #1 checked in to the double room #1, thread: 0x70000b4c0000
Time: 765 ms, Gentleman #4 came to your hotel
Time: 765 ms, Gentleman #2 checked in to the double room #0, thread: 0x70000b543000
Time: 915 ms, Lady #5 came to your hotel
Time: 915 ms, Lady #3 checked in to the double room #1, thread: 0x70000b5c6000
Time: 1070 ms, Gentleman #6 came to your hotel
Time: 1225 ms, Lady #7 came to your hotel
Time: 1225 ms, Gentleman #4 checked in to the double room #2, thread: 0x70000b649000
Time: 1375 ms, Lady #8 came to your hotel
```

```
src — -zsh — 97x26
Time: 610 ms, Lady #1 checked in to the double room #1, thread: 0x70000b4c0000
Time: 765 ms, Gentleman #4 came to your hotel
Time: 765 ms, Gentleman #2 checked in to the double room #0, thread: 0x70000b543000
Time: 915 ms, Lady #5 came to your hotel
Time: 915 ms, Lady #3 checked in to the double room #1, thread: 0x70000b5c6000
Time: 1070 ms, Gentleman #6 came to your hotel
Time: 1225 ms, Lady #7 came to your hotel
Time: 1225 ms, Gentleman #4 checked in to the double room #2, thread: 0x70000b649000
Time: 1375 ms, Lady #8 came to your hotel
Time: 1375 ms, Lady #5 checked in to the double room #3, thread: 0x70000b6cc000
Time: 1530 ms, Lady #9 came to your hotel
Time: 1530 ms, Gentleman #6 checked in to the double room #2, thread: 0x70000b74f000
Time: 1530 ms, Lady #7 checked in to the double room #3, thread: 0x70000b7d2000
Time: 1530 ms, Lady #8 checked in to the double room #4, thread: 0x70000b855000
Time: 1530 ms, Lady #9 checked in to the double room #4, thread: 0x70000b8d8000

Single rooms:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| | | | | | | | | | |

Double rooms:
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| M M | W W | M M | W W | W W | | | | | | | | |

Number of people that could not find rooms: 0
Time (spend): 1.53 sec
nikitaigumnov@MacBook-Pro-Nikita-I src %
```

Программа отработала успешно

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Многопоточное программирование. Взаимодействие потоков. Микропроект. Требования к оформлению. 2020-2021 уч.г. [Электронный ресурс]. Режим доступа: <http://softcraft.ru/edu/comparch/tasks/mp02/>, свободный. (дата обращения: 13.12.2020)
- 2) `std::this_thread::get_id` [Электронный ресурс]. Режим доступа: https://en.cppreference.com/w/cpp/thread/get_id, свободный. (дата обращения: 13.12.2020)
- 3) Top 20 C++ multithreading mistakes and how to avoid them [Электронный ресурс]. Режим доступа: <https://www.acodersjourney.com/top-20-cplusplus-multithreading-mistakes/>, свободный. (дата обращения: 13.12.2020)
- 4) Invalid operands to binary expression ('const char*' and 'const char*') [Электронный ресурс]. Режим доступа: <https://stackoverflow.com/questions/49724851/invalid-operands-to-binary-expression-const-char-and-const-char>, свободный. (дата обращения: 13.12.2020)
- 5) Creating threads & joining them in a for loop [Электронный ресурс]. Режим доступа: <https://stackoverflow.com/questions/58996275/c-creating-threads-joining-them-in-a-for-loop>, свободный. (дата обращения: 13.12.2020)
- 6) Why joining all threads in one loop is different from starting and joining one thread at a time? [Электронный ресурс]. Режим доступа: <https://stackoverflow.com/questions/57192844/why-joining-all-threads-in-one-loop-is-different-from-starting-and-joining-one-t>, свободный. (дата обращения: 13.12.2020)
- 7) Такие удивительные семафоры [Электронный ресурс]. Режим доступа: <https://stackoverflow.com/questions/57192844/why-joining-all-threads-in-one-loop-is-different-from-starting-and-joining-one-t>, свободный. (дата обращения: 13.12.2020)
- 8) Добро пожаловать в параллельный мир [Электронный ресурс]. Режим доступа: <http://scrutator.me/post/2012/04/04/parallel-world-p1.aspx>, свободный. (дата обращения: 14.12.2020)
- 9) Семафоры: введение [Электронный ресурс]. Режим доступа: https://learn.c.info/c/pthreads_semaphores.html, свободный. (дата обращения: 13.12.2020)
- 10) Потоки, блокировки и условные переменные в C++11 [Электронный ресурс]. Режим доступа: <https://habr.com/ru/post/182610/>, свободный. (дата обращения: 13.12.2020)
- 11) Why are `sem_init()`, `sem_getvalue()`, `sem_destroy()` deprecated on Mac OS X - and what replaces them? [Электронный ресурс]. Режим доступа: <https://stackoverflow.com/questions/27736618/why-are-sem-init-sem-getvalue-sem-destroy-deprecated-on-mac-os-x-and-w>, свободный. (дата обращения: 13.12.2020)
- 12) Ошибки многопоточности в C++ [Электронный ресурс]. Режим доступа: <https://otus.ru/nest/post/145/>, свободный. (дата обращения: 17.11.2020)
- 13) Simple example of threading in C++ [Электронный ресурс]. Режим доступа: <https://stackoverflow.com/questions/266168/simple-example-of-threading-in-c>, свободный. (дата обращения: 17.11.2020)
- 14) `std::thread::hardware_concurrency` [Электронный ресурс]. Режим доступа: https://en.cppreference.com/w/cpp/thread/thread/hardware_concurrency, свободный. (дата обращения: 17.11.2020)

- 15) Measuring execution time of a fuction in C++ [Электронный ресурс]. Режим доступа: <https://stackoverflow.com/questions/22387586/measuring-execution-time-of-a-function-in-c>, свободный. (дата обращения: 17.11.2020)
- 16) Практические приёмы построения многопоточных приложений [Электронный ресурс]. Режим доступа: <http://www.softcraft.ru/edu/comparch/tasks/t03/>, свободный. (дата обращения: 17.11.2020)
- 17) Choosing the number of threads at runtime [Электронный ресурс]. Режим доступа: <https://livebook.manning.com/book/c-plus-plus-concurrency-in-action/chapter-2/92/>, свободный. (дата обращения: 17.11.2020)