**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 – число клиентов. Затем генерируются 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()

Изображение выглядит как текст

Автоматически созданное описание

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 – число потоков (в ). 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()

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 2 - функция Compute

Функция Compute() находит ответ на задачу, генерируя случайного человека и вызывая 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 ()

Изображение выглядит как текст

Автоматически созданное описание

Программа отработала успешно.

2.

Изображение выглядит как текст

Автоматически созданное описание

Программа отработала успешно.

3. (см. tests/answer1.txt)

Изображение выглядит как текст

Автоматически созданное описаниеПрограмма отработала успешно.

4. (см. tests/answer2.txt)

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описаниеПрограмма 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: 14.12.2020)
2. std::this\_thread::get\_id [Электронный ресурс]. Режим доступа: https://en.cppreference.com/w/cpp/thread/get\_id, свободный. (дата обращения: 14.12.2020)
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: 14.12.2020)
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: 14.12.2020)
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: 14.12.2020)
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: 14.12.2020)
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: 14.12.2020)
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://learnc.info/c/pthreads\_semaphores.html, свободный. (дата обращения: 14.12.2020)
10. Потоки, блокировки и условные переменные в C++11 [Электронный ресурс]. Режим доступа: https://habr.com/ru/post/182610/, свободный. (дата обращения: 14.12.2020)
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: 14.12.2020)
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)