

Правительство Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования «Национальный

исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Департамент программной инженерии

Отчет к домашнему заданию По дисциплине

«Архитектура вычислительных систем»

Работу выполнил:

Студент группы БПИ-194 Назмутдинов Р.Р.

Вариант 13

Москва 2020

СОДЕРЖАНИЕ

1. УСЛОВИЕ	2
2. РЕШЕНИЕ.....	3
3. ТЕСТИРОВАНИЕ.....	4
3.1. Передача корректных данных	4
3.2. Обработка некорректных данных	8
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	10

1. УСЛОВИЕ

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.

2. РЕШЕНИЕ

Для реализации программы использовалась стандартная библиотека C++ для работы с потоками. Вместе с этим также использовался заголовочный файл «semaphore.h» для работы с семафорами и библиотека «condition_variable» для работы с условными переменными.

В качестве гостиницы с 30 номерами выступает семафор **sem_t hotel**, контролирующий количество гостей в гостинице и не позволяющий заселиться в нее большему числу гостей. Для корректного вывода сообщений в консоль был создан **std::mutex messageMutex**.

Каждый гость едет в гостиницу некоторое время (от 100 миллисекунд, что меньше одного дня в программе, до 35000 миллисекунд, что равно одной неделе). Гость может приехать как на первый день, так и на седьмой. Как только гость приезжает в отель в консоль выводится сообщение «[Client {id}] Прибыл в отель». Только что прибывший гость проверяет наличие свободных мест в отеле, используя функцию **sem_getvalue(sem_t *sem, int *sval)**. В случае если мест в отеле нет, гость остается на улице и выводит в консоль следующее сообщение «[Client {id}] Придется ночевать на улице...» и ожидает освобождения комнаты. В противном же случае гость сразу заселяется в гостиницу, уменьшая значение семафора на 1 и выводя следующее сообщение в консоль «[Client {id}] Заселился в отель».

После заселения в отель каждый гость ждет начала следующего дня. Для этого используется условная переменная **std::condition_variable newDay**. Каждый гость после заселения вызывает метод **newDay.wait(lock)** и дожидается начала нового дня. Каждый 5000 миллисекунд основной поток вызывает метод **newDay.notify_all()**, давая всем ожидающим нового дня гостям понять, что новый день настал. По наступлению нового дня каждый гость, проживающий в отеле начинает готовиться к отъезду вместе с этим выводя сообщение «[Client {id}] Готовится к отъезду» в консоль. Подготовка к отъезду занимает не более одного дня (5000 миллисекунд). Как только поток готов выселяться из отеля он выводит сообщение «[Client {id}] Выселяется из отеля» в консоль и вызывает функцию **sem_post(&hotel)** тем самым увеличивая значение семафора на 1 и увеличивает на единицу глобальную переменную **countThreadLeaveFromHotel**, отвечающую за количество выселившихся гостей из гостиницы.

Как только **countThreadLeaveFromHotel** становится равно количеству гостей, указанному в начале программы, основной поток выходит из цикла, объединяет все потоки в один, удаляет массив гостей и уничтожает семафор.

3. ТЕСТИРОВАНИЕ

3.1. Передача корректных данных

Для более удобного анализа работоспособности программы в гостинице будет не 30 номеров, а всего 5, что сильно уменьшит количество выводимых данных.

При запуске программы пользователя просят ввести количество ожидаемых гостей (см. рис. 1)

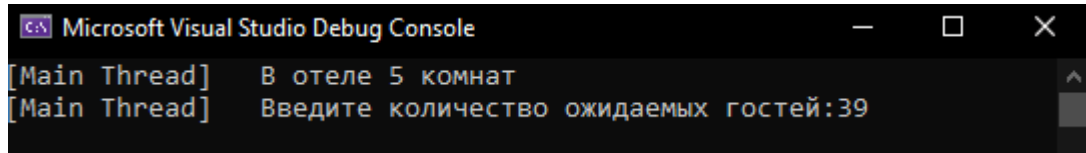


Рисунок 1. – Ввод количества гостей.

Как только пользователь вводит ожидаемое количество гостей на экран начинают выводиться итоги последующих дней (см. рис. 2 – рис. 5)

```

[Main Thread]    День 1
[Client 14]      Прибыл в отель
[Client 14]      Заселился в отель
[Client 28]      Прибыл в отель
[Client 28]      Заселился в отель
[Client 13]      Прибыл в отель
[Client 13]      Заселился в отель
[Client 27]      Прибыл в отель
[Client 27]      Заселился в отель
[Client 12]      Прибыл в отель
[Client 12]      Заселился в отель

[Main Thread]    День 2
[Client 12]      Готовится к отъезду
[Client 27]      Готовится к отъезду
[Client 13]      Готовится к отъезду
[Client 28]      Готовится к отъезду
[Client 14]      Готовится к отъезду
[Client 26]      Прибыл в отель
[Client 26]      Придется ночевать на улице...
[Client 12]      Выселяется из отеля
[Client 26]      Заселился в отель
[Client 27]      Выселяется из отеля
[Client 13]      Выселяется из отеля
[Client 11]      Прибыл в отель
[Client 11]      Заселился в отель
[Client 25]      Прибыл в отель
[Client 25]      Заселился в отель
[Client 39]      Прибыл в отель
[Client 39]      Придется ночевать на улице...
[Client 14]      Выселяется из отеля
[Client 39]      Заселился в отель
[Client 10]      Прибыл в отель
[Client 10]      Придется ночевать на улице...
[Client 24]      Прибыл в отель
[Client 24]      Придется ночевать на улице...
[Client 28]      Выселяется из отеля
[Client 10]      Заселился в отель
[Client 38]      Прибыл в отель
[Client 38]      Придется ночевать на улице...

```

Рисунок 2. – Результаты первых двух дней.

```

[Main Thread]    День 3
[Client 11]      Готовится к отъезду
[Client 39]      Готовится к отъезду
[Client 26]      Готовится к отъезду
[Client 10]      Готовится к отъезду
[Client 25]      Готовится к отъезду
[Client 25]      Выселяется из отеля
[Client 24]      Заселился в отель
[Client 11]      Выселяется из отеля
[Client 38]      Заселился в отель
[Client 26]      Выселяется из отеля
[Client 9]       Прибыл в отель
[Client 9]       Заселился в отель
[Client 23]      Прибыл в отель
[Client 23]      Придется ночевать на улице...
[Client 37]      Прибыл в отель
[Client 37]      Придется ночевать на улице...
[Client 39]      Выселяется из отеля
[Client 23]      Заселился в отель
[Client 10]      Выселяется из отеля
[Client 37]      Заселился в отель
[Client 8]       Прибыл в отель
[Client 8]       Придется ночевать на улице...
[Client 22]      Прибыл в отель
[Client 22]      Придется ночевать на улице...
[Client 36]      Прибыл в отель
[Client 36]      Придется ночевать на улице...

[Main Thread]    День 4
[Client 37]      Готовится к отъезду
[Client 9]       Готовится к отъезду
[Client 23]      Готовится к отъезду
[Client 38]      Готовится к отъезду
[Client 24]      Готовится к отъезду
[Client 7]       Прибыл в отель
[Client 7]       Придется ночевать на улице...
[Client 9]       Выселяется из отеля
[Client 8]       Заселился в отель
[Client 21]      Прибыл в отель
[Client 21]      Придется ночевать на улице...
[Client 35]      Прибыл в отель
[Client 35]      Придется ночевать на улице...
[Client 37]      Выселяется из отеля
[Client 7]       Заселился в отель
[Client 23]      Выселяется из отеля
[Client 21]      Заселился в отель
[Client 6]       Прибыл в отель
[Client 6]       Придется ночевать на улице...
[Client 20]      Прибыл в отель
[Client 20]      Придется ночевать на улице...
[Client 34]      Прибыл в отель
[Client 34]      Придется ночевать на улице...
[Client 38]      Выселяется из отеля
[Client 22]      Заселился в отель
[Client 24]      Выселяется из отеля
[Client 6]       Заселился в отель

```

Рисунок 3. – Результаты 3-го и 4-го дней.

```

[Main Thread]    День 5
[Client 6]       Готовится к отъезду
[Client 22]      Готовится к отъезду
[Client 21]      Готовится к отъезду
[Client 7]       Готовится к отъезду
[Client 8]       Готовится к отъезду
[Client 21]      Выселяется из отеля
[Client 20]      Заселился в отель
[Client 7]       Выселяется из отеля
[Client 34]      Заселился в отель
[Client 8]       Выселяется из отеля
[Client 35]      Заселился в отель
[Client 5]       Прибыл в отель
[Client 5]       Придется ночевать на улице...
[Client 19]      Прибыл в отель
[Client 19]      Придется ночевать на улице...
[Client 33]      Прибыл в отель
[Client 33]      Придется ночевать на улице...
[Client 22]      Выселяется из отеля
[Client 36]      Заселился в отель
[Client 4]       Прибыл в отель
[Client 4]       Придется ночевать на улице...
[Client 18]      Прибыл в отель
[Client 18]      Придется ночевать на улице...
[Client 32]      Прибыл в отель
[Client 32]      Придется ночевать на улице...
[Client 6]       Выселяется из отеля
[Client 5]       Заселился в отель

[Main Thread]    День 6
[Client 5]       Готовится к отъезду
[Client 34]      Готовится к отъезду
[Client 36]      Готовится к отъезду
[Client 35]      Готовится к отъезду
[Client 20]      Готовится к отъезду
[Client 3]       Прибыл в отель
[Client 3]       Придется ночевать на улице...
[Client 17]      Прибыл в отель
[Client 17]      Придется ночевать на улице...
[Client 31]      Прибыл в отель
[Client 31]      Придется ночевать на улице...
[Client 34]      Выселяется из отеля
[Client 4]       Заселился в отель
[Client 35]      Выселяется из отеля
[Client 3]       Заселился в отель
[Client 36]      Выселяется из отеля
[Client 17]      Заселился в отель
[Client 2]       Прибыл в отель
[Client 2]       Придется ночевать на улице...
[Client 16]      Прибыл в отель
[Client 16]      Придется ночевать на улице...
[Client 30]      Прибыл в отель
[Client 30]      Придется ночевать на улице...
[Client 5]       Выселяется из отеля
[Client 18]      Заселился в отель
[Client 20]      Выселяется из отеля
[Client 2]       Заселился в отель

```

Рисунок 4. – Результат 5-го и 6-го дней.


```

[Main Thread]    День 7
[Client 18]      Готовится к отъезду
[Client 3]       Готовится к отъезду
[Client 4]       Готовится к отъезду
[Client 2]       Готовится к отъезду
[Client 17]      Готовится к отъезду
[Client 1]       Прибыл в отель
[Client 1]       Придется ночевать на улице...
[Client 15]      Прибыл в отель
[Client 15]      Придется ночевать на улице...
[Client 4]       Выселяется из отеля
[Client 16]      Заселился в отель
[Client 29]      Прибыл в отель
[Client 29]      Придется ночевать на улице...
[Client 17]      Выселяется из отеля
[Client 1]       Заселился в отель
[Client 18]      Выселяется из отеля
[Client 15]      Заселился в отель
[Client 2]       Выселяется из отеля
[Client 19]      Заселился в отель
[Client 3]       Выселяется из отеля
[Client 29]      Заселился в отель

[Main Thread]    День 8
[Client 29]      Готовится к отъезду
[Client 15]      Готовится к отъезду
[Client 19]      Готовится к отъезду
[Client 16]      Готовится к отъезду
[Client 1]       Готовится к отъезду
[Client 29]      Выселяется из отеля
[Client 30]      Заселился в отель
[Client 16]      Выселяется из отеля
[Client 31]      Заселился в отель
[Client 15]      Выселяется из отеля
[Client 32]      Заселился в отель
[Client 1]       Выселяется из отеля
[Client 33]      Заселился в отель
[Client 19]      Выселяется из отеля

[Main Thread]    День 9
[Client 31]      Готовится к отъезду
[Client 33]      Готовится к отъезду
[Client 32]      Готовится к отъезду
[Client 30]      Готовится к отъезду
[Client 33]      Выселяется из отеля
[Client 30]      Выселяется из отеля
[Client 31]      Выселяется из отеля
[Client 32]      Выселяется из отеля

```

Рисунок 5. – Результаты 7-9 дней.

3.2. Обработка некорректных данных

Программа не позволяет пользователю ввести некорректные входные данные (см. рис. 6)

```
[Main Thread] В отеле 5 комнат
[Main Thread] Введите количество ожидаемых гостей:-123
[Main Thread] Ну хотя бы один человек к нам-то уж приезжает
[Main Thread] Введите число гостей еще раз:0
[Main Thread] Ну хотя бы один человек к нам-то уж приезжает
[Main Thread] Введите число гостей еще раз:1001
[Main Thread] Слишком большое количество посетителей, у нас так много не бывает
[Main Thread] Введите число гостей еще раз:1000
```

Рисунок 6. – Обработка некорректных данных

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Cppreference (2020) «Документация по C++: std::condition_variable::wait» (https://en.cppreference.com/w/cpp/thread/condition_variable/wait).
2. Cppreference (2020) «Документация по C++: std::mutex» (<https://ru.cppreference.com/w/cpp/thread/mutex>).
3. Docs Microsoft (2020) «Creating Threads» (<https://docs.microsoft.com/en-us/windows/win32/procthread/creating-threads>)
4. Легалов А.И.(2020) «Архитектура параллельных вычислительных систем. Многопоточность » (<http://softcraft.ru/edu/comparch/lect/07-parthread/>)
5. Легалов А.И.(2020) «Многопоточность. Простая многопоточная программа. Основные функции» (<http://softcraft.ru/edu/comparch/practice/thread/01-simple/>).
6. Легалов А.И.(2020) «Многопоточность. Синхронизация потоков. Методы синхронизации» (<http://softcraft.ru/edu/comparch/practice/thread/02-sync/>).
7. Хабр «Такие удивительные семафоры» (<https://habr.com/ru/post/261273/>)
8. Learn.info «Семафоры: введение» (https://learn.info/c/pthreads_semaphores.html)