

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

Факультет компьютерных наук
Департамент программной инженерии

Микропроект 2

Вариант 13

Пояснительная записка

Исполнитель
студент группы БПИ 199
М.А. Кузнецов

Москва 2020

СОДЕРЖАНИЕ

1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ.....	3
1.1 Постановка задачи.....	3
2. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ.....	3
2.1 Описание сценария взаимодействия субъектов.....	3
2.2 Описание протокола взаимодействия субъектов.....	3
2.3 Описание алгоритма вычисления текущего времени в гостинице.....	3
2.4 Описание алгоритма вычисления времени нахождения в гостинице	3
2.5 Описание входных данных.....	3
2.6 Описание выходных данных	4
3. ТЕСТЫ.....	4
3.1 Проверка числа входных аргументов	4
3.2 Проверка длины часа	4
3.3 Проверка верхней границы задержки.....	4
3.4 Проверка числа клиентов.....	5
3.5 Ввод корректных данных	5
4. ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА.....	6

1. ФУНКЦИОНАЛЬНЫЕ ТРЕБОВАНИЯ

1.1 Постановка задачи

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. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

2.1 Описание сценария взаимодействия субъектов

Гостиница работает круглосуточно. Клиенты приходят в любое время в течение суток. Будем считать, что они приходят только в начале часа. Сначала клиент заходит в гостиницу. Если есть какой-то свободный номер, то клиент получает его и остается в нем на ночь. Так как каждый клиент остается именно на ночь, то покидает свой номер он только после 6 утра. Если клиент приходит после 6 утра, то он остается в гостинице до следующего дня. При этом он не сразу покидает свой номера. Каждый клиент может пробыть в своем номере еще максимум 3 часа после 6 часов утра

2.2 Описание протокола взаимодействия субъектов

Каждый клиент после того, как зашел в гостиницу, уменьшает счетчик семафора на 1. Если он равен -1, то клиента не регистрируют и не пускают дальше. Он ждет пока счетчик перестанет быть равным -1. Если же он больше -1, то клиент регистрируется и остается в гостинице. После того, как время нахождения в гостинице прошло и клиент должен покинуть ее, он увеличивает счетчик семафора на единицу.

После того, как в гостиницу зашел последний клиент, программа ждет пока номера в гостинице не опустеют.

2.3 Описание алгоритма вычисления текущего времени в гостинице

Будем считать, что при запуске программы в гостинице 12 часов дня.

Длина дня равна произведению числа секунд в часе на число часов в дне, то есть на 24

В начале работы программа сохраняет текущее время на компьютере.

Для того чтобы получить текущее время в гостинице нужно из текущего системного времени на компьютере вычесть сохраненное при старте программы время и получить число пройденных с момента запуска секунд. К нему прибавляется число равное длине половины дня, так как отсчет времени в гостинице начинается с 12 часов дня (то есть половины от 24 часов). Итоговое время будет равно частному остатка от деления нашей суммы на длину дня и количества секунд в часе. (см. рис. 1)

$$\frac{\left(currentTime - startTime + \frac{dayLength}{2} \right) \bmod dayLength}{hourLength}$$

Рисунок 1– Формула вычисления текущего времени в гостинице

2.4 Описание алгоритма вычисления времени нахождения в гостинице

Сначала вычисляется сколько часов по модулю длины дня в гостинице прошло с запуска программы и сохраняется в переменную `currentTime`.

Если прошло как минимум 18 часов, то мы сначала узнаем, сколько часов осталось до 24 и к этому числу прибавляем 18, так как с 12 часов дня до 6 утра должно пройти 18 часов. После чего прибавляем к результату случайное число часов от 0 до 3

Если прошло меньше 18 часов, то мы из 18 часов вычитаем значение, которое хранится в переменной `currentTime`, чтобы узнать сколько времени должно пройти до 6 часов утра. После чего прибавляем к результату случайное число часов от 0 до 3

2.5 Описание входных данных

На вход программы подается два числа в качестве аргументов командной строки:

`./a.out <длина часа в секундах> <верхняя граница задержки> <число клиентов>`

Длина часа должна быть положительным целым числом.

Верхняя граница задержки – это максимальное допустимое число часов между появлениями клиентов в гостинице. Данный параметр должен быть положительным целым числом.

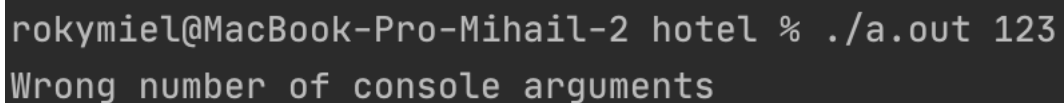
Число клиентов должно быть неотрицательным числом

2.6 Описание выходных данных

В ходе выполнения программа выводит информацию о совершённом клиентом действии и времени его совершения.

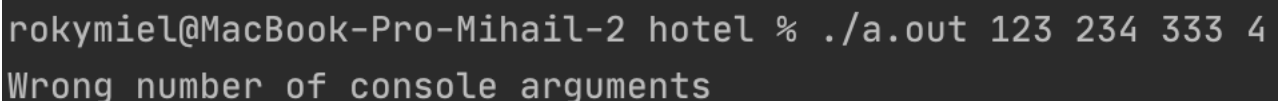
3. ТЕСТЫ

3.1 Проверка числа входных аргументов



```
rokymiel@MacBook-Pro-Mihail-2 hotel % ./a.out 123
Wrong number of console arguments
```

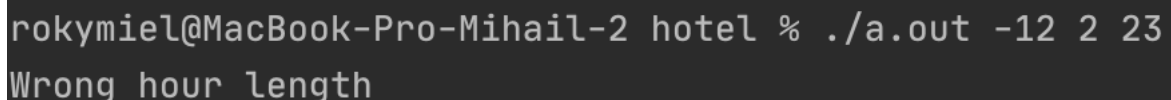
Рисунок 2– Ввод меньше трех аргументов



```
rokymiel@MacBook-Pro-Mihail-2 hotel % ./a.out 123 234 333 4
Wrong number of console arguments
```

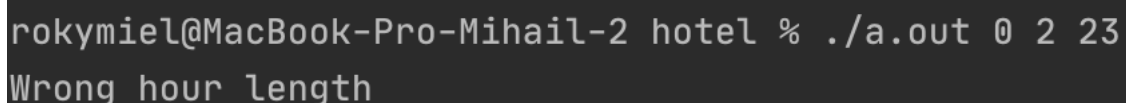
Рисунок 3– Ввод больше трех аргументов

3.2 Проверка длины часа



```
rokymiel@MacBook-Pro-Mihail-2 hotel % ./a.out -12 2 23
Wrong hour length
```

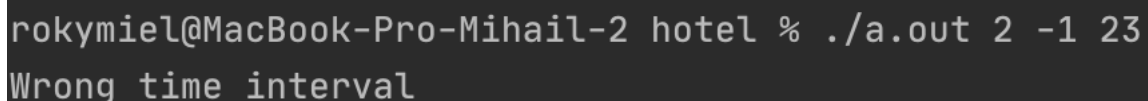
Рисунок 4– Ввод отрицательной длины часа



```
rokymiel@MacBook-Pro-Mihail-2 hotel % ./a.out 0 2 23
Wrong hour length
```

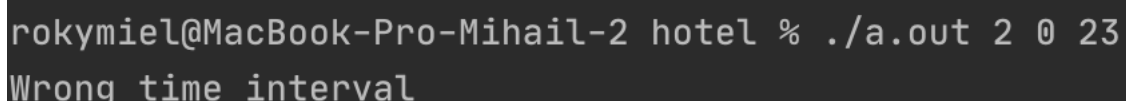
Рисунок 5– Ввод нулевой длины часа

3.3 Проверка верхней границы задержки



```
rokymiel@MacBook-Pro-Mihail-2 hotel % ./a.out 2 -1 23
Wrong time interval
```

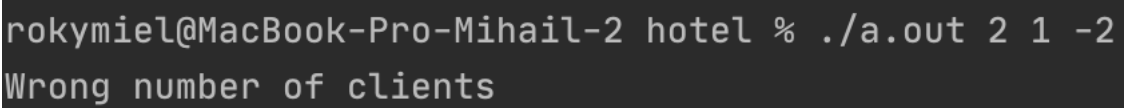
Рисунок 6– Ввод отрицательной границы



```
rokymiel@MacBook-Pro-Mihail-2 hotel % ./a.out 2 0 23
Wrong time interval
```

Рисунок 7– Ввод нулевой границы

3.4 Проверка числа клиентов



```
rokyriel@MacBook-Pro-Mihail-2 hotel % ./a.out 2 1 -2
Wrong number of clients
```

Рисунок 8– Ввод отрицательного числа клиентов

3.5 Ввод корректных данных

Из-за большого объема выходных данных результаты тестов находятся в папке «Тесты»

4. ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

- 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: 10.12.20).
- 2) Руководство по языку C++ [Электронный ресурс] Режим доступа: <https://en.cppreference.com> , свободный. (дата обращения: 11.12.20).
- 3) Многопоточное программирование. Синхронизация / Примеры программ [Электронный ресурс] Режим доступа: <http://softcraft.ru/edu/comparch/practice/thread/02-sync/> , свободный. (дата обращения: 11.12.20).
- 4) Unetwa. Сообщество программистов. C++ / Дата и время [Электронный ресурс] Режим доступа: <https://unetway.com/tutorial/c-data-i-vrema> , свободный. (дата обращения: 11.12.20).
- 5)