

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук, Департамент программной инженерии
Дисциплина: «Архитектура вычислительных систем»

Вариант 1

ПРОБЛЕМА СПЯЩЕГО БРАДОБРЕЯ

Пояснительная записка

Выполнил
студент БПИ197
Алексеев Валерий Михайлович

**Москва
2021**

Содержание

1. Текст задания.....	3
2. Применяемые расчетные методы.....	4
2.1. Теория решения задания.....	4
2.2. Дополнительный функционал программы.....	4
3. Тестирование программы.....	5
3.1. Корректные значения.....	5
3.2. Некорректные значения.....	7
3.3. Результаты тестирования.....	8
Список литературы.....	9

1. Текст задания

Задача о парикмахере.

В тихом городке есть парикмахерская. Салонпарикмахерской мал, ходить там может только парикмахер и один посетитель. Парикмахер всю жизнь обслуживает посетителей. Когда в салоне никого нет, он спит в кресле. Когда посетитель приходит и видит спящего парикмахера, он будит его, садится в кресло и спит, пока парикмахер занят стрижкой. Если посетитель приходит, а парикмахер занят, то он встает в очередь и засыпает. После стрижки парикмахер сам провожает посетителя. Если есть ожидающие посетители, то парикмахер будит одного из них и ждет пока тот сядет в кресло парикмахера и начинает стрижку. Если никого нет, он снова садится в свое кресло и засыпает до прихода посетителя. Создать многопоточное приложение, моделирующее рабочий день парикмахерской.

2. Применяемые расчетные методы

2.1. Теория решения задания

В информатике проблема спящего парикмахера — классическая проблема синхронизации и межпроцессного взаимодействия (interprocess) в многопроцессорной OS.

Доступно множество возможных решений. Основной элемент каждого — mutex, который гарантирует, что изменить состояние (isBusy) может только один из участников. Парикмахер должен захватить это mutex исключение, прежде чем проверить клиентов, и освободить его, когда он начинает или спать, или работать.

Клиент должен захватить mutex, прежде чем войти в магазин, и освободить его, как только он займет место или в приемной, или у парикмахера. Это устраняет обе проблемы, упомянутые в предыдущей секции. Семафоры также обязаны указывать на состояние системы. Например, можно было бы сохранить число людей в приемной.

2.2. Дополнительный функционал программы

Пользователь вводит количество посетителей, максимальное время работы парикмахера с клиентом, время между приходом посетителей. Все неверные значения вводимых данных заканчивают выполнение программы.

3. Тестирование программы

При запуске программы через консоль программа требует от пользователя ввод трех числовых значений:

- 1) Количество посетителей за день (целочисленное значение)
- 2) Максимальное время работы парикмахера с клиентом (значение с плавающей точкой в секундах)
- 3) Время между приходом посетителей (значение с плавающей точкой в секундах).

3.1. Корректные данные

Проверим корректные данные двух случаях:

- 1) Максимальное время работы парикмахера с клиентом меньше времени между приходом посетителей (1 секунда и 3 секунды) и 5 посетителей
- 2) Максимальное время работы парикмахера с клиентом больше времени между приходом посетителей (1 секунда и 0.5 секунды) и 5 посетителей.

```
valera@valera-MacBookAir:~/CLionProjects/Barber$ sudo g++ -pthread main.cpp -o main
valera@valera-MacBookAir:~/CLionProjects/Barber$ ./main
Please enter the total customers: 5
Please enter the maximum served time (s): 1
Please enter the time between customers (s): 3
The working day has started.
Customer {1} is sleeping.
Customer {1} is being served.
Customer was served.
Customer {2} is sleeping.
Customer {2} is being served.
Customer was served.
Customer {3} is sleeping.
Customer {3} is being served.
Customer was served.
Customer {4} is sleeping.
Customer {4} is being served.
Customer was served.
Customer {5} is sleeping.
Customer {5} is being served.
Customer was served.
The working day has ended.
valera@valera-MacBookAir:~/CLionProjects/Barber$
```

Рисунок 1. Корректный ввод. Пример 1.

```
valera@valera-MacBookAir:~/CLionProjects/Barber$ sudo g++ -pthread main.cpp -o main
valera@valera-MacBookAir:~/CLionProjects/Barber$ ./main
Please enter the total customers: 5
Please enter the maximum served time (s): 1
Please enter the time between customers (s): 0.5
The working day has started.
Customer {1} is sleeping.
Customer {1} is being served.
Customer {2} is sleeping.
Customer {4} is sleeping.
Customer {5} is sleeping.
Customer {3} is sleeping.
Customer was served.
Customer {2} is being served.
Customer was served.
Customer {4} is being served.
Customer was served.
Customer {5} is being served.
Customer was served.
Customer {3} is being served.
Customer was served.
The working day has ended.
valera@valera-MacBookAir:~/CLionProjects/Barber$
```

Рисунок 2. Корректный ввод. Пример 2.

3.2. Некорректные данные

Чтобы проверить работу программы при некорректном вводе, введем неверные значения для разных входных данных (отрицательные значения и нечисловые).

```
valera@valera-MacBookAir:~/CLionProjects/Barber$ sudo g++ -pthread main.cpp -o main
valera@valera-MacBookAir:~/CLionProjects/Barber$ ./main
Please enter the total customers: -2
Incorrect number of total customers!
valera@valera-MacBookAir:~/CLionProjects/Barber$ ./main
Please enter the total customers: 4
Please enter the maximum served time (s): asd
Incorrect number of maximum serving time!
valera@valera-MacBookAir:~/CLionProjects/Barber$ ./main
Please enter the total customers: 10
Please enter the maximum served time (s): 2
Please enter the time between customers (s): -4
Incorrect number of time between customers!
valera@valera-MacBookAir:~/CLionProjects/Barber$ #
```

Рисунок 3. Некорректный ввод.

3.3. Результаты тестирования

Тестирование прошло успешно:

1) Если парикмахер работает с клиентом меньше, чем промежутки между приходом новых посетителей, то новые посетители не ждут как он доработает и сразу же садятся в кресло.

2) Если парикмахер работает с клиентом дольше, чем промежутки между приходом новых посетителей, то новые посетители ждут как он доработает и после этого садятся в кресло.

3) При некорректных данных программа выдает сообщение об ошибке и сразу же завершает работу.

Список литературы

1. Таненбаум Э. Архитектура компьютера. 6-е изд. —СПб.: Изд. Питер, 2017. – 816 с.
2. POSIX Threads. [Электронный ресурс] // Режим доступа: свободный, URL: https://ru.wikipedia.org/wiki/POSIX_Threads (дата обращения: 13.12.2020)
3. The Sleeping-Barbers Problem. [Электронный ресурс] // Режим доступа: свободный, URL: <http://courses.washington.edu/css503/prog/prog2.pdf> (дата обращения: 13.12.2020)
4. Проблема спящего парикмахера. [Электронный ресурс] // Режим доступа: свободный, URL: <https://dic.academic.ru/dic.nsf/ruwiki/1854686> (дата обращения: 13.12.2020)