

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

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

**Вариант 1**

**ПРОБЛЕМА СПЯЩЕГО БРАДОБРЕЯ**

Пояснительная записка

**Выполнила:**  
Абу Аль Лабан Надя,  
*студент гр. БПИ198.*

**Москва**  
2020

## Содержание

<b>1. Текст задания.....</b>	<b>3</b>
<b>2. Применяемые расчетные методы .....</b>	<b>4</b>
<b>2.1. Теория решения задания .....</b>	<b>4</b>
<b>2.2. Дополнительный функционал программы .....</b>	<b>4</b>
<b>3. Тестирование программы.....</b>	<b>5</b>
<b>3.1. Корректные значения .....</b>	<b>5</b>
<b>3.2. Некорректные значения .....</b>	<b>7</b>
<b>Список литературы.....</b>	<b>8</b>

## 1. Текст задания

Задача о парикмахере.

В тихом городке есть парикмахерская. Салон парикмахерской 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.

### **2.2. Дополнительный функционал программы**

Пользователь вводит количество посетителей. Поскольку это число должно быть положительным и целым, при попытке ввести число меньше 1 или строку программа заменяет количество посетителей на значение по умолчанию (1).

### 3. Тестирование программы

При запуске программы через консоль нужно первым (и единственным) аргументом ввести количество посетителей.

#### 3.1. Корректные значения

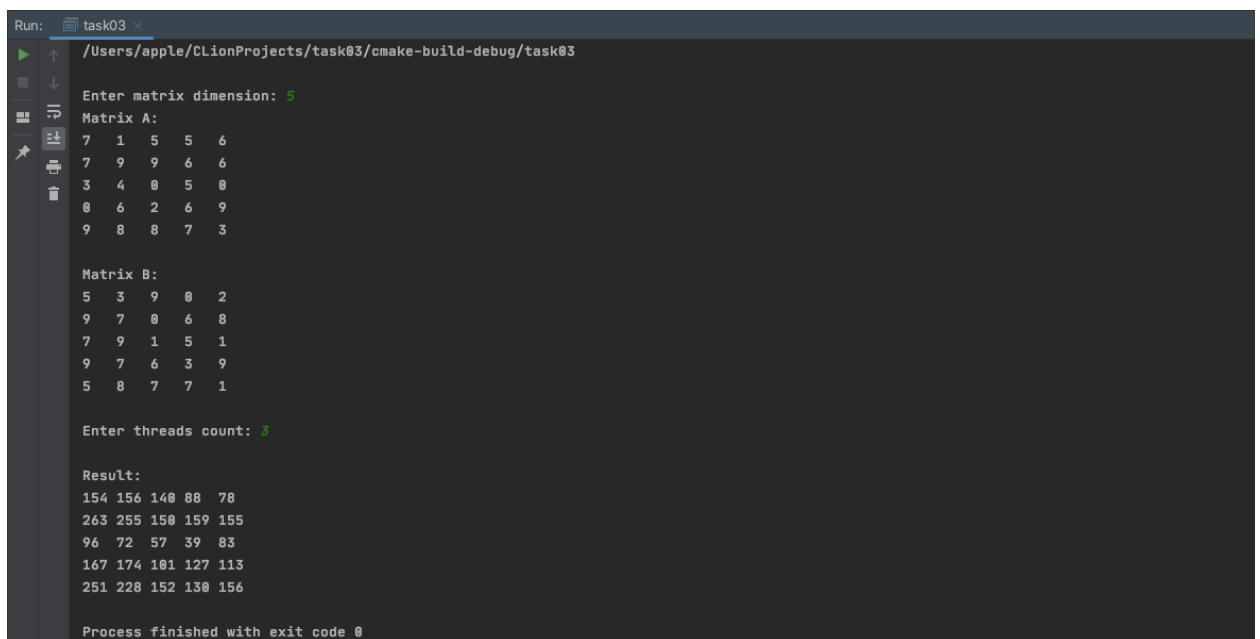
Для начала введем корректные значения и проверим, что программа работоспособна.

Введем 20 и увидим вывод (рис.1)

```
C:\Users\HYPERPC\source\repos\Nadya\Nadya>Nadya 20
Barber is sleeping
Barber is sleeping
Client 19 is having a new haircut.
Client 7 is having a new haircut.
Client 20 is having a new haircut.
Client 8 is having a new haircut.
Client 9 is having a new haircut.
Barber is sleeping
Client 10 is having a new haircut.
Client 11 is having a new haircut.
Client 12 is having a new haircut.
Barber is sleeping
Client 13 is having a new haircut.
Client 1 is having a new haircut.
Client 2 is having a new haircut.
Client 14 is having a new haircut.
Client 15 is having a new haircut.
Client 3 is having a new haircut.
Barber is sleeping
Client 16 is having a new haircut.
Client 4 is having a new haircut.
Client 17 is having a new haircut.
Client 5 is having a new haircut.
Client 18 is having a new haircut.
Client 6 is having a new haircut.
Barber is sleeping
Barber is sleeping
Barber is sleeping
Barber is sleeping
Barber is sleeping
Barber is sleeping
Barber is sleeping
```

*.Рисунок 1. Корректный ввод.*

Введем значение для размерности 5 и количество потоков меньше 5, например, 3 (рис. 4). Результат верный (рис. 5).



```

Run: task03 x
/Users/apple/CLionProjects/task03/cmake-build-debug/task03

Enter matrix dimension: 5
Matrix A:
7 1 5 5 6
7 9 9 6 6
3 4 0 5 0
0 6 2 6 9
9 8 8 7 3

Matrix B:
5 3 9 0 2
9 7 0 6 8
7 9 1 5 1
9 7 6 3 9
5 8 7 7 1

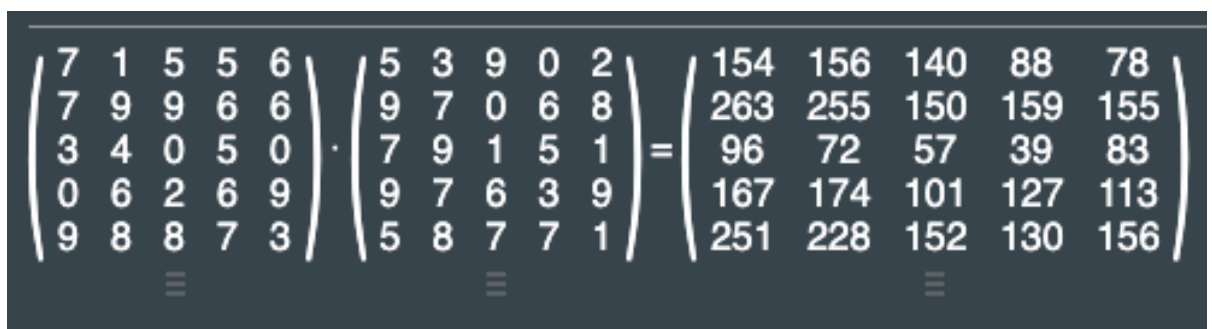
Enter threads count: 3

Result:
154 156 140 88 78
263 255 150 159 155
96 72 57 39 83
167 174 101 127 113
251 228 152 130 156

Process finished with exit code 0

```

Рисунок 4. Ввод количества потоков меньшего, чем размерность.

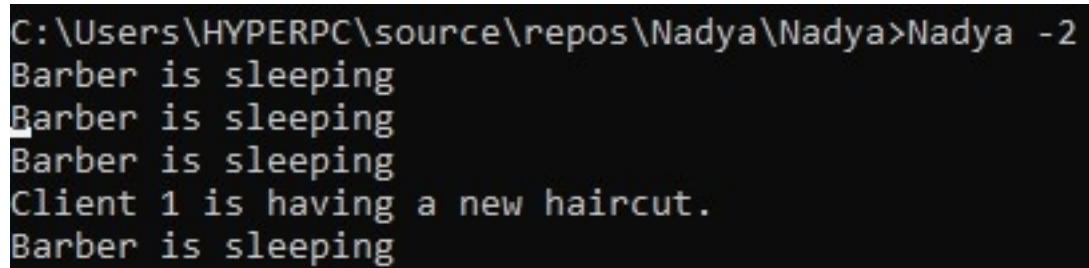


$$\begin{pmatrix} 7 & 1 & 5 & 5 & 6 \\ 7 & 9 & 9 & 6 & 6 \\ 3 & 4 & 0 & 5 & 0 \\ 0 & 6 & 2 & 6 & 9 \\ 9 & 8 & 8 & 7 & 3 \end{pmatrix} \cdot \begin{pmatrix} 5 & 3 & 9 & 0 & 2 \\ 9 & 7 & 0 & 6 & 8 \\ 7 & 9 & 1 & 5 & 1 \\ 9 & 7 & 6 & 3 & 9 \\ 5 & 8 & 7 & 7 & 1 \end{pmatrix} = \begin{pmatrix} 154 & 156 & 140 & 88 & 78 \\ 263 & 255 & 150 & 159 & 155 \\ 96 & 72 & 57 & 39 & 83 \\ 167 & 174 & 101 & 127 & 113 \\ 251 & 228 & 152 & 130 & 156 \end{pmatrix}$$

Рисунок 5. Подсчет результата второго примера на калькуляторе.

### 3.2. Некорректные значения

Чтобы проверить работу программы при некорректном вводе, введем отрицательное число.



```
C:\Users\HYPERPC\source\repos\Nadya\Nadya>Nadya -2
Barber is sleeping
Barber is sleeping
Barber is sleeping
Client 1 is having a new haircut.
Barber is sleeping
```

*Рисунок 2. Некорректный ввод.*

Таким образом, мы показали, что программа обрабатывает некорректные данные и работает на значениях по умолчанию.

**Список литературы**

1. The Sleeping-Barbers Problem. [Электронный ресурс] // Режим доступа: свободный, URL: <http://courses.washington.edu/css503/prog/prog2.pdf> (дата обращения: 13.12.2020)
2. Проблема спящего парикмахера. [Электронный ресурс] // Режим доступа: свободный, URL: <https://dic.academic.ru/dic.nsf/ruwiki/1854686> (дата обращения: 13.12.2020)