

**Высшая школа экономики**  
**Факультет компьютерных наук**  
**«Программная инженерия»**

**Андреев Аркадий 192**

**Вариант №1**

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.

## Модель.

Мною была использована клиент-серверная модель параллелизма. Я выбрал ее, так как она хорошо олицетворяет мою задачу. Есть поток парикмахера (сервер), к которому по очереди обращаются потоки-клиенты. Когда потоки-клиенты ждут ответа (стригутся), они засыпают. Когда сервер (парикмахер) не получает запросов (нет клиентов), он спит.

Таким образом у меня есть функция для потоков-клиентов, которая состоит из двух mutex блоков. Первый отвечает за то, чтобы встать в очередь. Предположим, что дверь в парикмахерскую комнату ожидания одна, а значит в нее не смогут войти одновременно два и более клиентов. Как только поток заходит в этот блок его номер записывается в очередь (queue) и, если перед ним нет клиентов ни в комнате ожидания, ни в кресле парикмахера, то он будет спать парикмахера с помощью переменной состояния. Далее поток засыпает до тех пор, пока не получит ответа от сервера (парикмахера) о том, что стрижка завершена.

Также у меня есть функция, описывающая логику потока-парикмахера. Я завел булеву переменную, обозначающую конец рабочего дня. Она переводится в положение true, когда все потоки-клиенты достигнут конца программы (join). Таким образом, поток парикмахер «крутится» в mutex блоке до тех пор, пока не наступит конец рабочего дня. Внутри этого while есть еще два while условия, расположенных «параллельно». Первый проверяет есть ли клиенты, в комнате ожидания. Если нет, то поток засыпает через переменную состояния (к которой обращаются потоки-клиенты, когда видят спящего парикмахера). Второй while «крутится» до тех пор, пока в очереди есть клиенты. Он как раз и отвечает за стрижку. Как только стрижка заканчивается, поток-парикмахер отправляет запрос через переменную состояния постриженному потоку-клиенту, чтобы тот проснулся и освободил место для следующего. Если следующего клиента нет, а рабочий день еще не закончен, то парикмахер снова засыпает.

Подводя итоги, в моей программе использовано 3 mutex: для моделирования процесса входа в комнату ожидания, для моделирования процесса стрижки со стороны клиента и для моделирования процесса стрижки и сна со стороны парикмахера. Также я использовал две переменные состояния: первая для сна потока-клиента во время стрижки и вторая для сна потока-парикмахера во время отсутствия клиентов.

Вся информация была взята с сайта [softcraft.ru](http://softcraft.ru).

P.S. я осознаю, что в данном случае использование структуры queue было необязательно, но я оставил ее для наглядности.

## Тестирование.

Все необходимые исходные данные задаются в диалоге после запуска программы.

В моей программе на вход подается количество потоков-клиентов. Время входа каждого клиента в комнату ожидания генерируется с помощью рандома в диапазоне [2000, 5000)

мс. Время стрижки каждого клиента также генерируется случайно в диапазоне [1000, 4000) мс.

Если программа получает некорректные значения, то просто завершает работу (не аварийно).

```
Enter quantity of threads(customers):f
f isn't a number

Process finished with exit code 0
```

```
Enter quantity of threads(customers):-1
-1 is illegal!

Process finished with exit code 0
```

```
Enter quantity of threads(customers):0
Illegal quantity of threads!

Process finished with exit code 0
```

Далее я продемонстрирую тесты на от 1 до 5 клиентах. С 1-4 клиенты приходят с таким интервалом, что парикмахер всегда успевает поспать. На тесте с 5-ю клиентами под конец рабочего дня случился «завал», так что у парикмахера не было возможности поспать. В качестве номера клиента используется id потока.

Что касается комментариев программы: “Barber is sleeping” означает, что парикмахер спит. “Customer x entered” означает, что клиент вошел в комнату ожидания. “x is cutting...” означает, что клиент x находится в кресле парикмахера. “x is already cut!” означает, что клиент x пострижен и парикмахер его будит. “exit.” означает, что клиент вышел из парикмахерской.

```
Enter quantity of threads(customers):1
2828      Barber is sleeping...
4870      Customer 3 entered.
4879      3 is cutting...
5923      3 is already cut!
5927      3 exit.
```

```
Enter quantity of threads(customers):2
2341      Barber is sleeping...
4384      Customer 3 entered.
4396      3 is cutting...
5449      3 is already cut!
5452      3 exit.
5653      Barber is sleeping...
6437      Customer 4 entered.
6449      4 is cutting...
7922      4 is already cut!
7937      4 exit.
```

```
Enter quantity of threads(customers):3
1612     Barber is sleeping...
3654     Customer 3 entered.
3665     3 is cutting...
4715     3 is already cut!
4725     3 exit.
4926     Barber is sleeping...
5706     Customer 4 entered.
5719     4 is cutting...
7190     4 is already cut!
7197     4 exit.
7398     Barber is sleeping...
7760     Customer 5 entered.
7773     5 is cutting...
9109     5 is already cut!
9111     5 exit.
```

```
Enter quantity of threads(customers):4
1759     Barber is sleeping...
3802     Customer 3 entered.
3810     3 is cutting...
4853     3 is already cut!
4855     3 exit.
5056     Barber is sleeping...
5851     Customer 4 entered.
5860     4 is cutting...
7330     4 is already cut!
7340     4 exit.
7540     Barber is sleeping...
7901     Customer 5 entered.
7903     5 is cutting...
9238     5 is already cut!
9240     5 exit.
9440     Barber is sleeping...
9945     Customer 6 entered.
9961     6 is cutting...
13470    6 is already cut!
13480    6 exit.
```

Как я писал выше, на пятом тесте во время стрижки четвертого(id6) клиента пришел пятый(id7).

```
Enter quantity of threads(customers):5
1888     Barber is sleeping...
3931     Customer 3 entered.
3936     3 is cutting...
4979     3 is already cut!
4980     3 exit.
5181     Barber is sleeping...
5978     Customer 4 entered.
5990     4 is cutting...
7467     4 is already cut!
7469     4 exit.
7670     Barber is sleeping...
8032     Customer 5 entered.
8036     5 is cutting...
9371     5 is already cut!
9373     5 exit.
9574     Barber is sleeping...
10077    Customer 6 entered.
10078    6 is cutting...
12120    Customer 7 entered.
13580    6 is already cut!
13581    6 exit.
13781    7 is cutting...
15952    7 is already cut!
15963    7 exit.
```