**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

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

**ПАРИКМАХЕРСКАЯ “У БИЛЛИ” МИКРОПРОЕКТ**

Пояснительная записка

**Выполнил:**   
Фомин Иван,   
*студент гр. БПИ197.*

**Москва**2020

Содержание

[1. Текст задания 2](#_Toc55056090)

[2. Применяемые расчетные методы](#_Toc55056091) 3

[2.1. Теория решения задания](#_Toc55056092) 3

[3. Тестирование программы](#_Toc55056094) 3

[ПРИЛОЖЕНИЕ 1.](#_Toc55056097) [Список литературы](#_Toc55056098) 6

[ПРИЛОЖЕНИЕ 2.](#_Toc55056099) [Код программы](#_Toc55056100) 7

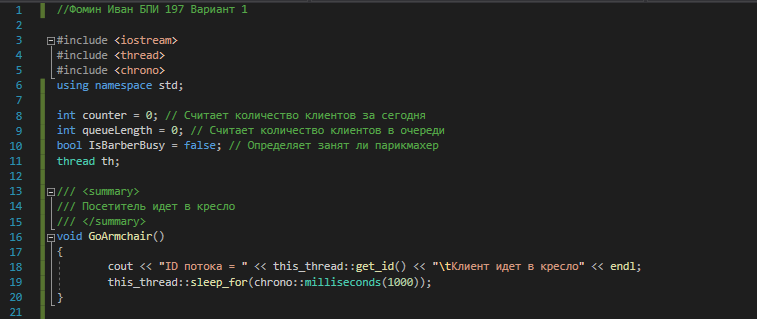
1. Текст задания

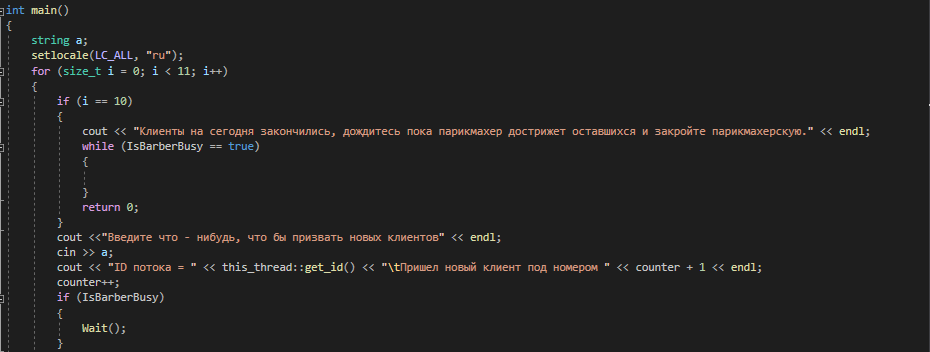
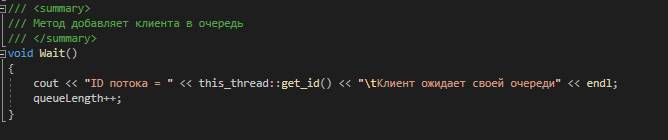
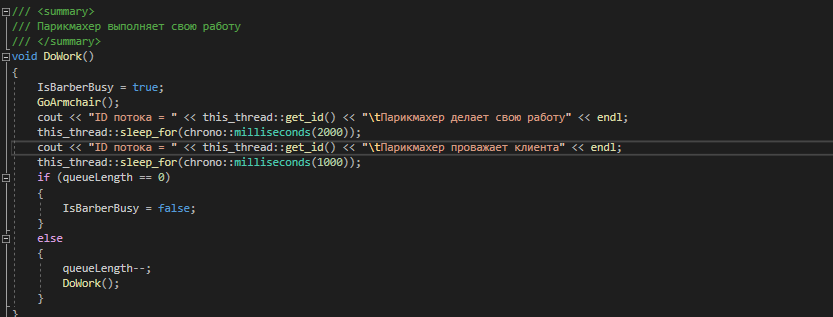
В тихом городке есть парикмахерская. Салон парикмахерской мал, ходить там может только парикмахер и один посетитель. Парикмахер всю жизнь обслуживает посетителей. Когда в салоне никого нет, он спит в кресле. Когда посетитель приходит и видит спящего парикмахера, он будет его, садится в кресло и спит, 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.

Применяемые расчетные методы

* 1. Теория решения задания

Для решения данной задачи мною было использовано два потока. Один поток моделировал парикмахера и его работу, а второй поток моделировал клиентов и очередь. Пусть за рабочий день к парикмахеру приходит 10 клиентов. Тогда если клиент приходит, а очереди нет, то вызывается метод DoWork() и парикмахер приступает к работе в отдельном потоке (одновременно с этим могут приходить еще клиенты). Если на момент прихода клиента парикмахер занят, то клиент добавляется в очередь и его начинают стричь сразу после ухода последнего клиента перед ним. Если пришли все 10 клиентов, то ждем, пока парикмахер закончит работу и завершаем день.



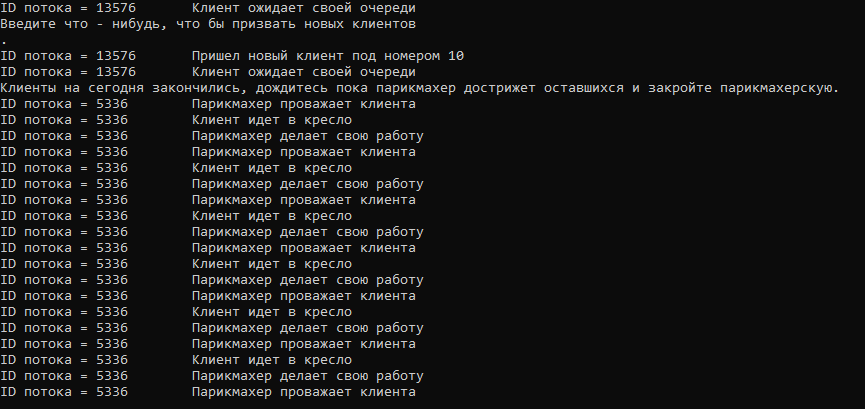
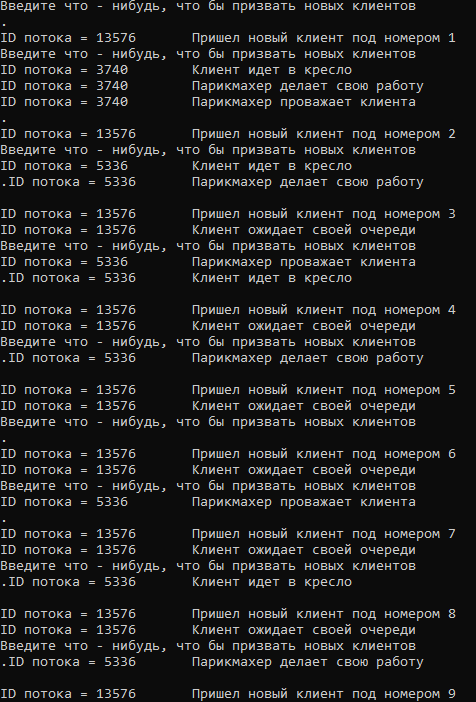


*Рисунок 1. Код программы*

1. Тестирование программы

Пользователь в любой момент времени может призвать нового клиента путем ввода чего – либо в командную строку/консоль. Призванные клиенты будут либо немедленно обслужены, либо добавлены в очередь.

Программа будет выводить ID потока и информацию о состоянии клиента/парикмахера.



*Рисунок 2. Пример результата работы программы*

ПРИЛОЖЕНИЕ 1

Список литературы

1. Потоки, блокировки и условные переменные в C++// [https://habr.com/ru/post/182610/](https://habr.com/ru/post/182610/%20) (14.12.2020)
2. Многопоточность C++ // [https://www.youtube.com/watch?v=NawpxG81RRk&t](https://www.youtube.com/watch?v=NawpxG81RRk&t%20) (14.12.2020)

ПРИЛОЖЕНИЕ 2

Код программы

//Фомин Иван БПИ 197 Вариант 1

#include <iostream>

#include <thread>

#include <chrono>

using namespace std;

int counter = 0; // Считает количество клиентов за сегодня

int queueLength = 0; // Считает количество клиентов в очереди

bool IsBarberBusy = false; // Определяет занят ли парикмахер

thread th;

/// <summary>

/// Посетитель идет в кресло

/// </summary>

void GoArmchair()

{

cout << "ID потока = " << this\_thread::get\_id() << "\tКлиент идет в кресло" << endl;

this\_thread::sleep\_for(chrono::milliseconds(1000));

}

/// <summary>

/// Парикмахер выполняет свою работу

/// </summary>

void DoWork()

{

IsBarberBusy = true;

GoArmchair();

cout << "ID потока = " << this\_thread::get\_id() << "\tПарикмахер делает свою работу" << endl;

this\_thread::sleep\_for(chrono::milliseconds(2000));

cout << "ID потока = " << this\_thread::get\_id() << "\tПарикмахер проважает клиента" << endl;

this\_thread::sleep\_for(chrono::milliseconds(1000));

if (queueLength == 0)

{

IsBarberBusy = false;

}

else

{

queueLength--;

DoWork();

}

}

/// <summary>

/// Метод добавляет клиента в очередь

/// </summary>

void Wait()

{

cout << "ID потока = " << this\_thread::get\_id() << "\tКлиент ожидает своей очереди" << endl;

queueLength++;

}

int main()

{

string a;

setlocale(LC\_ALL, "ru");

for (size\_t i = 0; i < 11; i++)

{

if (i == 10)

{

cout << "Клиенты на сегодня закончились, дождитесь пока парикмахер дострижет оставшихся и закройте парикмахерскую." << endl;

while (IsBarberBusy == true) // Дожидаемся, пока парикмахер дострижет клиентов

{

}

return 0;

}

cout <<"Введите что - нибудь, что бы призвать новых клиентов" << endl;

cin >> a;

cout << "ID потока = " << this\_thread::get\_id() << "\tПришел новый клиент под номером " << counter + 1 << endl;

counter++;

if (IsBarberBusy)

{

Wait();

}

else

{

thread th(DoWork);

th.detach();

}

}

return 0;

}