Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук

Пояснительная записка к микропроекту №2

По дисциплине

“Архитектура вычислительных систем”

На тему

“ Многопоточное программирование. Взаимодействие потоков”

Работу выполнил

Студент группы БПИ-194 (2 подгруппы) _____ М. Г. Савинов

подпись, дата

Работу проверил _____ А. И. Легалов

подпись, дата

Москва 2020

Содержание

Постановка задачи

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 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.

Описание задачи

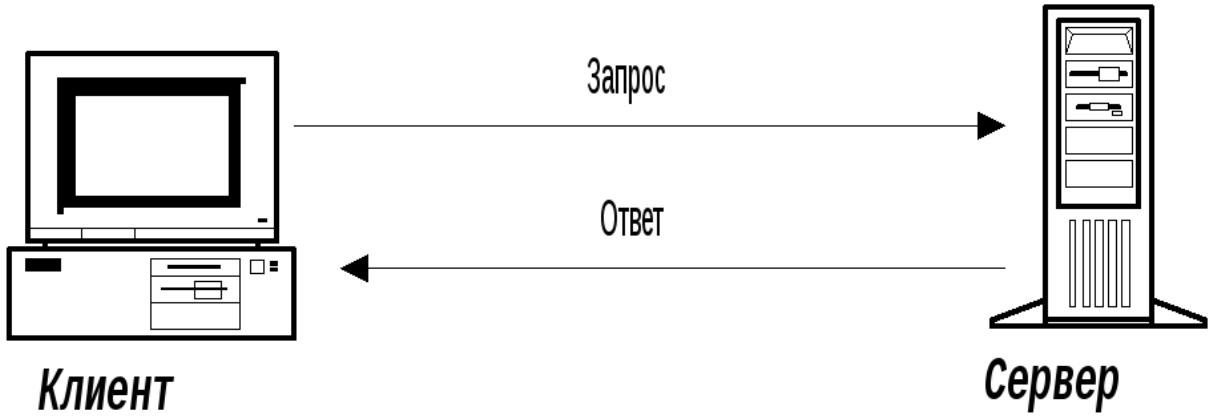
1) Необходимо разработать и отладить многопоточную программу с использованием семафоров и(или) условных переменных в соответствии с выданным заданием.
    a. Для заданной программы была использована стандартная библиотека C++, библиотека OpenMP для запуска потоков клиентов и парадигма «сервер-клиент».
    b. Создадим 2 класса (сервер и клиент).
    c. Класс сервер Barber имеет поле – очередь из клиентов clients, если очередь пуста, он спит, если очередь не пуста, то он переходит к клиенту с номером, указанным в приватном поле clientsCount.
    d. Класс клиент Client, спит пока до него не дойдет очередь.
    e. Для симуляции действий парикмахера и клиентов, были использованы семафоры, barberSemaphore, заставляет поток парикмахера ожидать пока i-ый клиент сядет в кресло. Также для каждого потока-клиента есть собственный семафор, хранящийся в векторе semaphors, который заставляет i-ого клиента спать пока до него не дошла очередь или пока его стрижет парикмахер.
    f. Стрижка происходит случайное время, которое вычисляется по формуле: случайное число из диапазона от 5 до 50 умножить на 100ms. Именно такие константы выбраны исключительно для наглядности работы программы, при желании их можно изменить.
    g. Первоначально ассинхронно запускает поток сервера, а затем потоки-клиенты вызываются с использованием библиотеки OpenMP, а именно omp for, с задержкой для i-ого потока равной i * 2500ms (константа опять же выбрана для наглядности работы программы и может быть изменена). Если очередь пустая, то i-ый клиент будит парикмахера, когда добавляется в очередь clients.
    h. Генерация количества клиентов в рабочий день выполняется с помощью использования заголовочного файла random на основе текущего времени. Для удобства использования программы выставлен диапазон целых чисел от 5 до 50, однако его можно менять, что не повлияет на работоспособность программы.
2) Для выполнения программы не требуется входных данных, однако в коде программы можно изменить границы генерации чисел.
3) Программа выдает действия парикмахера и клиентов и успешно завершается с кодом 0.

Пояснение задачи

Как уже было сказано выше для решения задачи использована парадигма «клиент-сервер».

«Клиент — сервер» — вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Фактически клиент и сервер — это программное обеспечение. Обычно эти программы расположены на разных вычислительных машинах и взаимодействуют между собой через вычислительную сеть посредством сетевых протоколов, но они могут быть расположены также и на одной машине. Программы-серверы ожидают от клиентских программ запросы и предоставляют им свои ресурсы в виде данных.

Можно сказать, что это способ взаимодействия неравноправных потоков. Клиентский поток запрашивает сервер и ждет ответа. Серверный поток ожидает запроса от клиента, затем действует в соответствии с поступившим запросом.



Пример работы программы

Сгенерированное число клиентов 33. Программа выводит приветственные слова и начинает работу:

```
Сегодня к парикмахеру записались 33 человек

Парикмахер пришел на работу

Посетителей нет, можно поспать

Клиент 1 будит парикмахера

Парикмахер приступает к клиенту 1
Клиент 1 сел в кресло
Парикмахер стрижет клиента 1
Стрижка окончена, до свидания, людей в очереди: 0
Посетителей нет, можно поспать
```

Заметим, что так как количество время на стрижку случайное, то парикмахер может опять уснуть во время работы, как например после 6 клиента, тогда 7 клиент будит нашего парикмахера:

```
Парикмахер стрижет клиента 2
Стрижка окончена, до свидания, людей в очереди: 1

Парикмахер приступает к клиенту 3
Клиент 3 сел в кресло
Парикмахер стрижет клиента 3
Стрижка окончена, до свидания, людей в очереди: 2

Парикмахер приступает к клиенту 4
Клиент 4 сел в кресло
Парикмахер стрижет клиента 4
Стрижка окончена, до свидания, людей в очереди: 1

Парикмахер приступает к клиенту 5
Клиент 5 сел в кресло
Парикмахер стрижет клиента 5
Стрижка окончена, до свидания, людей в очереди: 1

Парикмахер приступает к клиенту 6
Клиент 6 сел в кресло
Парикмахер стрижет клиента 6
Стрижка окончена, до свидания, людей в очереди: 0
Посетителей нет, можно поспать

Клиент 7 будит парикмахера

Парикмахер приступает к клиенту 7
Клиент 7 сел в кресло
Парикмахер стрижет клиента 7
```

После этого программа успешно продолжает работу:

```
Парикмахер приступает к клиенту 8
Клиент 8 сел в кресло
Парикмахер стрижет клиента 8
Стрижка окончена, до свидания, людей в очереди: 3

Парикмахер приступает к клиенту 9
Клиент 9 сел в кресло
Парикмахер стрижет клиента 9
Стрижка окончена, до свидания, людей в очереди: 3

Парикмахер приступает к клиенту 10
Клиент 10 сел в кресло
Парикмахер стрижет клиента 10
Стрижка окончена, до свидания, людей в очереди: 3

Парикмахер приступает к клиенту 11
Клиент 11 сел в кресло
Парикмахер стрижет клиента 11
Стрижка окончена, до свидания, людей в очереди: 3

Парикмахер приступает к клиенту 12
Клиент 12 сел в кресло
Парикмахер стрижет клиента 12
Стрижка окончена, до свидания, людей в очереди: 4

Парикмахер приступает к клиенту 13
Клиент 13 сел в кресло
Парикмахер стрижет клиента 13
```

Программа также успешно завершилось с кодом 0:

```
Клиент 29 сел в кресло
Парикмахер стрижет клиента 29
Стрижка окончена, до свидания, людей в очереди: 4

Парикмахер приступает к клиенту 30
Клиент 30 сел в кресло
Парикмахер стрижет клиента 30
Стрижка окончена, до свидания, людей в очереди: 3

Парикмахер приступает к клиенту 31
Клиент 31 сел в кресло
Парикмахер стрижет клиента 31
Стрижка окончена, до свидания, людей в очереди: 2

Парикмахер приступает к клиенту 32
Клиент 32 сел в кресло
Парикмахер стрижет клиента 32
Стрижка окончена, до свидания, людей в очереди: 1

Парикмахер приступает к клиенту 33
Клиент 33 сел в кресло
Парикмахер стрижет клиента 33
Стрижка окончена, до свидания, людей в очереди: 0

Рабочий день окончен
C:\Users\2001_\source\repos\Barber\Debug\Barber.exe (процесс 6312) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "А
томатически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно…
```

Список использованной литературы

1. Легалов А.И.(2020) «Архитектура параллельных вычислительных систем. Многопоточность» (http://softcraft.ru/edu/comparch/lect/07-parthread/).
2. Легалов А.И.(2020) «Многопоточное программирование. OpenMP» (http://www.softcraft.ru/edu/comparch/practice/thread/03-openmp/)
3. Microsoft docs «Директивы OpenMP» (https://docs.microsoft.com/ru-ru/cpp/parallel/openmp/reference/openmp-directives?view=msvc-160#sections-openmp)
4. Wikipedia «Клиент-сервер».