**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

Дисциплина: «Архитектура вычислительных систем»

**Микропроект № 2**

**Тема: «Решение задачи о читателях и писателях с использованием семафоров»**

**Вариант 3**

Пояснительная записка

**Выполнила:** Резуник Людмила  
*студент гр. БПИ198*

**Москва**2020

Содержание

[1. Текст задания 2](#_Toc58787580)

[2. Постановка задачи. 2](#_Toc58787581)

[3. Описание работы программы 2](#_Toc58787582)

[3.1. Работа с потоками 2](#_Toc58787583)

[3.2. Работа потоков с базой данных 3](#_Toc58787584)

[4. Описание входных и выходных данных 3](#_Toc58787585)

[5. Тестирование программы 3](#_Toc58787586)

[Список использованных источников 5](#_Toc58787587)

1. **Текст задания**

**Вариант 3**

Задача о читателях и писателях. Базу данных разделяют два типа процессов – читатели и писатели. Читатели выполняют транзакции, которые просматривают записи базы данных, транзакции писателей и просматривают и изменяют записи. Предполагается, что в начале БД находится в непротиворечивом состоянии (т.е. отношения между данными имеют смысл). Каждая отдельная транзакция переводит БД из одного непротиворечивого состояния в другое. Для предотвращения взаимного влияния транзакций процесс-писатель должен иметь исключительный доступ к БД. Если к БД не обращается ни один из процессов-писателей, то выполнять транзакции могут одновременно сколько угодно читателей. Создать многопоточное приложение с потоками-писателями и потоками-читателями. Реализовать решение, используя семафоры.

1. **Постановка задачи.**

Разработать консольную программу, которая моделирует взаимодействие потоков читателей и писателей с базой данных, используя семафоры.

1. **Описание работы программы**

Логику программы можно разделить на несколько частей. Первая часть – работа с потоками, а вторая часть – работа потоков с базой данных.

* 1. **Работа с потоками**

На начальном этапе происходит инициализация семафора, а также 10 потоков читателей и писателей. Далее, каждый поток в цикле начинает работу параллельно другим.

За реализацию потоков-читателей отвечает метод reader, в котором поток захватывает блокировку исключительного доступа у семафора, взаимодействует с базой данных (описывается в следующем пункте) и при выходе освобождает блокировку для следующих потоков.

Потоки-писатели работают схожим образом. Сначала они захватывают блокировку в методе writer, работают с базой и после освобождают блокировку.

* 1. **Работа потоков с базой данных**

В качестве модели для базы данных был использован класс Data и методы add, remove и read. Данные хранятся в векторе строк.

Потоки-читатели для взаимодействия с базой используют только метод read, который осуществляет вывод всех строк, хранящихся в векторе.

Потоки-писатели добавляют новые строки в базу данных используя метод add, который принимает на вход строку, которую нужно добавить. Строки генерируются рандомно. Помимо этого, потоки-писатели рандомно удаляют записи из базы. В процессе работы потоков на экран выводятся соответствующие сообщения.

Стоит отметить, что для моделирования работы с базой данных применялся метод sleep (длительностью в 3 секунды), чтобы замедлить выполнение потока.

1. **Описание входных и выходных данных**

Ввод данных пользователем не запрашивается, количество генерируемых потоков каждого типа задано в коде программы (10 потоков-читателей и столько же писателей).

Выходные данные выводятся на консоль и представляют собой строки с информацией о работе каждого потока, получаемые в рантайме.

1. **Тестирование программы**

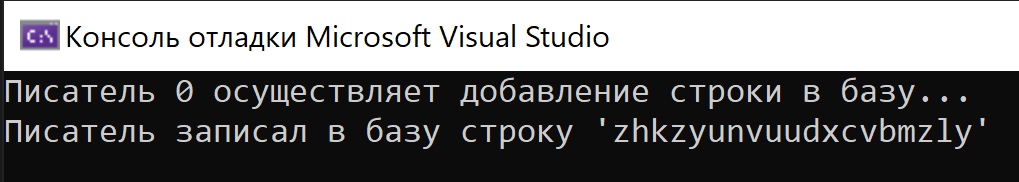
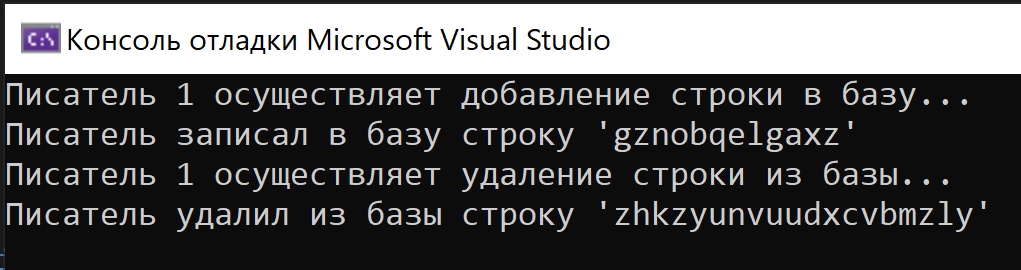
Далее будут представлены скриншоты работы программы.

Рисунок 1 – Вывод информации о работе потока-писателя (только запись в базу)

  
Рисунок 2 – Вывод информации о работе потока-писателя (удаление и запись)

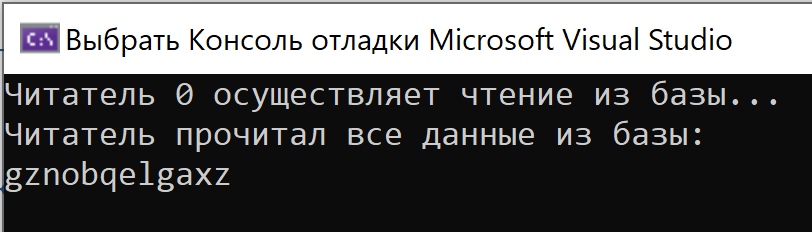


Рисунок 3 – Вывод информации о работе потока-читателя

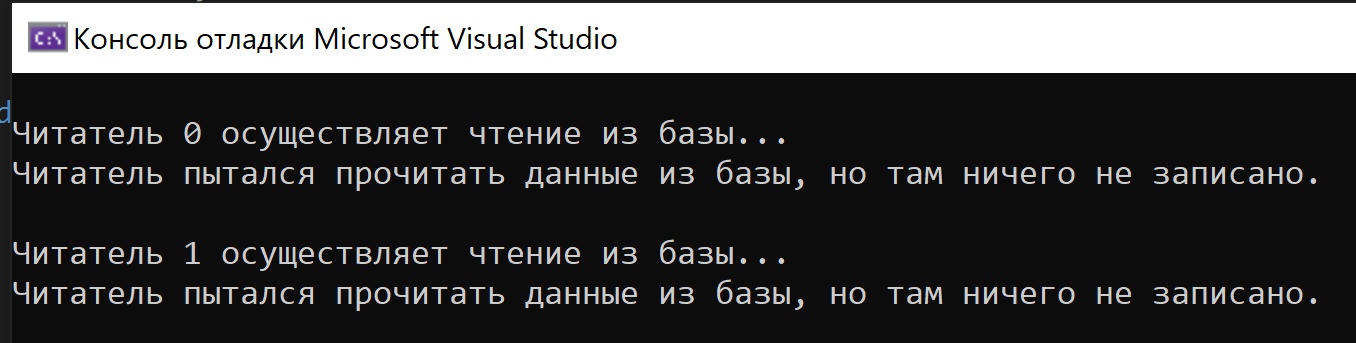


Рисунок 4 – Вывод информации о работе читателя в случае пустой базы данных

ПРИЛОЖЕНИЕ 1

Список использованных источников

1. Глава 4. Семафоры [Электронный ресурс] // URL: <https://l.wzm.me/_coder/custom/parallel.programming/004.htm> (дата обращения: 12.12.2020)
2. Многопоточность в C++ с использованием pthreads [Электронный ресурс] // URL: <https://eax.me/pthreads/> (дата обращения: 13.12.2020)