**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ   
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
|  |  |
|  | **Микропроект № 1**  **по дисциплине**  **"Архитектура вычислительных систем"**  **Пояснительная записка**  Исполнитель:  Студент группы БПИ191  / Власюк А. В./  «13» декабря 2020 г.  **Москва 2020** | |  |

Содержание

[1. ВВЕДЕНИЕ 6](#_Toc58794061)

[1.1. Наименование программы 6](#_Toc58794062)

[1.2. Документы, на основании которых ведётся разработка 6](#_Toc58794063)

[1.3. Описание программы 6](#_Toc58794064)

[2. ПРИНЦИПЫ РЕАЛИЗАЦИИ 7](#_Toc58794065)

[2.1. Многопоточность 7](#_Toc58794066)

[2.1.1. Читатель 7](#_Toc58794067)

[2.1.2. Писатель 7](#_Toc58794068)

[2.2. База данных 7](#_Toc58794069)

[2.3. Описание переменных 8](#_Toc58794070)

[2.4. Описание методов 8](#_Toc58794071)

[2.4. Входные данные 9](#_Toc58794072)

[2.5. Выходные данные 9](#_Toc58794073)

[3. ТЕСТИРОВАНИЕ 10](#_Toc58794074)

[3.1. Тест-1 10](#_Toc58794075)

[3.2. Тест-2 11](#_Toc58794076)

[3.3. Тест-3 12](#_Toc58794077)

[4. СПИСОК ИСТОЧНИКОВ 13](#_Toc58794078)

1. **ВВЕДЕНИЕ**

1.1. **Наименование программы**

Наименование программы – «8. Задача о читателях и писателях-2 (грязное чтение)».

* 1. Документы, на основании которых ведётся разработка

Программа выполнена в рамках задания по дисциплине "Архитектура вычислительных систем", в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия», 2 курс 2 модуль.

Основанием для разработки является письмо профессора факультета компьютерных наук Легалова Александра Ивановича от 28.11.20

* 1. Описание программы

Базу данных разделяют два типа потоков – 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). Транзакции выполняются в режиме «грязного чтения», то есть процесс-писатель не может получить доступ к БД только в том случае, если ее занял другой процесс-писатель, а процессы-читатели ему не мешают. Создать многопоточное приложение с потоками-писателями и потоками-читателями. Реализовать решение, используя семафоры, и не используя блокировки чтения-записи.

.

1. ПРИНЦИПЫ РЕАЛИЗАЦИИ

Данная программа представляет собой упрощённую версию задачи о писателях и читателях – регулируются только действия читателей.

2.1. Многопоточность

В программе реализовано 2 типа потоков: писатели и читатели. Они работают с одной базой данных, при этом в один момент с ней может работать сколько угодно читателей, но лишь один писатель.

Для синхронизации работы писателей с базой данных используется семафор с начальным значением 1.

Многопоточность программы реализована с использованием библиотеки pthread.

2.1.1. Читатель

Читатель выбирает ячейку и считывает её значение, после чего выводит его в консоль. Одновременно с данными могут работать сколько угодно читателей и они никак не мешают писателям, поэтому их работа не синхронизируется.

2.1.2. Писатель

После выбора ячейки базы данных писатель пытается получить доступ к базе данных, то есть уменьшает значение семафора. Таким образом, если в данный момент ни один из писателей не работает с данными, то данный писатель получает к ним доступ. После изменения данных, (а именно - писатель прибавляет в конец строки данных свой номер), писатель выводит изменения в консоль и увеличивает значение семафора - тогда доступ к базе данных откроется для следующего писателя. Таким образом, работа писателей образована по принципу очереди – FIFO.

* 1. База данных

База данных реализована в виде вектора строк. Изначально в каждой ячейке хранится строка

вида “dataIND”, где IND – индекс данной ячейки в векторе, начиная с 1.

2.3. Описание переменных

|  |  |  |  |
| --- | --- | --- | --- |
| Переменная | Тип | Значение | Смысл |
| N | unsigned | Натуральное число. | Кол-во итераций чтения/записи каждого потока. Вводится пользователем в коносль. |
| **database** | vector<string> | vector<string>(numOfDataCells) | База данных, с которой работают потоки. |
| **numOfDataCells** | unsigned | Натуральное число. | Кол-во ячеек базы данных. Вводится пользователем в консоль. |
| **write\_semaphore** | sem\_t | - | Семафор, синхронизирующий работу писателей. |
| **writers** | vector<pthread\_t> | vector<pthread\_t>(numOfThreads) | Вектор потоков-писателей. |
| **readers** | vector<pthread\_t> | vector<pthread\_t>(numOfThreads) | Вектор потоков-читателей. |
| **numOfThreads** | unsigned | Натуральное число. | Кол-во потоков читателей и писателей. |

2.4. Описание методов

|  |  |
| --- | --- |
| **Метод** | **Цель** |
| **reader** | Реализует работу потока-читателя |
| **writer** | Реализует работу потока-писателя. |
| **getUserInput** | Получает на ввод от пользователя натуральное число. Проверяет корректность ввода. |
| **main** | Инициализирует переменные, запускает работу потоков. |

2.4. Входные данные

В начале работы программы пользователь вводит три 32-битных положительных числа – numOfDataCells, numOfThreads, N (именно в таком порядке). Перед вводом каждого значения программа выводит в коносль сообщение с просьбой ввода, начинающееся как: “Input a number of …”. Если пользователь введет некорректное значение, программа завершается.

2.5. Выходные данные

В качестве выходных данных служат сообщения, выводимые потоками-писателями и -читателями.

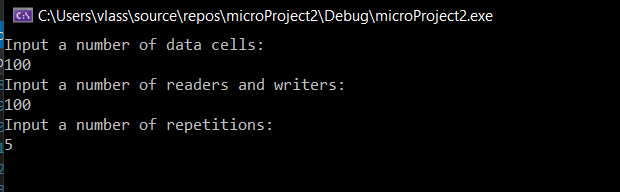
Читатель после каждого прочтения данных выводит сообщение вида: “reader №{1} reads data cell №{2}: ‘{3}’”, где {1} – номер данного читателя, {2} – номер читаемой ячейки базы данных, {3} – считанное значение.

Писатель, во время ожидания доступа к базе данных, выводит сообщение вида: “writer №{1} is trying to access the database…”, где {1} – номер данного писателя. После получения доступа, читатель изменяет данные и выводит сообщение: “writer №{1} has changed data cell №{2} from '{3}' to '{4}'”, где {1} – номер данного писателя, {2} – номер изменяемой им ячейки данных, {3} – значение ячейки данных до изменения, {4} – значение после изменения. Затем писатель предоставляет доступ к данным для следующего читателя в очереди и выводит сообщение “writer №{1} has finished working with the data”, где {1} – номер данного писателя.

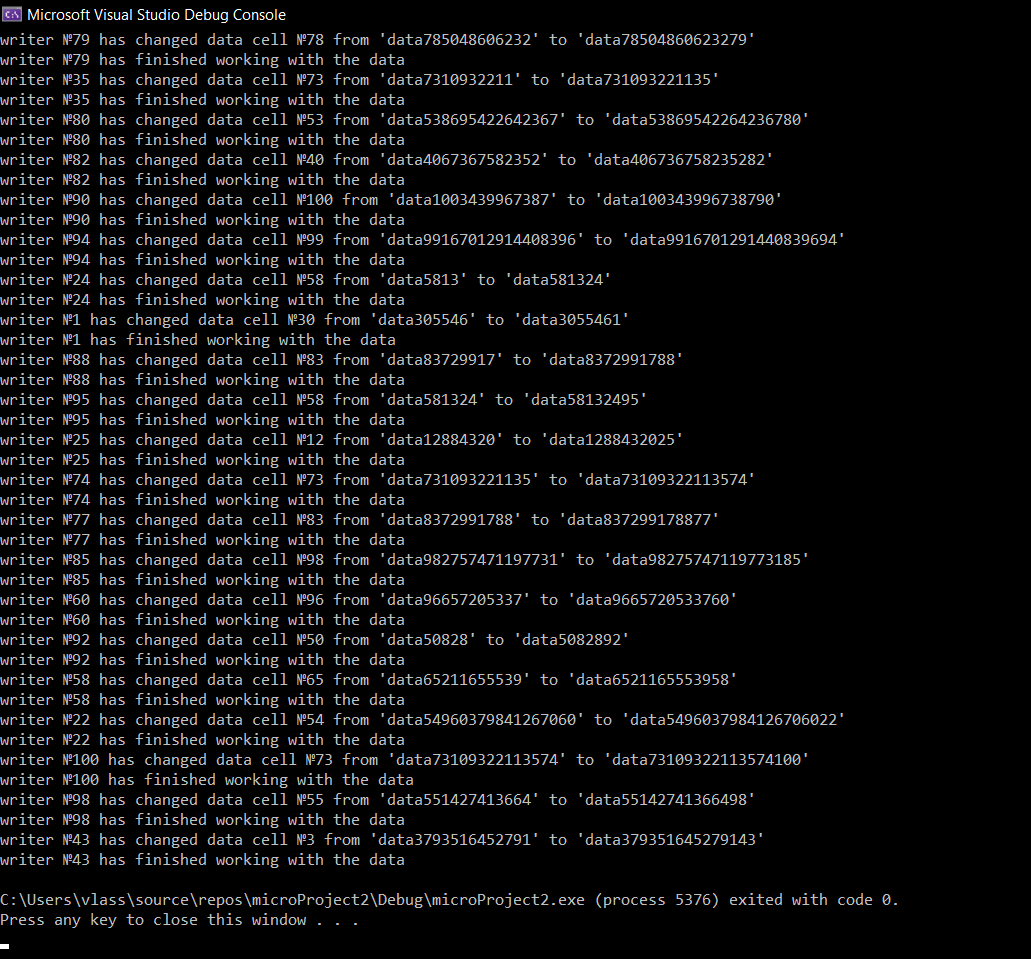
1. ТЕСТИРОВАНИЕ

3.1. Тест-1

Зададим в качестве входных данных 100 ячеек базы данных, 100 читателей и писателей и кол-во итераций каждого читателя и писателя = 5:

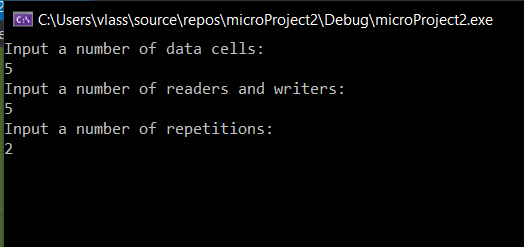


Программа успешно отрабатывает:

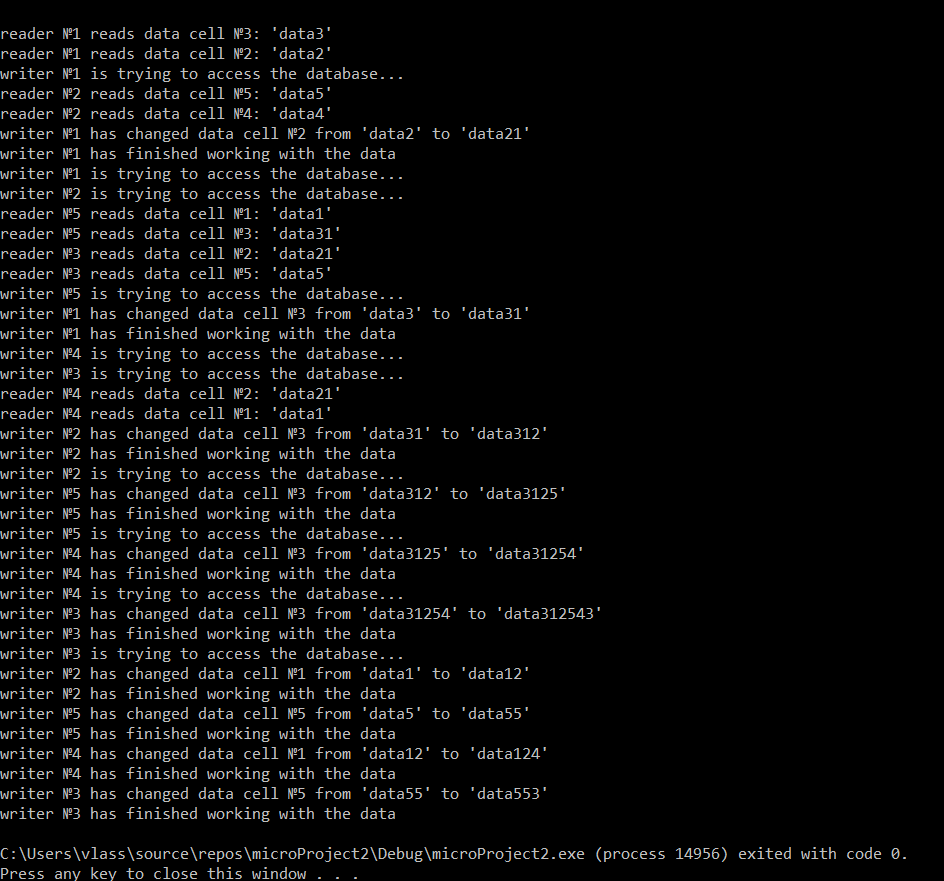


* 1. Тест-2

Сделаем тест попроще: 5 ячеек базы данных, 5 читателей и писателей, по 2 итерации.

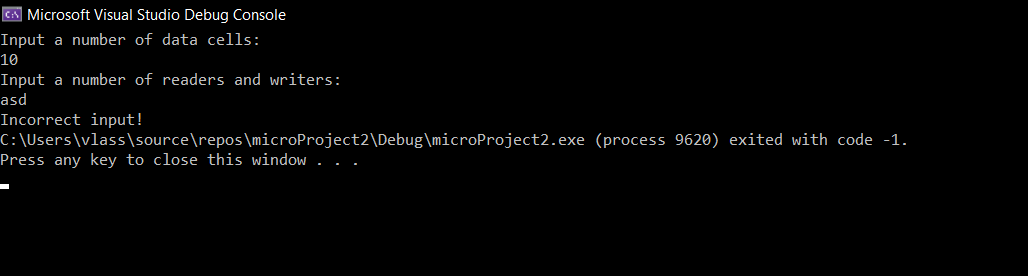


Вывод программы:

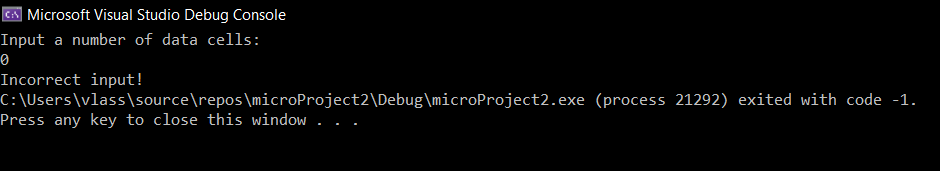


* 1. Тест-3

Протестируем программу на некорректных входных данных, попытаемся ввести строку вместо числа:



Или неположительное число:



Как видно, программа в обоих случаях выводит сообщение о некорректном вводе и завершает работу.

1. СПИСОК ИСТОЧНИКОВ
2. [Примеры использования библиотеки pthread](https://eax.me/pthreads/)
3. [Википедия, статья о семафорах](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D1%84%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)#:~:text=%D0%A1%D0%B5%D0%BC%D0%B0%D1%84%D0%BE%CC%81%D1%80%20(%D0%B0%D0%BD%D0%B3%D0%BB.,%D0%BD%D1%83%D0%BB%D0%B5%D0%B2%D0%BE%D0%B3%D0%BE%20%D0%B7%D0%BD%D0%B0%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%20%D1%81%D1%87%D1%91%D1%82%D1%87%D0%B8%D0%BA%D0%B0%20%D1%8F%D0%B2%D0%BB%D1%8F%D0%B5%D1%82%D1%81%D1%8F%20%D0%B1%D0%BB%D0%BE%D0%BA%D0%B8%D1%80%D1%83%D1%8E%D1%89%D0%B5%D0%B9%D1%81%D1%8F.)
4. [Википедия, статья о задаче о читателях-писателях](https://en.wikipedia.org/wiki/Readers%E2%80%93writers_problem)
5. [SoftCraft, страница с условием задания](http://softcraft.ru/edu/comparch/tasks/mp02/)
6. [Документация msvc](https://docs.microsoft.com/ru-ru/cpp/?view=msvc-160)