

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

Факультет компьютерных наук  
Департамент программной инженерии

**ЗАДАЧА О ЧИТАТЕЛЯХ И ПИСАТЕЛЯХ**

**Пояснительная записка**

**Листов 12**

Исполнитель:

студентка группы БПИ196

Баранова Анастасия Андреевна

## СОДЕРЖАНИЕ

|  |           |
|--|-----------|
| <b>1. ПОСТАНОВКА ЗАДАЧИ.....</b>                           | <b>3</b>  |
| <b>2. СЦЕНАРИЙ ВЗАИМОДЕЙСТВИЯ ОБЪЕКТОВ ПРОГРАММЫ .....</b> | <b>4</b>  |
| <b>2.1. Общие сведения .....</b>                           | <b>4</b>  |
| <b>2.2. Сценарий работы читателя .....</b>                 | <b>4</b>  |
| <b>2.3. Сценарий работы писателя .....</b>                 | <b>4</b>  |
| <b>2.4. Сценарий взаимодействия объектов .....</b>         | <b>5</b>  |
| <b>3. ПРОТОКОЛ ВЗАИМОДЕЙСТВИЯ СУБЪЕКТОВ .....</b>          | <b>9</b>  |
| <b>4. ЗАПУСК ПРОГРАММЫ.....</b>                            | <b>11</b> |
| <b>5. СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....</b>              | <b>12</b> |

## 1. ПОСТАНОВКА ЗАДАЧИ

### 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.

## **2. СЦЕНАРИЙ ВЗАИМОДЕЙСТВИЯ ОБЪЕКТОВ ПРОГРАММЫ**

### **2.1. Общие сведения**

В начале работы программы создаются 4 потока-писателя и 4 потока-читателя, которые работают вместе (параллельно) с одной «базой данных» - массивом из 10 целых чисел (далее БД). Писатели выполняют запись в массив, читатели – читают данные из массива. Каждый писатель и читатель выполняет соответствующую ему операцию определенное количество раз – задается пользователем в аргументах командной строки при запуске (см. п. 3. Запуск программы). Программа завершается, когда каждый из потоков завершает свою работу.

### **2.2. Сценарий работы читателя**

#### **2.2.1. Начало работы**

Поток-читатель, приступая к работе, запрашивает доступ на чтение (может быть закрыт потоком-писателем или временно заблокирован другим читателем). Получая разрешение на чтение, читатель ожидает доступ к переменной-счетчику количества читателей, получив который увеличивает счетчик. Если данный читатель первый – он блокирует доступ к БД для записи. Далее читатель отдает доступ к счетчику и доступ на чтение другим потокам.

#### **2.2.2. Чтение**

Читатель выбирает случайную ячейку в БД и читает значение из выбранной ячейки.

Чтение происходит в критической секции. Одновременно выполнять чтение могут несколько читателей, однако если хотя бы один читатель работает с БД, ни один писатель не может получить к ней доступ.

#### **2.2.3. Окончание работы**

Перед завершением работы читатель должен уменьшить счетчик читателей. Для этого читатель ожидает доступ к счетчику, после чего уменьшает его. Если данный читатель был последним, он разблокирует доступ на запись. Читатель отдает доступ к счетчику.

### **2.3. Сценарий работы писателя**

#### **2.3.1. Начало работы**

Перед началом работы писателю необходимо увеличить счетчик писателей. Для этого писатель ожидает доступ к переменной-счетчику, после чего увеличивает его. Если данный писатель первый, он ожидает окончания работы читателей, которые уже начали выполнение своей работы, и блокирует доступ на чтение (пока писатель не завершит работу, новые читатели не смогут получить доступ к БД). Писатель отдает доступ к переменной-счетчику.

### **2.3.2. Запись**

Для произведения записи в БД писатель ожидает доступ к БД для записи (может быть занят другим писателем или заблокирован работающими читателями). Получив доступ, писатель выбирает случайную ячейку в БД и увеличивает значение в данной ячейке на единицу. Произведя запись, писатель отдает доступ для записи.

Запись происходит в критической секции. Одновременно выполнять запись может только один писатель, в этот момент ни один другой поток (ни читатель, ни писатель) не имеет доступа к БД.

### **2.3.3. Окончание работы**

Перед завершением работы писатель должен уменьшить счетчик писателей. Для этого поток-писатель ожидает доступ к счетчику, после чего уменьшает его. Если данный писатель был последним, он разблокирует доступ на чтение. Писатель отдает доступ к счетчику.

## **2.4. Сценарий взаимодействия объектов**

Одновременно к БД могут иметь доступ либо единственный писатель, либо сколь угодно много читателей.

Читатель не может начать работу, пока не получит разрешение на чтение. Доступ к БД для чтения может быть закрыт работающим писателем или временно заблокирован другим читателем. В первом случае читатель ожидает окончания работы всех писателей (когда последний писатель закончит работу, он откроет доступ к БД для читателей). Во втором случае читатель ожидает, когда другой читатель разблокирует доступ.

Пока происходит чтение ни один писатель не должен иметь доступа к БД, поэтому первый пришедший читатель, получив разрешение на чтение блокирует доступ к БД для записи. Доступ для писателей будет вновь открыт, когда последний читатель завершит работу.

Писатель не начинает запись, пока последний читатель не завершит свою работу. В начале своей работы писатель ожидает, когда освободится доступ на чтение и закрывает его. Это обеспечивает приоритетность писателей – когда появляется хотя бы один писатель, ни один новый читатель не может начать работу и находится в состоянии ожидания доступа. Доступ для читателей будет открыт, когда последний писатель завершит свою работу.

Далее приведены возможные состояния потоков писателей и читателей при их взаимодействии с другими потоками писателями и читателями, а также действия этих потоков в различных условиях, обусловленных параллельным выполнением других потоков.

## **2.4.1. Писатель**

### **2.4.1.1. Взаимодействие с писателями**

#### **2.4.1.1.1. Другой писатель выполняет работу со счетчиком:**

Ожидание доступа к счетчику от других писателей, доступ будет предоставлен, когда другой поток завершит работу с переменной.

#### **2.4.1.1.2. Работа с общей переменной-счетчиком:**

Для корректной работы программы, поток-писатель, получая доступ к счетчику блокирует его для других потоков, так что на время работы с переменной другой писатель не может обратиться к ней.

#### **2.4.1.1.3. Предоставление доступа к счетчику:**

По окончании работы со счетчиком поток-писатель отдает доступ к ней, после чего с переменной может работать другой поток.

#### **2.4.1.1.4. Другой писатель выполняет запись в БД:**

Ожидание доступа к БД на запись. Доступ будет получен, когда предыдущий поток завершит запись.

#### **2.4.1.1.5. Работа с общей БД:**

Писатель имеет исключительный доступ к БД, во время записи в БД одним потоком ни один другой не может получить к ней доступ. Поэтому перед тем, как начать запись писатель забирает доступ к БД для записи.

#### **2.4.1.1.6. Предоставление доступа к БД для записи:**

По окончании транзакции писатель отдает доступ к БД для записи, после чего другой писатель может начать транзакцию, захватив доступ.

### **2.4.1.2. Взаимодействие с читателями**

#### **2.4.1.2.1. Писатель начитает работу до завершения транзакций читателей:**

Ожидание окончания работы всех читателей, которые начали чтение на момент начала работы писателя.

#### **2.4.1.2.2. Блокировка доступа к БД на чтение:**

Писатель имеет исключительный доступ к БД. Во время работы писателя ни один читатель не может иметь доступ к БД. Поэтому первый пришедший писатель блокирует доступ к БД для читателей.

#### 2.4.1.2.3. Предоставление доступа к БД для чтения:

После окончания работы последнего писателя он разблокирует доступ к БД для читателей. После этого читатели могут возобновить чтение.

### 2.4.2. Читатель

#### 2.4.2.1. Взаимодействие с писателями

##### 2.4.2.1.1. Читатель начинает работу до завершения транзакций писателей:

Читатель ожидает открытия доступа к БД для чтения. Доступ будет получен после завершения работы последнего писателя.

##### 2.4.2.1.2. Блокировка доступа к БД на чтение:

После получения доступа к БД для чтения писатель вынужден временно блокировать доступ для других потоков. Когда читатель блокирует доступ к БД для чтения, он гарантированно закончит свою работу до писателей, которые придут после блокировки.

##### 2.4.2.1.3. Блокировка доступа к БД для записи:

Первый поток-читатель, начинающий работу, обладая блокировкой доступа для чтения, блокирует доступ к БД для записи, так что во время выполнения транзакций читателей ни один писатель не может получить доступ к БД.

##### 2.4.2.1.4. Предоставление доступа к БД для чтения:

Читатель открывает доступ к БД для чтения, после чего поток-писатель может занять его для выполнения транзакции (дождавшись окончания выполнения работы писателей).

##### 2.4.2.1.5. Предоставление доступа к БД для записи:

Последний поток-читатель, заканчивающий работу, открывает доступ к БД для записи, так что поток-писатель, ожидающий доступа к БД, может начать запись.

#### 2.4.2.2. Взаимодействие с читателями

##### 2.4.2.2.1. Другой читатель заблокировал доступ для чтения:

Ожидание доступа к БД для чтения. Доступ будет предоставлен, когда другой поток-читатель перейдет к непосредственно чтению.

##### 2.4.2.2.2. Блокировка доступа к БД для чтения:

Для корректной работы программы, приступая к работе, поток-читатель вынужден временно блокировать доступ к БД для чтения для других потоков (см. п.

3.2.1.2.), что сказывается на работе других потоков-читателей, которым приходится ожидать доступа для начала работы.

2.4.2.2.3. Другой читатель выполняет работу со счетчиком:

Ожидание доступа к счетчику от читателя, доступ будет предоставлен, когда другой поток завершит работу с переменной.

2.4.2.2.4. Работа с общей переменной-счетчиком:

Для корректной работы программы, поток-читатель, получая доступ к счетчику блокирует его для других потоков, так что на время работы с переменной другой читатель не может обратиться к ней.

2.4.2.2.5. Предоставление доступа к счетчику:

По окончании работы со счетчиком поток-читатель отдает доступ к ней, после чего с переменной может работать другой поток.

2.4.2.2.6. Работа с общей БД:

Одновременно чтение могут выполнять несколько потоков-читателей.



### 3. ПРОТОКОЛ ВЗАИМОДЕЙСТВИЯ СУБЪЕКТОВ

Сообщения обо всех действиях читателей и писателей выводятся в выходной поток. Программа выводит информацию в виде протокола происходящих событий, однозначно определяя номер выполняемого потока и происходящее событие. Сообщается информация о том, когда писатель или читатель приступает к работе и когда ее завершает, о результатах записи (что и куда было записано) или чтения (что и откуда было прочитано), а также событиях, влияющих на работу других потоков или связанных с ней (блокировках и ожиданиях доступа). Каждое сообщение содержит отметку времени, когда произошло событие, информацию о номере потока и его типе (писатель или читатель), а также само сообщение о событии.

#### 3.1. Пример протокола работы читателя:

Time 23. Reader 1 started.  
Time 23. Reader 1 is waiting for an access to read.  
Time 24. Reader 1 blocked an access for writers.  
Time 24. Reader 1 read value 0 at the 1 element of a database.  
Time 24. Reader 1 released an access for writers.  
Time 24. Reader 1 finished their work.

#### 3.2. Пример протокола работы писателя:

Time 47. Writer 1 started.  
Time 48. Writer 1 is waiting for readers to finish.  
Time 48. Writer 1 blocked an access for readers.  
Time 48. Writer 1 is waiting for an access to the database.  
Time 48. Writer 1 wrote 1 to the 1 element of a database.  
Time 49. Writer 1 released an access for readers.  
Time 49. Writer 1 finished their work.

#### 3.3. Пример протокола совместной работы читателя и писателя:

Time 22. Writer 1 started.  
Time 22. Writer 1 is waiting for readers to finish.  
Time 22. Reader 1 started.  
Time 23. Writer 1 blocked an access for readers.  
Time 23. Reader 1 is waiting for an access to read.  
Time 23. Writer 1 is waiting for an access to the database.  
Time 23. Writer 1 wrote 1 to the 1 element of a database.  
Time 23. Writer 1 released an access for readers.

Time 23. Writer 1 finished their work.

Time 23. Reader 1 blocked an access for writers.

Time 24. Reader 1 read value 1 at the 1 element of a database.

Time 24. Reader 1 released an access for writers.

Time 24. Reader 1 finished their work.

#### 4. ЗАПУСК ПРОГРАММЫ.

Для запуска приложения используется командная строка.

Аргументом командной строки является единственное число – количество итераций при работе каждого потока читателя и писателя. Вводимое число должно быть целым и положительным.

Из директории, содержащей исполняемый файл программы с именем 02microproject, запустить программу можно, используя команду **02microproject n**. Здесь на месте n должно быть указано некоторое целое положительное число.

Стоит также обратить внимание, что вывод протокола работы потоков может быть достаточно объемным, поэтому рекомендуется перенаправлять вывод из консоли в файл для изучения результатов после окончания работы программы. Например, для вывода протокола в файл protocol.txt можно запустить программу командой **02microproject n > protocol.txt**.

## 5. СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1) Jeff Preshing, Semaphores are Surprisingly Versatile: [Электронный ресурс]: Режим доступа: URL: <https://preshing.com/20150316/semaphores-are-surprisingly-versatile>, свободный. (дата обращения: 13.12.2020).
- 2) Oracle, Multithreaded Programming Guide, Chapter 4 Programming with Synchronization Objects, Semaphores: [Электронный ресурс]: Режим доступа: URL: <https://docs.oracle.com/cd/E19455-01/806-5257/6je9h032s/index.html>, свободный. (дата обращения: 13.12.2020).
- 3) Oracle, Multithreaded Programming Guide, The Pthreads Library: [Электронный ресурс]: Режим доступа: URL: [https://docs.oracle.com/cd/E26502\\_01/html/E35303/tlib-1.html](https://docs.oracle.com/cd/E26502_01/html/E35303/tlib-1.html), свободный. (дата обращения: 13.12.2020).
- 4) Reader Writer Problem using Semaphore: [Электронный ресурс]: Режим доступа: URL: <https://www.youtube.com/watch?v=qbSVrkRFPOk>, свободный. (дата обращения: 13.12.2020).
- 5) Readers-Writers Problem | Set 1 (Introduction and Readers Preference Solution): [Электронный ресурс]: Режим доступа: URL: <https://www.geeksforgeeks.org/readers-writers-problem-set-1-introduction-and-readers-preference-solution/>, свободный. (дата обращения: 13.12.2020).
- 6) Архитектуры параллельных ВС (многопоточность): [Электронный ресурс]: Режим доступа: URL: <http://softcraft.ru/edu/comparch/lect/07-parthread/multitreading.pdf>, свободный. (дата обращения: 13.12.2020).
- 7) Википедия, Задача о читателях-писателях: [Электронный ресурс]: Режим доступа: URL: [https://ru.wikipedia.org/wiki/Задача\\_о\\_читателях-писателях](https://ru.wikipedia.org/wiki/Задача_о_читателях-писателях), свободный. (дата обращения: 13.12.2020).
- 8) Многопоточное программирование. Взаимодействие потоков. Микропроект. Требования к оформлению. 2020-2021 уч.г.: [Электронный ресурс]: Режим доступа: URL: <http://softcraft.ru/edu/comparch/tasks/mp02/>, свободный. (дата обращения: 13.12.2020).
- 9) НОУ ИНТУИТ, Лекция 3: Операционные системы - аспекты параллелизма, 3.6.2. Задача "Читатели-Писатели": [Электронный ресурс]: Режим доступа: URL: <https://intuit.ru/studies/courses/4447/983/lecture/14923?page=5>. (дата обращения: 13.12.2020).