

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

Факультет компьютерных наук  
Департамент программной инженерии

**МНОГОПОТОЧНОЕ ПРИЛОЖЕНИЕ ДЛЯ СИМУЛЯЦИИ  
РАБОТЫ БАЗЫ ДАННЫХ  
Пояснительная записка**

Исполнитель:  
студент группы БПИ199  
\_\_\_\_\_ К. Н. Борисов  
«\_\_\_\_\_» \_\_\_\_\_ 2020 г.

# Содержание

<b>1</b>	<b>Текст задания</b>	<b>3</b>
<b>2</b>	<b>Применяемые расчетные методы</b>	<b>4</b>
2.1	Структура БД	4
2.2	Вывод результата в консоль	4
2.3	Входные данные	4
2.4	Алгоритм взаимодействия потоков	4
<b>3</b>	<b>Тестовые примеры</b>	<b>5</b>
<b>4</b>	<b>Список использованной литературы</b>	<b>7</b>

## 1. Текст задания

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]

## 2. Применяемые расчетные методы

### 2.1. Структура БД

Из условия задачи сложно понять чем именно является база данных и как она должна выглядеть. Делать полноценную реляционную базу данных с таблицами для такой задачи будет очень сложно. Но и обычный список не пройдет, потому что у него не может быть противоречивых состояний и связей между объектами. Я решил использовать дерево, потому что его можно очень просто хранить: как массив индексов, где в каждом элементе хранится индекс его родителя. Если сделать ему фиксированный размер, то такое дерево будет сохраняться в файл в одну строчку: `fwrite(db, sizeof(int), DB_SIZE, file)`. Также у него могут быть противоречивые состояния, например, при отсутствии общего корневого узла или при наличии циклов.

### 2.2. Вывод результата в консоль

Для большой наглядности вывод в консоль использует `\x1b` чтобы изменять цвет текста [5]. Также в `printf`-е используется опция выравнивания чисел пробелами.

### 2.3. Входные данные

Делать ввод списка всех читателей и писателей тут не надо. Понять правильно ли работает симуляция можно и с 10 читателями и писателями, запущенными со случайными задержками.

### 2.4. Алгоритм взаимодействия потоков

В программе есть один мьютекс для блокировки писателей. Его также используют читатели, чтобы понять есть ли сейчас активный писатель. Также у нас в семафоре хранится количество активных читателей и, когда оно достигает нуля, мы будим всех писателей, которые могли ждать закрытия всех читателей, потому что писатели, перед тем как начать писать, проверяют количество читателей в этом семафоре и ждут изменений, если из не ноль.

### 3. Тестовые примеры

Программа проводит симуляцию и наглядно выводит всё происходящее. (см. рисунок 1)

```
cortan122@ThinkPad-T480: /mnt/c/Users/user/OneDrive/dz2020/asm/mp02
cortan122@ThinkPad-T480:/mnt/c/Users/user/OneDrive/dz2020/asm/mp02$ ./db
Время 3ms: Писатель 9 приступил к замене 49714ого и 51625ого листа
Время 3ms: Писатель 9 поменял 49714ого и 51625ого листы местами
Время 4ms: Читатель 3 приступил к подсчету количества детей 15393ого листа
Время 4ms: Читатель 3 понял что у 15393ого листа 3 детей
Время 4ms: Писатель 3 приступил к замене 56303ого и 2897ого листа
Время 5ms: Писатель 3 поменял 56303ого и 2897ого листы местами
Время 5ms: Писатель 8 приступил к замене 5806ого и 5162ого листа
Время 6ms: Писатель 8 поменял 5806ого и 5162ого листы местами
Время 6ms: Писатель 1 приступил к замене 9702ого и 73267ого листа
Время 6ms: Писатель 1 поменял 9702ого и 73267ого листы местами
Время 7ms: Читатель 9 приступил к подсчету количества детей 33683ого листа
Время 7ms: Читатель 9 понял что у 33683ого листа 1 детей
Время 7ms: Писатель 7 приступил к замене 21100ого и 17151ого листа
Время 8ms: Писатель 7 поменял 21100ого и 17151ого листы местами
Время 8ms: Читатель 5 приступил к подсчету количества детей 39749ого листа
Время 8ms: Читатель 10 приступил к подсчету количества детей 10548ого листа
Время 8ms: Читатель 5 понял что у 39749ого листа 1 детей
Время 8ms: Читатель 8 приступил к подсчету количества детей 2544ого листа
Время 8ms: Читатель 10 понял что у 10548ого листа 3 детей
Время 10ms: Читатель 8 понял что у 2544ого листа 7 детей
Время 17ms: Писатель 2 приступил к замене 52678ого и 39815ого листа
Время 23ms: Писатель 2 поменял 52678ого и 39815ого листы местами
Время 24ms: Писатель 5 приступил к замене 24777ого и 43578ого листа
Время 25ms: Писатель 5 поменял 24777ого и 43578ого листы местами
Время 26ms: Писатель 10 приступил к замене 65433ого и 12000ого листа
Время 29ms: Писатель 10 поменял 65433ого и 12000ого листы местами
Время 30ms: Читатель 1 приступил к подсчету количества детей 6513ого листа
Время 35ms: Читатель 1 понял что у 6513ого листа 5 детей
Время 36ms: Писатель 6 приступил к замене 6836ого и 62ого листа
Время 39ms: Писатель 6 поменял 6836ого и 62ого листы местами
Время 39ms: Писатель 4 приступил к замене 10947ого и 47525ого листа
Время 40ms: Писатель 4 поменял 10947ого и 47525ого листы местами
Время 45ms: Читатель 4 приступил к подсчету количества детей 7957ого листа
Время 45ms: Читатель 6 приступил к подсчету количества детей 63056ого листа
Время 45ms: Читатель 2 приступил к подсчету количества детей 6030ого листа
Время 45ms: Читатель 7 приступил к подсчету количества детей 78683ого листа
Время 46ms: Читатель 4 понял что у 7957ого листа 3 детей
Время 51ms: Читатель 6 понял что у 63056ого листа 3 детей
Время 52ms: Читатель 2 понял что у 6030ого листа 3 детей
Время 54ms: Читатель 7 понял что у 78683ого листа 2 детей
```

Рисунок 1: Цветной вывод

Программа работает с базой данных и меняет её на диске. (см. рисунок 2)

```
cortan122@ThinkPad-T480:/mnt/c/Users/user/OneDrive/dz2020/asm/mp02$ sha1sum db.bin
b2e2bf90a280b94367ede2bb3da38ee0f99f9a6b  db.bin
cortan122@ThinkPad-T480:/mnt/c/Users/user/OneDrive/dz2020/asm/mp02$ ./db >/dev/null
cortan122@ThinkPad-T480:/mnt/c/Users/user/OneDrive/dz2020/asm/mp02$ sha1sum db.bin
2d9c37191dc0b20a1633247c8b350b823519cb9e  db.bin
cortan122@ThinkPad-T480:/mnt/c/Users/user/OneDrive/dz2020/asm/mp02$ |
```

Рисунок 2: База данных меняется на диске

## 4. Список использованной литературы

- [1] Практические приемы построения многопоточных приложений [Электронный ресурс]. //URL: <http://softcraft.ru/edu/comparch/tasks/mp02/> (Дата обращения: 13.12.2020, режим доступа: свободный)
- [2] pthreads(7) Linux User's Manual [Электронный ресурс]. //URL: <https://man7.org/linux/man-pages/man7/pthreads.7.html> (Дата обращения: 15.11.2020, режим доступа: свободный)
- [3] sem\_overview(7) Linux User's Manual [Электронный ресурс]. //URL: [https://man7.org/linux/man-pages/man7/sem\\_overview.7.html](https://man7.org/linux/man-pages/man7/sem_overview.7.html) (Дата обращения: 13.12.2020, режим доступа: свободный)
- [4] pthread\_mutex\_lock(3p) Linux User's Manual [Электронный ресурс]. //URL: [https://linux.die.net/man/3/pthread\\_mutex\\_lock](https://linux.die.net/man/3/pthread_mutex_lock) (Дата обращения: 13.12.2020, режим доступа: свободный)
- [5] Статья «ANSI escape code» Wikipedia.org //URL: [https://en.wikipedia.org/wiki/ANSI\\_escape\\_code](https://en.wikipedia.org/wiki/ANSI_escape_code) (Дата обращения: 13.12.2020, режим доступа: свободный)