

**Отчёт о выполнении индивидуального домашнего задания №2**

**Вариант №5**

**(Задание на 5 баллов)**

Отчёт подготовлен студентом  
ФКН БПИ228  
Лысиным Кириллом

## Условие задания

Задача о читателях и писателях. Базу данных, представленную массивом целых положительных чисел, разделяют два типа процессов: N читателей и K писателей. Читатели периодически просматривают случайные записи базы данных и выводя номер свой номер (например, PID), индекс записи, ее значение, а также вычисленное значение числа Фибоначчи. Писатели изменяют случайные записи на случайное число и также выводят информацию о своем номере, индексе записи, старом значении и новом значении. Предполагается, что в начале БД находится в непротиворечивом состоянии (все числа отсортированы, например, по возрастанию). Каждая отдельная новая запись переводит БД из одного непротиворечивого состояния в другое (то есть, новая сортировка может поменять индексы записей или переставить числа). 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 многопроцессное приложение с потоками-писателями и потоками-читателями. 7 Каждый читатель и писатель моделируется отдельным процессом.

## Общая схема решения

Программа моделирует задачу о читателях и писателях, используя множество процессов для представления читателей и писателей. Читатели периодически выбирают случайные записи из базы данных и считывают их значения, вычисляя при этом числа Фибоначчи. Писатели изменяют случайные записи базы данных, заменяя старые значения новыми. При этом, чтобы избежать конфликтов, процесс-писатель имеет исключительный доступ к базе данных.

Каждый процесс работает независимо друг от друга, что обеспечивает параллельное выполнение транзакций чтения и записи. Для предотвращения состояний гонки и обеспечения синхронизации доступа используются семафоры. Эти механизмы гарантируют безопасность операций чтения и записи, предотвращая взаимное влияние транзакций и обеспечивая непротиворечивость базы данных.

Таким образом, программа обеспечивает эффективное управление доступом к базе данных и обеспечивает безопасность параллельного доступа для чтения и записи. Все процессы взаимодействуют, соблюдая правила доступа к данным и обеспечивая корректное функционирование базы данных.

Формат ввода: `./main <readers_count> <writers_count>`

Набор тестов (включает не менее 5 тестовых файлов) хранится в родительской директории в папке Tests.

Результаты работы программы находятся в папке TestResults.