Штанов Андрей Николаевич

БПИ208

Вариант 35

**Задача о читателях и писателях - 2 («грязное чтение»).** 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 (например, если каждый элемент — число, то они все отсортированы). 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.

Разработанная программа демонстрирует применение нескольких потоков, записывающих и считывающих информацию в файле и конкурирующих между собой за право доступа к ресурсу.

Количество требуемых операций на запись настраивается в программе, а путь к текстовому файлу (с данными, разделёнными символами пробела или переносами строки) спрашивается у пользователя в начале выполнения программы.

Осуществляется запуск нескольких потоков на запись и четырёх – на чтение. Потоки на запись записывают в текстовый файл случайные числа, пока не достигнуто ограничение на количество записей. Потоки на чтение считывают числа, записанные в файл, пока количество считанных чисел, меньше количества записанных в файл. Также поток на чтение проверяет, было ли записано достаточно информации в файл, чтобы считать следующий символ.

Для того, чтобы не произошло нарушение целостности переменных, а также содержимого в файле необходимо использовать один из методов синхронизации потоков. В данном случае используется механизм mutex-ов, как частный случай семафоров. При выполнении критической операции (доступа к разделяемым несколькими потоками данным) выполняющийся в данный момент времени поток блокирует возможность доступа к данным для другим потоками, выполняет требуемые действия и снимает блокировку.

Таким образом, появляется возможность выполнения множества потоков, обращающихся к общим данным – они выполняют какие-то операции, а в нужный момент ставят блокировку и записывают / считывают информацию из общих данных.

Источники информации:

1. https://en.cppreference.com/w/cpp/thread/mutex
2. https://medium.com/nuances-of-programming/c-мьютекс-пишем-наш-первый-код-для-многопоточной-среды-543a3d60ef30
3. https://ru.wikipedia.org/wiki/Мьютекс