БГУИР

Кафедра ЭВМ

Отчет по лабораторной работе № 7

Тема: «Блокировки чтения/записи и условные переменные»

Выполнил:

студент группы 150502 Альхимович Н.Г.

Проверил:

старший преподаватель Поденок Л.П.

Минск

2023

**1 ЦЕЛЬ РАБОТЫ**

Реализовать задачу «производители-потребители» с использованием потоков в рамках одного процесса и условных переменных. Изучить способ конкурентного доступа к совместно используемому файлу посредством блокировки чтения/записи.

**2 ПОСТАНОВКА ЗАДАЧИ**

Программа в режиме конкурентного доступа читает из и пишет в файл, содержащий записи фиксированного формата. Формат записей произвольный. Примерный формат записи:

struct record\_s {

char name[80]; // Ф.И.О. студента

char address[80]; // адрес проживания

uint8\_t semester; // семестр

}

Файл должен содержать не менее 10 записей. Создается и наполняется с помощью любых средств.

Программа должна выполнять следующие операции:

1. LST ‒ Отображение содержимого файла с последовательной нумерацией записей;
2. GET(Rec\_No) ‒ получение записи с порядковым номером Rec\_No;
3. Модификацию полей записи;
4. PUT() ‒ сохранение последней прочитанной и модифицированной записи по месту. Интерфейс с пользователем на «вкус» студента.

Алгоритм конкурентного доступ к записи:

REC <-- get(Rec\_No)

Again:

REC\_SAV <-- REC

if (REC модифицирована) {

lock(Rec\_No)

REC\_NEW <-- get(Rec\_No)

if (REC\_NEW != REC\_SAV) {

unlock(Rec\_No)

REC <-- REC\_NEW

goto Again

}

put(REC, Rec\_No) unlock(Rec\_No)

}

Для отладки и тестирования используется не менее двух экземпляров программы.

**3 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ**

1. Результат работы программы main\_thread.c:

Результат работы программы с опцией c:

nina@nina-MacBook-AIr:~/Документы/Альхимович Н.Г./Lab07:main -$ make

gcc -W -Wall -Wextra -Wno-unused-parameter -Wno-unused-variable -std=c11 -pedantic main\_thread.c -o main\_thread

gcc -W -Wall -Wextra -Wno-unused-parameter -Wno-unused-variable -std=c11 -pedantic concurrency.c -o concurrency

nina@nina-MacBook-AIr:~/Документы/Альхимович Н.Г./Lab07:main -$ ./main\_thread

Выберите действие:

c - создать поток-производитель;

+ - создать поток-потребитель;

d - закрыть поток-производитель;

- - закрыть поток-потребитель;

> - увеличить размер очереди;

< - уменьшить размер очереди;

q - завершить работу.

c

Поток-производитель 281473798566176 создан

Итого: 1 производителей

1-ое сообщение добавлено в очередь

Контрольные данные: 8545

Длина данных: 58

Данные сообщения: 15 184 193 62 97 240 45 219 247 140 168 51 93 200 252 8 38 159 57 231 53 213 15 109 181 120 225 243 91 239 34 106 167 227 169 9 211 214 228 203 98 140 254 191 85 250 199 123 153 1 99 207 214 114 60 140 234 30 127 70

Результат работы программы с опцией +:

+

Поток-потребитель 281473798566176 создан

Итого: 1 потребителей

1 сообщений извлечено из очереди

Результат работы программы с опцией d:

d

Поток 281473798566176 закрыт

Осталось 0 потоков

Результат работы программы с опцией -:

-

Поток 281473798566176 закрыт

Осталось 0 потоков

Результат работы программы с опцией >:

>

Размер очереди увеличен с 100 до 101

Результат работы программы с опцией q:

q

Потоки не были созданы

Поток 281473798566176 закрыт

Все потоки закрыты

1. Результат работы программы concurrency.c:

Результат работы программы с опцией 1:

nina@nina-MacBook-AIr:~/Документы/Альхимович Н.Г./Lab07:main -$ make

gcc -W -Wall -Wextra -Wno-unused-parameter -Wno-unused-variable -std=c11 -pedantic main\_thread.c -o main\_thread

gcc -W -Wall -Wextra -Wno-unused-parameter -Wno-unused-variable -std=c11 -pedantic concurrency.c -o concurrency

nina@nina-MacBook-AIr:~/Документы/Альхимович Н.Г./Lab07:main -$ ./concurrency

Выберите действие:

1 - LST (вывести содержимое файла);

2 - GET (получить запись по номеру);

3 - MOD (модифицировать поля записи);

4 - PUT (сохранить последнюю запись);

0 - завершить работу.

1

Запись №0:

ФИО: Lee

Адрес: Seoul

Семестр: 7

Запись №1:

ФИО: Han

Адрес: Seoul

Семестр: 4

Запись №2:

ФИО: Popov

Адрес: Omsk

Семестр: 8

Запись №3:

ФИО: Holland

Адрес: London

Семестр: 6

Запись №4:

ФИО: Holland

Адрес: London

Семестр: 6

Запись №5:

ФИО: Koval

Адрес: Kovrov

Семестр: 7

Запись №6:

ФИО: Anisimov

Адрес: Vladivostok

Семестр: 7

Запись №7:

ФИО: Yang

Адрес: Busan

Семестр: 1

Запись №8:

ФИО: Morgan

Адрес: Liverpool

Семестр: 5

Запись №9:

ФИО: Gavi

Адрес: Barcelona

Семестр: 3

Вывод завершен

Результат работы программы с опцией 2:

2

Введите номер записи: 8

ФИО: Morgan

Адрес: Liverpool

Семестр: 5

Результат работы программы с опцией 3:

3

Введите номер записи, которую нужно изменить: 4

ФИО: Holland

Адрес: London

Семестр: 6

Заполните поля, требующие модификации:

ФИО: Orlov

Адрес: Minsk

Семестр: 2

Для сохранения изменений выберите четвертый пункт меню

4

Результат работы программы с опцией 4:

Запись №4 модифицирована

Выберите действие:

1 - LST (вывести содержимое файла);

2 - GET (получить запись по номеру);

3 - MOD (модифицировать поля записи);

4 - PUT (сохранить последнюю запись);

0 - завершить работу.

1

Запись №0:

ФИО: Lee

Адрес: Seoul

Семестр: 7

Запись №1:

ФИО: Han

Адрес: Seoul

Семестр: 4

Запись №2:

ФИО: Popov

Адрес: Omsk

Семестр: 8

Запись №3:

ФИО: Holland

Адрес: London

Семестр: 6

Запись №4:

ФИО: Orlov

Адрес: Minsk

Семестр: 2

…

Вывод завершен

**4 ВЫВОД**

В ходе лабораторной работы были разработаны 2 программы: организующая создание потоков двух видов в зависимости от выбранной опции (производитель (producer), формирующий сообщение и добавляющий его в очередь, и потребитель (consumer), извлекающий сообщение из очереди), управляющая их работой и создающая очередь сообщений заданной структуры с использованием условных переменных; и осуществляющая работу с файлом в режиме конкурентного доступа, позволяющая просматривать его содержимое, получать запись по запросу и модифицировать ее, если требуется.