

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин

Лабораторная работа №7
по курсу
«Операционные системы и системное программирование»
на тему
«Блокировки чтения/записи»

Выполнил:

студент группы 350501

Проверил:

Русак Г.Д.
старший преподаватель каф. ЭВМ
Поденок Л.П.

Минск 2025

1 ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Конкурентный доступ к совместно используемому файлу, используя блокировку чтения-записи. Изучаемые системные вызовы: `fcntl(F_GETLK, F_SETLK, F_SETLKW, F_UNLK)`.

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) отображение содержимого файла с последовательной нумерацией записей (LST);

2) получение записи с порядковым номером `Rec_No` (GET `Rec_No`);

3) модификацию полей записи;

4) сохранение последней прочитанной и модифицированной записи по месту (PUT).

Интерфейс с пользователем на «вкус» студента.

Алгоритм конкурентного доступа к записи:

```
REC <-- get(Rec_No) // читаем запись в буфер REC
Again:
REC_WRK <-- REC // копируем в рабочую область
... // делаем что-нибудь с записью в REC_WRK и желаем ее
сохранить
if (REC_WRK != REC) { //
    lock(Rec_No) // блокируем запись для модификации в файле
    REC_NEW <-- get(Rec_No) // и перечитываем запись из файла
    if (REC_NEW != REC) { // кто-то изменил запись после ее
получения
        unlock(Rec_No) // освобождаем запись
        message // сообщаем, что запись кто-то изменил
        REC <-- REC_NEW // повторим все с ее новым содержимым
        goto Again
    }
    put(REC_WRK, Rec_No) // сохраняем новое содержимое
    unlock(Rec_No) // освобождаем запись
}
```

Для отладки и тестирования используется не менее двух экземпляров программы.

Для блокирования записей используется `fcntl()` и исключительные блокировки на основе файловых описаний (`F_OFD_*`).

2 ОПИСАНИЕ АЛГОРИТМОВ И РЕШЕНИЙ

2.1 Общая структура

Программа представляет собой интерактивную систему для работы с записями студентов, хранящимися в файле. Основные функции:

Чтение и вывод записей (просмотр всех или конкретной записи).

Изменение данных с проверкой на конфликты.

Блокировка записей для обеспечения согласованности при многопользовательской работе.

Автоматический повтор операции при обнаружении изменений другим процессом. Программа спрашивает у пользователя, необходимо ли заменить данные, которые были изменены другим процессом.

Программа использует низкоуровневые файловые операции (POSIX API) для работы с данными, включая блокировки на уровне записей (fcntl).

2.2 Алгоритм работы основного процесса

Алгоритм работы основного процесса:

- 1) инициализация;
- 2) открытие файла данных (создание, если не существует);
- 3) проверка доступности файла и прав доступа;
- 4) главный цикл обработки команд;
- 5) вывод меню с доступными операциями;
- 6) ожидание ввода команд:
 - LST - вывод всех записей;
 - GET - чтение конкретной записи;
 - PUT - сохранение изменений с проверкой на конфликты;
 - EXIT - завершение работы.
- 7) обработка команд;
- 8) завершение работы;
- 9) корректное закрытие файла.

3 ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОЕКТА

Система состоит из нескольких ключевых модулей, каждый из которых выполняет определенную функцию.

Модуль управления файлом записей (`file_operations`) выполняет следующие функции:

1) Инициализация файла с тестовыми записями при первом запуске. Проверка существования и доступности файла;

2) Реализация механизма блокировок (`F_RDLCK`, `F_WRLCK`) для безопасного многопользовательского доступа. Автоматическое снятие блокировок при завершении операций;

3) Вывод одной записи с проверкой блокировок `print_record()`;

4) Чтение записи с блокировкой для чтения `get_record()`;

5) Изменение записи `modify_record()`;

6) Сохранение изменений с проверкой конфликтов `save_record()`.

Модуль пользовательского интерфейса (`user_interface`) выполняет следующие функции:

1) Обработка пользовательского ввода `menu()`;

2) Очистка буфера ввода `clear_input()`;

3) Вывод всех записей;

4) Получение конкретной записи;

5) Изменение записи.

Модуль управления программой (`main`) выполняет следующие функции:

1) Инициализация системы. Обработка аргументов командной строки. Открытие/создание файла данных. Установка обработчика сигналов;

2) Обработка сигналов `handle_signal()`;

3) Запуск пользовательского интерфейса.

Модуль структуры данных (`data_structures`) включает следующее:

1) Дескриптор файла `fd`;

2) Структура для блокировок `fl`;

3) Константы (`MAX_RECORDS`, `MAX_NAME_LEN`);

4) `record_t` - структура записи (имя, адрес, семестр).

4 ПОРЯДОК СБОРКИ И ЗАПУСКА ПРОЕКТА

Порядок сборки и запуска состоит в следующем:

1) Клонировать репозиторий, используя команду

```
$git clone https://github.com/Everolfe/lab07-osasp,
```

или разархивировать каталог с проектом;

2) Перейти в каталог с проектом

```
$cd lab07-osasp,
```

или

```
$cd "Русак Г.Д./lab07";
```

3) Собрать проект используя make;

4) После сборки проекта можно использовать, прописав

```
./build/release/main test
```

5 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

```
grusak@fedora:~/tar_working_dir/Рысак          Г.Д./lab07$
./build/release/main test
=====
Select an action:
l - LST
g - GET
p - PUT
q - QUIT
=====
Command: g
Enter the record number: 2
=====
Record      2:
Name:      Ada;
Address:   ZDAD;
Semester:  6.
=====

Command: p
Enter the new name for the record: Ivan
Enter the new address for the record: Minsk
Enter the new semester for the record: 5
Trying to acquire write lock for record 2...
Write lock acquired for record 2.
Record 2 successfully saved.
Write lock released for record 2.

Command: g
Enter the record number: 2
Record 2 is locked for writing. Cannot display now.

Trying to acquire write lock for record 2...
Write lock acquired for record 2.
WARNING: Record 2 was modified by another process.
Current contents:
Name: Ea
Address: As
Semester: 4
Your modified record was NOT saved.
Do you want to retry saving with updated data? (y/n): y
Trying to acquire write lock for record 2...
Write lock acquired for record 2.
Record 2 successfully saved.
Write lock released for record 2.
```