

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
М.В. Антимонов

Проверил:

старший преподаватель каф. ЭВМ
Л.П. Поденок

Минск 2025

1 УСЛОВИЕ ЛАБОРАТОРНОЙ РАБОТЫ

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 доступ 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 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;
MOD – редактирование последней прочитанной записи;
PUT – сохранение изменений с проверкой на конфликты;
EXIT – завершение работы.
7. Обработка команд;
8. Завершение работы;
9. Корректное закрытие файла.

2.2 Работа с записями

Алгоритм работы с записями:

1. Чтение записи;
 2. Определение смещения в файле на основе номера записи;
 3. Установка позиции в файле и чтение данных;
 4. Проверка корректности операции (существует ли запись);
 5. Сохранение записи;
 6. Блокировка записи для исключения одновременного изменения;
 7. Проверка, не была ли запись изменена другим процессом;
 8. Если изменений не было – сохранение данных;
 9. Если обнаружен конфликт – повторная попытка с новыми данными.
- Особенности блокировки и разблокировки включают в себя

использование эксклюзивных блокировок (F_WRLCK) для предотвращения гонки данных. Блокировка устанавливается перед изменением и снимается после завершения операции.

2.3 Механизм обработки конфликтов

Механизм обработки конфликтов при сохранении изменений:

1. Захватывается блокировка записи;
2. Текущее состояние записи сравнивается с исходным (до редактирования);
3. Если данные не изменились - изменения сохраняются;
4. Если запись была изменена другим процессом:
 - Пользователь видит разницу между текущими и своими изменениями;
 - Происходит автоматическая повторная попытка сохранения с новыми данными.

Этот подход гарантирует целостность данных даже при параллельной работе нескольких процессов.

2.4 Особенности реализации

Особенности оптимистичной блокировки:

- Изменения применяются только если запись не менялась с момента чтения;
- При конфликте – повторная попытка вместо принудительной перезаписи.

Особенности интерактивного редактирования:

- Пользователь может изменять отдельные поля, оставляя остальные без изменений;
- Валидация ввода (например, семестр должен быть от 1 до 12).

Особенности эффективного управления файлом:

- Прямой доступ к записям через смещения (lseek);
- Минимальное число операций ввода-вывода;
- Безопасность;
- Контроль границ файла при чтении.

3 ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОЕКТА

Реализация системы построена вокруг функции `main()`, которая выполняет инициализацию файловых дескрипторов, обработку пользовательских команд и управление блокировками. В начале программы осуществляется открытие файла данных с проверкой его доступности, при этом если файл не существует, он автоматически создается с правами доступа 0644.

После успешного открытия файла программа переходит в основной цикл обработки команд, где последовательно выполняет:

1. Вывод интерактивного меню с доступными операциями;
2. Чтение введенной пользователем команды;
3. Вызов соответствующей функциональности.

Система поддерживает пять основных команд:

LST – вывод полного списка записей;

GET – чтение конкретной записи по номеру;

MOD – модификация последней прочитанной записи;

PUT – сохранение изменений с проверкой конфликтов;

EXIT – корректное завершение работы.

Для работы с записями реализованы следующие ключевые функции:

`list_records()` – вычисляет общее количество записей в файле и последовательно выводит их содержимое;

`get_record()` – осуществляет позиционирование в файле и чтение конкретной записи по заданному номеру;

`put_record()` – выполняет запись данных в указанную позицию файла;

`lock_record()` и `unlock_record()` – управляют эксклюзивными блокировками записей через механизм `fcntl`.

Особое внимание уделено механизму модификации записей, который реализует оптимистичную блокировку:

1. Пользователь инициирует изменение через команду MOD;
2. Программа создает рабочую копию записи;
3. При сохранении (PUT) система проверяет, не была ли запись изменена другим процессом;
4. В случае обнаружения конфликта пользователю показываются

различия и предлагается повторить операцию.

Синхронизация доступа организована через:

- Эксклюзивные блокировки записей (F_WRLCK) на время изменений;
- Автоматические повторные попытки при обнаружении конфликтов;
- Атомарные операции чтения-записи.

Работа с файлом построена на низкоуровневых операциях:

- Позиционирование через lseek;
- Чтение/запись через read/write;
- Управление блокировками через fcntl.

Валидация данных включает:

- Проверку допустимых номеров записей;
- Контроль границ файла при операциях чтения;
- Валидацию вводимых значений (например, семестр 1-12).

Завершение работы программы включает:

- Корректное закрытие файлового дескриптора;
- Освобождение всех системных ресурсов.

Особенности реализации:

1. Потокбезопасность – блокировки записей дают работать нескольким процессам одновременно;
2. Отказоустойчивость – автоматическое восстановление при конфликтах изменений;
3. Эффективность – минимальное количество операций ввода-вывода за счет прямого доступа;
4. Удобство интерфейса – подсказки и сохранение предыдущих значений при редактировании.

Структура данных record_s содержит три поля:

- name – строка имени (80 символов);
- address – строка адреса (80 символов);
- semester – номер семестра (1 байт).

Программа эффективно работает с файлами любого размера, поддерживая как единичные записи, так и их большие массивы, благодаря использованию прямого позиционирования и фиксированного размера структур.

4 ПОРЯДОК СБОРКИ И ЗАПУСКА

1) Перейти в каталог проекта:

```
$ cd 'Антимонов М.В./lab07'
```

2) Собрать проект с помощью make:

```
$ make
```

3) Запустить программу:

```
$ ./concurrentFileWork
```


5 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Доступные команды:

LST - отобразить все записи

GET <номер> - получить запись по номеру

MOD - модифицировать последнюю прочитанную запись

PUT - сохранить последнюю модифицированную запись

EXIT - выйти из программы

Введите команду: LST

Список всех записей (10):

Запись #1:

Имя: Иванов Иван Иванович

Адрес: г. Москва, ул. Ленина, д. 1, кв. 10

Семестр: 1

Запись #2:

Имя: Петров Петр Петрович

Адрес: г. Санкт-Петербург, пр. Невский, д. 5, кв. 42

Семестр: 2

Запись #3:

Имя: Сидорова Анна Сергеевна

Адрес: г. Екатеринбург, ул. Мира, д. 17, кв. 89

Семестр: 3

Запись #4:

Имя: Козлова Мария Александровна

Адрес: г. Новосибирск, ул. Кирова, д. 25, кв. 15

Семестр: 4

Запись #5:

Имя: Смирнов Алексей Дмитриевич

Адрес: г. Казань, ул. Пушкина, д. 10, кв. 76

Семестр: 5

Запись #6:

Имя: Федоров Николай Андреевич

Адрес: г. Нижний Новгород, ул. Советская, д. 8, кв. 33

Семестр: 6

Запись #7:

Имя: Морозова Екатерина Ивановна

Адрес: г. Самара, ул. Гагарина, д. 12, кв. 51

Семестр: 1

Запись #8:

Имя: Волков Сергей Владимирович

Адрес: г. Омск, ул. Ленина, д. 7, кв. 22

Семестр: 2

Запись #9:

Имя: Лебедева Ольга Викторовна

Адрес: г. Ростов-на-Дону, ул. Космонавтов, д. 15, кв. 68

Семестр: 3

Запись #10:

Имя: UI IU

Адрес: КНУ J

Семестр: 5