

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра ЭВМ

Дисциплина: Операционные системы и системное программирование

ОТЧЁТ  
к лабораторной работе №7  
на тему  
Блокировки чтения/записи.

Выполнил студент гр.230501 Лазовский И.А.

Проверил старший преподаватель кафедры ЭВМ  
Поденок Л.П.

Минск 2024

## 1 УСЛОВИЕ ЛАБОРАТОРНОЙ РАБОТЫ.

### Задание

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) 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) // читаем запись
Again:
REC_SAV <-- REC // сохраним копию
... // делаем что-нибудь с записью и желаем ее сохранить
if (REC модифицирована) {
    lock(Rec_No) // блокируем запись для модификации в файле
    REC_NEW <-- get(Rec_No) // и перечитываем
    if (REC_NEW != REC_SAV) { // кто-то изменил запись после получения ее нами
        unlock(Rec_No) // освобождаем запись
        message // сообщаем, что запись кто-то изменил
        REC <-- REC_NEW // повторим все с ее новым содержимым если нужно
        goto Again
    }
    put(REC, Rec_No) // сохраняем новое содержимое
    unlock(Rec_No) // освобождаем запись
}
```

Для отладки и тестирования используется не менее двух экземпляров программы.

## **2 ОПИСАНИЕ АЛГОРИТМОВ И РЕШЕНИЙ.**

`write_records()`: Записывает все записи из массива `records` в файл.  
`write_record()`: Записывает одну запись в файл.  
`print_record()`: Выводит на экран данные одной записи.  
`print_records()`: Читает все записи из файла и выводит их на экран.  
`readCurrentRecord()`: Читает одну запись из файла.  
`get()`: Возвращает запись из файла по ее номеру.  
`clear_file()`: Очищает содержимое файла.  
`delay()`: Ожидает нажатия любой клавиши для продолжения.  
`menu()`: Выводит на экран меню с доступными действиями.  
`equals()`: Сравнивает две записи и возвращает `true`, если они идентичны.  
`modificate()`: Позволяет пользователю изменять данные одной записи.  
`save()`: Сохраняет изменения в записи с помощью блокировки файла.  
`menu2()`: Выводит меню для изменения данных записи.

Главная функция `main()`: Основной цикл программы, в котором пользователь может выбирать действия из меню: просмотр всех записей, получение записи по номеру, выход из программы. Перед выходом из программы вызывается функция `write_records()` для записи данных в файл.

## **3. ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОЕКТА.**

Проект собирается с помощью `makefile`. Пример запуска:

```
ilua@fedora:~/Рабочий стол/labFor$ build/debug/main
```

Для запуска проекта нам требуется в терминале запустить программу `main`. Где программа сразу предоставляет меню для работы.

В проекте имеется каталог для сборки `debug` и `release`. Каталог `git` для системы контроля версий моего проекта. Директория `src` с исходным кодом. И `makefile` для компиляции и сборки проекта.

## **4. ПОРЯДОК СБОРКИ И ИСПОЛЬЗОВАНИЯ.**

Для компиляции и сборки проекта используется `makefile`.

**Порядок сборки:**

4CC: Это неявная переменная, которая указывает на компилятор. В данном случае используется gcc.

CFLAGS\_DEBUG: Флаги компиляции для режима отладки.

CFLAGS\_RELEASE: Флаги компиляции для релизного режима.

DEBUG: Путь к директории с отладочными файлами.

RELEASE: Путь к директории с релизными файлами.

OUT\_DIR: Исходно установлен на \$(DEBUG), но может изменяться на \$(RELEASE) в зависимости от режима сборки.

all: Цель, которая собирает исполняемый файл main.

\$(prog): Это правило для сборки исполняемого файла main. Оно зависит от объектных файлов \$(objects).

\$(OUT\_DIR)/%.o: Это правило для сборки объектных файлов из исходных файлов .c.

ifeq (\$(MODE), release): Если переменная MODE установлена в release, то используются флаги для релизного режима.

.PHONY: clean: Это объявление говорит make, что clean - это фиктивная цель (не связанная с файлами).

clean: Удаляет все файлы в директориях \$(DEBUG) и \$(RELEASE).

### **Порядок использования.**

#### **1. Компиляция:**

Для отладочной сборки: make или make MODE=debug

Для релизной сборки: make MODE=release

#### **2. Очистка:**

make clean - удаляет все объектные файлы и исполняемый файл.

#### **3. Запуск:**

После успешной компиляции запустите исполняемый файл, например: ./build/debug/main.

### **5. МЕТОД ТЕСТИРОВАНИЯ И РЕЗУЛЬТАТ ТЕСТИРОВАНИЯ.**

```
root@fedora:~/Рабочий стол/lab7# build/debug/main
```

```
1 - VIEW all students
```

```
2 - GET student bu number
```

```
e - EXIT
```

```
1
```

```
List of students:
```

```
Sarah Johnson 11 Maple Ave 3
```

```
Ethan Rodriguez 098 Elm Rd 1
```

```
Ava Nguyen 83 Cedar Ln 2
```

Lucas Hernandez 15 Oak Blvd 4  
Isabella Morales 41 Pine St 2  
Noah Ramirez 88 Birch Ct 1  
Mia Diaz 87 Walnut Dr 3  
Jacob Vargas 009 Spruce Rd 4  
Abigail Flores 647 Maple Ter 2  
William Castillo 123 Elm Ct 1  
Нажмите любую клавишу для продолжения: f  
1 - VIEW all students  
2 - GET student bu number  
e - EXIT  
2  
Enter number student: 2  
Ethan Rodriguez 098 Elm Rd 1  
'1' - change data  
'2' - Exit  
1  
Ethan Rodriguez 098 Elm Rd 1  
'1' - change full name  
'2' - change address  
'3' - change semester  
'4' - cancel  
4  
Нажмите любую клавишу для продолжения: f  
1 - VIEW all students  
2 - GET student bu number  
e - EXIT  
e

