Министерство образования Республики Беларусь
Учреждение Образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных машин

Лабораторная работа № 7
«Блокировка чтения/записи»

Проверил: Поденок Л.П.
Выполнил: ст. гр. 350501
Маслаков Н.А.

Минск 2025

1. УСЛОВИЕ ЛАБАРАТОРНОЙ 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];
char address[80];
uint8_t semester;
};
```

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.

2. ОПИСАНИЕ АЛГОРИТМОВ И РЕШЕНИЙ

Программа представляет собой консольное приложение на языке C с интерактивной оболочкой, предназначенное для конкурентной работы с файлом `data.bin`, содержащим фиксированные записи студентов. Каждая запись имеет размер 161 байт и состоит из трёх полей: ФИО (80 байт), адрес (80 байт) и семестр (1 байт). Смещение нужной записи вычисляется как `off_t offset = rec_no * sizeof(struct record_s)` и устанавливается вызовом `lseek()`. Чтение и запись выполняются через `read()` и `write()` соответственно.

Для синхронизации нескольких процессов применяется механизм OFD-блокировок на основе дескриптора файла `(F_OFD_*)`. Реализованы две функции: `lock_record()` и `unlock_record()`, инкапсулирующие работу с `fcntl()`. При чтении записи перед `read()` ставится разделяемая блокировка `F_RDLCK` (несколько процессов могут одновременно читать одну и ту же запись), а при модификации — эксклюзивная блокировка `F_WRLCK`, чтобы ни чтение, ни запись другими процессами не мешали текущей операции. Команда `F_OFD_SETLKW` в блокируется до освобождения записи, а для снятия блокировки используется неблокирующая `F_OFD_SETLK` с типом `F_UNLCK`.

3. ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОЕКТА

`void cmd_lst().`

Функция `cmd_lst` выводит в консоль все записи файла `data.bin`.

`void cmd_get(int rec_no).`

Функция `cmd_get` используется для задания чтения и отображения в консоль одной записи из файла.

Принимает аргумент `rec_no` - номер записи.

`void cmd_put(int rec_no).`

Функция `cmp_index_s` используется для изменения записи под переданным номером как.

Принимает аргумент `rec_no` - номер записи.

`int lock_record(int fd, int rec_no, short type).`

Функция `lock_record` используется для установки блокировки на определенную запись в файле.

Принимает аргументы:

1) `int fd`. Файловый дескриптор.
2) `int rec_no`. Номер записи.
3) `short type`. Тип блокировки(чтение или запись).

`int unlock_record(int fd, int rec_no).`

Функция `unlock_record` используется для снятия блокировки на определенную запись в файле.

Принимает аргументы:

1) `int fd`. Файловый дескриптор.
2) `int rec_no`. Номер записи.

`ssize_t read_record(int fd, int rec_no, struct record_s *rec).`

Функция `l read_record` используется для чтения одной записи из файла по ее номеру и сохранения в структуру `record_s`.

Принимает аргументы:

1) `int fd`. Файловый дескриптор.
2) `int rec_no`. Номер записи.
3) `record_s *rec`. Структура, куда будет помещен результат.

`ssize_t write_record(int fd, int rec_no, const struct record_s *rec).`

Функция `write_record` используется для записи структуры в файл.

Принимает аргументы:

1) `int fd`. Файловый дескриптор.
2) `int rec_no`. Номер записи.
3) `record_s *rec`. Структура, что будет записана.

4. ПОРЯДОК СБОРКИ И ЗАПУСКА

1) Перейти в каталог проекта.

```
$ cd 'Маслаков Н.А./lab07'
```

2) Собрать проект с помощью `make`. по умолчанию сборка происходит в режиме отладки.

```
$ make
```

3) Запустить программу.

```
$ build/debug/app
```

5. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

1) Обычное использование

```
~/lab07$ ./build/debug/app
Интерактивный режим. Доступные команды: LST, GET, PUT, INIT, QUIT
> LST
0: , , sem=0
1: , , sem=0
2: , , sem=0
3: ff, ffff, sem=11
4: , , sem=0
5: , , sem=0
6: , , sem=0
7: , , sem=0
8: , , sem=0
9: , , sem=0
> INIT
Инициализирован файл 'data.bin' с 10 записями
> LST
0: Student00, Address00, sem=1
1: Student01, Address01, sem=2
2: Student02, Address02, sem=3
3: Student03, Address03, sem=4
4: Student04, Address04, sem=5
5: Student05, Address05, sem=6
6: Student06, Address06, sem=7
7: Student07, Address07, sem=8
8: Student08, Address08, sem=1
9: Student09, Address09, sem=2
> GET 5
Неизвестная команда: ПааGET
> GET 5
REC[5]: Student05, Address05, sem=6
> PUT 5
Текущие данные:
Name: Student05
Address: Address05
Semester: 6
Введите новое ФИО (или Enter для пропуска): LFE f
Введите новый адрес (или Enter): fwef  ff
Введите семестр (0-255, Enter для пропуска): 11
Запись 5 обновлена.
> GET 5
REC[5]: LFE f, fwef  ff, sem=11
```

```
> LST
0: Student00, Address00, sem=1
1: Student01, Address01, sem=2
2: Student02, Address02, sem=3
3: Student03, Address03, sem=4
4: Student04, Address04, sem=5
5: LFE f, fwef  ff, sem=11
6: Student06, Address06, sem=7
7: Student07, Address07, sem=8
8: Student08, Address08, sem=1
9: Student09, Address09, sem=2
> QUIT
```

2) Блокировка изменения

Терминал 1

```
~/lab07$./build/debug/app PUT 3
Текущие данные:
Name: Student03
Address: Address03
Semester: 4
Введите новое ФИО (или Enter для пропуска): qq
Введите новый адрес (или Enter): qqqq
Введите семестр (0–255, Enter для пропуска): 123
Запись 3 обновлена.
```

Терминал 2

```
~/lab07$./build/debug/app PUT 3
Текущие данные:
Name: Student03
Address: Address03
Semester: 4
Введите новое ФИО (или Enter для пропуска): dd
Введите новый адрес (или Enter): GEGGG GE
Введите семестр (0–255, Enter для пропуска): 2
!!! Конфликт при сохранении: запись 3 уже изменена. Новые данные:
  qq, qqqq, sem=123
Текущие данные:
Name: qq
Address: qqqq
Semester: 123
Введите новое ФИО (или Enter для пропуска): La Lala
Введите новый адрес (или Enter): FF 12
Введите семестр (0–255, Enter для пропуска): 1
Запись 3 обновлена.

~/lab07$./build/debug/app GET 3
REC[3]: La Lala, FF 12, sem=1
```