

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Отчет

Лабораторная работа №7

Выполнил студент гр. 150502:

Бейнар А.В.

Проверил:

Поденок Л.П.

Минск 2023

Тема работы:

Блокировки чтения/записи и условные переменные

Цель работы:

Задача «производители-потребители»; конкурентный доступ к совместно используемому файлу, используя блокировку чтения-записи.

Задание:

Здесь две программы.

1) Задача «производители-потребители». Аналогична лабораторной № 4, но для потоков с использованием условных переменных (см. лекции СПОВМ).

Изучаемые системные вызовы (префикс pthread_ опущен): cond_init(), cond_destroy(), cond_*wait(), cond_signal().

2) 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 доступа к записи

```
REC <-- get(Rec_No)  
Again:  
REC_SAV <-- REC // сохраним копию  
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 и  
        REC <-- REC_NEW // повторим все с ее новым содержимым
```

```

goto Again
}
put(REC, Rec_No) // сохраняем новое содержимое
unlock(Rec_No) // освобождаем запись
}

```

Для отладки и тестирования используется не менее двух экземпляров программы.

Вывод в консоли после компиляции:

1) Задача «производители-потребители»

```

MENU:
1 - print options
2 - create producer
3 - delete producer
4 - create consumer
5 - delete consumer
6 - info
+ - increase queue
- - decrease queue
7 - quit
2
1-p) 140612872365632 produce msg: hash=20A6A874
2-p) 140612872365632 produce msg: hash=B6498865
3
3
Amount producers = 0
6
Count producer: 0; count consumer: 0; count messages in queue: 2
4
1-c) 140612872365632 consume msg: hash=20A6A874
2-c) 140612872365632 consume msg: hash=B6498865
7

```

2) Конкурентный доступ к файлу

```

Commands:
ADD - add a new record
LST - list all records
GET <record number> - get record with number n
MOD - modify an existing record
PUT - save modified record
QUIT - exit the program

```

> LST

Records:

0. alecs, add, 1
1. Pasha, dfj, 1
2. Galina, fff, 3
3. , , 0
4. Nadya, sad, 3
5. kjf, kjf, 1

> MOD

Nothing to edit. Call GET function!

> GET 3

Record #3

Name:

Address:

Semester: 0

> MOD

Enter name: alecsandr

Enter address: okt

Enter semester: 4

Record modified at position 3

> PUT

Record #3 saved

> LST

Records:

0. alecs, add, 1
1. Pasha, dfj, 1
2. Galina, fff, 3
3. Aleksandr, Oktyabria, 4
4. Nadya, sad, 3
5. kjf, kjf, 1

В это же время вывод 2 экземпляра программы:

Commands:

ADD - add a new record

LST - list all records

GET <record number> - get record with number n

MOD - modify an existing record

PUT - save modified record

QUIT - exit the program

> MOD

Nothing to edit. Call GET function!

> GET 3

Record #3

Name:
Address:
Semester: 0

> MOD 3
Enter name: df
Enter address: df
Enter semester: 3
Data has been updated, please try again!
Enter name: Aleksandr
Enter address: Oktyabria
Enter semester: 4
Record modified at position 3

> LST
Records:
0. alecs, add, 1
1. Pasha, dfj, 1
2. Galina, fff, 3
3. alecsandr, okt, 4
4. Nadya, sad, 3
5. kjf, kjf, 1
> PUT
Record #3 saved

> GET 5
Record #5
Name: kjf
Address: kjf
Semester: 1

> MOD
Enter name: Alice
Enter address: ggg
Enter semester: 123
Record modified at position 5

> LST
Records:
0. alecs, add, 1
1. Pasha, dfj, 1
2. Galina, fff, 3
3. Aleksandr, Oktyabria, 4
4. Nadya, sad, 3
5. kjf, kjf, 1

> PUT
Record #5 saved

> LST

Records:

- 0. alecs, add, 1
- 1. Pasha, dfj, 1
- 2. Galina, fff, 3
- 3. Aleksandr, Oktyabria, 4
- 4. Nadya, sad, 3
- 5. Alice, ggg, 123