

**«Санкт-Петербургский государственный электротехнический  
университет «ЛЭТИ» им. В.И.Ульянова (Ленина)»  
СПбГЭТУ «ЛЭТИ»**

Кафедра вычислительной техники

Отчет по по лабораторной работе №4 по дисциплине  
«Организация процессов и программирование в среде Linux»  
Тема: «УПРАВЛЕНИЕ ПОТОКАМИ»

Студент гр.8306

Преподаватель

---

---

Слепов А.Э.

Разумосвский Г.В.

Санкт-Петербург

2021

## Цель работы

Знакомство с организацией потоков и способами синхронизации предков и потомков

## Задание

1. Написать программу, которая открывает входной файл и 2 выходных файла. Затем она должна в цикле построчно читать входной файл и порождать 2 потока. Одному потоку передавать нечетную строку, а другому – четную. Оба потока должны работать параллельно. Каждый поток записывает в свой выходной файл полученную строку и завершает работу. Программа должна ожидать завершения работы каждого потока и повторять цикл порождения потоков и чтения строк входного файла, пока не прочтет последнюю строку, после чего закрыть все файлы.

2. Откомпилировать программу и запустить ее

## Порядок выполнения работы

Запуск программы для входного файла с 20 строками и результат работы программы представлен на рисунке 1.

```
artem@legion:~/Документы/linux/lab4$ ./a.out input.txt
Read file start
Read file end
artem@legion:~/Документы/linux/lab4$ cat output1.txt
String 1
String 3
String 5
String 7
String 9
String 11
String 13
String 15
String 17
String 19
artem@legion:~/Документы/linux/lab4$ cat output2.txt
String 2
String 4
String 6
String 8
String 10
String 12
String 14
String 16
String 18
String 20
artem@legion:~/Документы/linux/lab4$
```

Рисунок 1. Первый запуск программы

Запуск программы для входного файла с 3 строками и результат работы программы представлен на рисунке 2.

```
String 3
artem@legion:~/Документы/linux/lab4$ ./a.out input_1.txt
Read file start
Read file end
artem@legion:~/Документы/linux/lab4$ cat output1.txt
String 1
String 3
artem@legion:~/Документы/linux/lab4$ cat output2.txt
String 2
artem@legion:~/Документы/linux/lab4$
```

Рисунок 2. Второй запуск программы

Запуск программы для пустого входного файла и результат работы программы представлен на рисунке 2.

```
artem@legion:~/Документы/linux/lab4$ ./a.out input_3.txt
Read file start
Read file end
artem@legion:~/Документы/linux/lab4$ cat output1.txt
artem@legion:~/Документы/linux/lab4$ cat output2.txt
artem@legion:~/Документы/linux/lab4$
```

Рисунок 3. Третий запуск программы (для пустого входного!)

## Выводы

В ходе работы были изучены механизмы создания и управления потоков на примере параллельного переписывания входного файла в 2 выходных.

## ПРИЛОЖЕНИЕ А

### Текст программы

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

#define BUF_SIZE 256

typedef struct thread_args_{
    FILE *file;
    char *str;
}thread_args;

void *thread_routine(void *args){
    thread_args *arg = (thread_args*)args;
    fputs(arg->str, arg->file);
}

int main(int argc, char **argv){
    FILE *inputFile;
    if(argc == 2)
        inputFile = fopen(argv[1], "r");
    else
        inputFile = fopen("input.txt", "r");

    thread_args thr_arg_1, thr_arg_2;
    thr_arg_1.file = fopen("output1.txt", "w");
    thr_arg_2.file = fopen("output2.txt", "w");
    char eofFlag = 0;
    char buf1[BUF_SIZE], buf2[BUF_SIZE];
    pthread_t thread_1 = 0, thread_2 = 0;

    if(inputFile && thr_arg_1.file && thr_arg_2.file){
        printf("Read file start\n");
        while(!feof(inputFile)){
            if(thr_arg_1.str = fgets(buf1, BUF_SIZE, inputFile)){
                if(pthread_create(&thread_1, NULL, &thread_routine,
&thr_arg_1))
                    printf("Thread_1 create ERROR");

                if(thr_arg_2.str = fgets(buf2, BUF_SIZE, inputFile)){
                    if(pthread_create(&thread_2, NULL, &thread_routine,
&thr_arg_2))
                        printf("Thread_2 create ERROR");

                    pthread_join(thread_1, NULL);
                    pthread_join(thread_2, NULL);

                    if(feof(inputFile))
                        printf("Read file end\n");
                }
            }
            fclose(inputFile);
            fclose(thr_arg_1.file);
            fclose(thr_arg_2.file);
        }
    }
    else{
        printf("Open file error\n");
    }
    return 0;
}
```

## **ПРИЛОЖЕНИЕ Б**

### **Текст input.txt**

String 1  
String 2  
String 3  
String 4  
String 5  
String 6  
String 7  
String 8  
String 9  
String 10  
String 11  
String 12  
String 13  
String 14  
String 15  
String 16  
String 17  
String 18  
String 19  
String 20

**ПРИЛОЖЕНИЕ В**  
**Текст input.txt**

String 1  
String 2  
String 3