## МИНОБРНАУКИ РОССИИ

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

## «ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)

**Кафедра Вычислительной техники**

## ОТЧЕТ

**по лабораторной работе №4**

## по дисциплине «Организация процессов и программирование в среде Linux»

**Тема: Управление потоками**

|  |  |
| --- | --- |
| Студент гр. 8308 | Петров Г.А. |
| Преподаватель | Разумовский Г.В. |

Санкт-Петербург 2021

# Цель работы

Целью лабораторной работы является знакомство с организацией потоков и способами синхронизации предков и потомков.

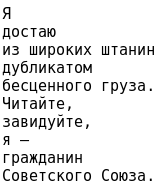
# Задание

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. Откомпилировать программу и запустить ее.

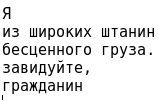
# Примеры выполнения программы

Программа была разработана таким образом, что программа считывает информацию из файла “Text.txt”, а затем порожденные потоки 1 и 2 выводят информацию в файлы “Thread1.txt” и “Thread2.txt” соответственно. Выполним программу предварительно введя текст в файл “Text.txt”.

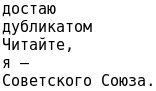
Распечатка файла “Text.txt”:



Распечатка файла “Thread1.txt”:



Распечатка файла “Thread2.txt”:



# Исходный код программ

## lab4.cpp

#include <iostream>

#include <fstream>

#include <pthread.h>

//создаем и открываем потоки ввода и вывода в файлы

std::ofstream Output1("Thread1.txt");

std::ofstream Output2("Thread2.txt");

std::ifstream Input("Text.txt");

//работа первого потока

void\* work\_thread1(void\* \_arg)

{

std::string arg=\*static\_cast<std::string\*>(\_arg);

Output1<<arg<<std::endl;//вывод строки в файл

pthread\_exit(NULL);//окончание потока

}

//работа второго потока

void\* work\_thread2(void\* \_arg)

{

std::string arg=\*static\_cast<std::string\*>(\_arg);

Output2<<arg<<std::endl;//вывод в файл

pthread\_exit(NULL);//конец работы потока

}

int main()

{

pthread\_t thread1,thread2;//объявление потоков

std::string str;//строка чтения

while(true)

{

std::string str1,str2;

bool end\_flag1=false;

bool end\_flag2=false;

//создание потока (ид-тор потока, атрибуты, функция, аргументы)

if(std::getline(Input,str1))

pthread\_create(&thread1,NULL,work\_thread1,&str1);

else

end\_flag1=true;

if(std::getline(Input,str2))

pthread\_create(&thread2,NULL,work\_thread2,&str2);

else

end\_flag2=true;

//ожидание завершения потока (иден-тор потка, код завершения)

pthread\_join(thread1,NULL);

pthread\_join(thread2,NULL);

if(end\_flag1||end\_flag2)//выход из цикла

{

std::cout<<"Окончание чтения из файла"<<std::endl;

break;

}

}

//закрытие файловых потоков

Output1.close();

Output2.close();

Input.close();

return 0;

}

# Вывод

При выполнении лабораторной работы изучены и использованы системные функции, обеспечивающие порождение и идентификацию потоков; разработана программа, порождающая 2 потока и выводящая четные и нечетные строки в соответствующие им файлы.