Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №3 по курсу**

**«Операционные системы»**

**Тема работы**

**“Потоки”**

Студент: Курихин Максим Викторович

Группа: М8О-206Б-20

Вариант: 18

Преподаватель: Миронов Евгений Сергеевич

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2022

**Содержание**

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

**Репозиторий**

<https://github.com/Kurmaxim/OS>

**Постановка задачи**

Программа должна обрабатывать данные в многопоточном режиме, используя стандартные средства создания потоков операционной системы Unix. Ограничение потоков должно быть задано ключом запуска программы.

Найти образец в строке наивным алгоритмом.

**Общие сведения о программе**

Программа написана на языке Си в UNIX-подобной операционной системе. Для компиляции требуется указать ключ –pthread. Для запуска программы в качестве первого аргумента командной строки необходимо указать количество потоков.

Сборка проекта происходит при помощи make-файла  
gсс –pthread os\_lab3.c

**Общий метод и алгоритм решения**

Создаем отдельную функцию, которая выполняет функцию поиска подстроки в строке, в коде самой программы распараллеливаем программу по строке.

В программе, используются две функции, связанные с потоками:  
pthread\_create – создание потока.

pthread\_join(pthread\_t thread, void \*\*value\_ptr)- Откладывает выполнение вызывающего (эту функцию) потока, до тех пор, пока не будет выполнен поток thread.

**Исходный код**

**os\_lab3.c**

**#include <iostream>**

**#include <pthread.h>**

**using namespace std;**

**string str, templ;**

**int N;**

**typedef struct arguments {**

**int start;**

**int stop;**

**} Arg;**

**void \*thread\_function(void \*args) {**

**Arg \*arg = (Arg \*)args;**

**int start = arg->start;//начало скобочки**

**int stop = arg->stop;//конец скобочки) полуинтервал [ )**

**for (int j = start; j < stop; ++j) {**

**bool k = true;**

**for (int i = 0; i < N; ++i) {**

**if (str[j + i] != templ[i]) {**

**k = false;**

**}**

**}**

**if (k) {**

**cout << "Substring starts at " << j << endl;**

**}**

**}**

**return nullptr;**

**}**

**int main() {**

**int threads\_num;**

**cin >> str;**

**cin >> templ;**

**cin >> threads\_num;**

**N = (int)templ.size();**

**auto \*threads = new pthread\_t[threads\_num];**

**int points\_for\_thread = ((int)str.size()) / threads\_num;**

**int num\_created\_threads = 0;**

**Arg a;**

**bool kostil = false;**

**a.start = a.stop = 0;**

**for (int i = 0; (i < threads\_num) && !kostil; i++) { //создает потоки**

**a.start = a.stop;**

**a.stop = a.start + points\_for\_thread + 1;**

**if (a.stop >= str.size()) {**

**a.stop = ((int)str.size()) - N + 1;**

**kostil = true;**

**}**

**if (pthread\_create(&threads[i], nullptr, thread\_function, &(a)) != 0) {**

**cout << "Can not create thread" << endl;**

**exit(1);**

**}**

**if (pthread\_join(threads[i], nullptr) != 0) {**

**cout << "Join error" << endl;**

**exit(1);**

**}**

**++num\_created\_threads;**

**}**

**cout << num\_created\_threads << " threads was created" << endl;**

**delete threads;**

**return 0;**

**}**

**Демонстрация работы программы**

**Kurmaxim@DESKTOP-OL36FK8:/mnt/c/Users/max-/Desktop/os\_lab3/src$ g++ os3.cpp -pthread -o lol**

**Kurmaxim@DESKTOP-OL36FK8:/mnt/c/Users/max-/Desktop/os\_lab3/src$ ./lol**

**kek o**

**1**

**Substring starts at 1**

**1 threads was created**

**Kurmaxim@DESKTOP-OL36FK8:/mnt/c/Users/max-/Desktop/os\_lab3/src$ ./lol**

**charmasica mas 1**

**Substring starts at 4**

**1 threads was created**

**Kurmaxim@DESKTOP-OL36FK8:/mnt/c/Users/max-/Desktop/os\_lab3/src$ ./lol**

**cdwdaopwdkawop aop 5**

**Substring starts at 4**

**5 threads was created**

**Kurmaxim@DESKTOP-OL36FK8:/mnt/c/Users/max-/Desktop/os\_lab3/src$**

**Выводы**

Эта лабораторная работа ознакомила и научила меня работать с потоками. Я изучил основные функции для работы с потоками в Си, как совершать системные запросы на создание потока, ожидание завершения потока. Я изучал понятие “многопоточность”. Она позволяет ускорить обработку данных в программе