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: Молчанов Владислав Дмитриевич
Группа: М8О-208Б-20
Вариант: 18
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2021

Содержание

Репозиторий

https://github.com/molch4nov/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

gcc –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

vladislav@DESKTOP-OL36FK8:/mnt/c/Users/vlad-/Desktop/os_lab3/src$ g++ os3.cpp -pthread -o lol

vladislav@DESKTOP-OL36FK8:/mnt/c/Users/vlad-/Desktop/os_lab3/src$ ./lol

jopa o

1

Substring starts at 1

1 threads was created

vladislav@DESKTOP-OL36FK8:/mnt/c/Users/vlad-/Desktop/os_lab3/src$ ./lol

charmasica mas 1

Substring starts at 4

1 threads was created

vladislav@DESKTOP-OL36FK8:/mnt/c/Users/vlad-/Desktop/os_lab3/src$ ./lol

cdwdaopwdkawop aop 5

Substring starts at 4

5 threads was created

vladislav@DESKTOP-OL36FK8:/mnt/c/Users/vlad-/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