Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу

«Операционные системы»

Многопоточие

Студент: [ Нелюбин В.С ]

Группа: М8О–206Б–19

Вариант: 18

Преподаватель: Соколов Андрей Алексеевич

Оценка: \_\_\_\_\_\_\_\_\_\_

Дата: \_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_

Москва, 2020.

Постановка задачи

**Цель работы**

Приобретение практических навыков в:

* Управление потоками в ОС
* Обеспечение синхронизации между потоками

## Задание

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработке использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение потоков должно быть задано ключом запуска вашей программы.

Так же необходимо уметь продемонстрировать количество потоков, используемое вашей программой с помощью стандартных средств операционной системы.

В отчете привести исследование зависимости ускорения и эффективности алгоритма от входящих данных и количества потоков. Получившиеся результаты необходимо объяснить.

Найти образец в строке наивным алгоритмом

**Общие сведения о программе**

Программа компилируется из файла thr.c. Также используется заголовочные файлы: stdio.h stdlib.h и pthread.h. Основные команды программы:

1. **pthread\_create** – Создаёт новый поток, исполняющий указанную в аргументах функцию.
2. **pthread\_join** – ждёт, пока поток не готов к завершению, а затем прекращает его.

Для проверки времени работы используется команда bash **time**

Для проверки количества используемых потоков используется утилита **top   
-H**

Программа получает на вход 3 аргумента: максимальное количество потоков, строка, и подстрока, вхождения которой нужно найти. Если количество аргументов больше, программа будет вести себя специальным образом. Все потоки будут входить в бесконечный while цикл. Это делает проверку числа потоков легче.

**Общий метод и алгоритм решения**.

Для реализации поставленной задачи необходимо:

1. Изучить принципы работы потоков.
2. Придумать, как использовать потоки для решения данной задачи.
3. Написать саму программу.
4. Найти средства измерения времени и количества потоков программы

**Основные файлы программы**

**thr.c:**

#include <stdio.h>

#include <stdlib.h>

#include <pthread.h>

typedef struct cci{

char\*r;

char\*w;

int o;

}cci;

void\* isSubstr(void\*vdi){

cci\*inpt=(cci\*)vdi;

while(inpt->o==-1){}

for(int i=0;inpt->w[i]!='\0';i++){

if(inpt->r[inpt->o+i]!=inpt->w[i]){

inpt->o=-1;

break;

}

}

if(inpt->o!=-1){

printf("sample found at %d\n",inpt->o);

}

}

int main(int argc, char \*argv[]){

int limt,strl,smpl;

if(argc < 4) {

printf("bad command line arguments :(\n");

return -1;

}

for(smpl=0;argv[3][smpl]!='\0';smpl++);

for(strl=0;argv[2][strl]!='\0';strl++);

if(smpl>strl){printf("sample is bigger than the line. very bad\n");return -1;}

limt=0;

for(int i=0;argv[1][i]!='\0';i++){

limt=limt\*10+argv[1][i]-'0';

}

printf("max threads = %d\nline = %s\nsample = %s\n",limt,argv[2],argv[3]);

pthread\_t\*thrds=malloc(sizeof(pthread\_t)\*limt);

int rertn,ei,ii;

cci\*\*fori=malloc(sizeof(cci\*)\*limt);

for(ei=0;ei<limt;ei++){

fori[ei]=malloc(sizeof(cci));

}

for(ei=0;ei<=strl-smpl;ei++){

ii=ei%limt;

if(ei>=limt){

if(pthread\_join( thrds[ii], NULL)!=0){printf("join error at i=%d\n",ii);}

}

fori[ii]->r=argv[2];

fori[ii]->w=argv[3];

fori[ii]->o=ei;

if(argc>4){fori[ii]->o=-1;}

if(pthread\_create( &thrds[ii], NULL, isSubstr, (void\*)fori[ii])!=0){printf("call error at i=%d\n",ii);}

}

ii++;

if(ei!=ei%limt){

ii=limt;

}

for(int j=0;j<ii;j++){

if(pthread\_join( thrds[j], NULL)!=0){printf("join error at i=%d\n",j);}

}

for(ei=0;ei<limt;ei++){

free(fori[ei]);

}

free(fori);

free(thrds);

return 0;

}

**Пример работы**

aaaaa@TRASHBox:~/Desktop/OC/laba3$ ./a.out 12 blablablablawequhgbalalaika ala

max threads = 12

line = blablablablawequhgbalalaika

sample = ala

sample found at 21

sample found at 19

aaaaa@TRASHBox:~/Desktop/OC/laba3$ time ./a.out 8 blablablablawequhgbalalaika ala

max threads = 8

line = blablablablawequhgbalalaika

sample = ala

sample found at 21

sample found at 19

real 0m0,003s

user 0m0,000s

sys 0m0,002s

aaaaa@TRASHBox:~/Desktop/OC/laba3$ time ./a.out 16 blablablablawequhgbalalaika ala

max threads = 16

line = blablablablawequhgbalalaika

sample = ala

sample found at 21

sample found at 19

real 0m0,004s

user 0m0,000s

sys 0m0,002s

**Вывод**

Количество потоков не повлияло на скорость выполнения программы, поскольку доступное устройство имеет одноядерный процессор, и все потоки работают поочерёдно, а не параллельно.