Московский Авиационный Институт

(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

**Лабораторная работа №2 по курсу**

**«Операционные системы»**

**Управление потоками в ОС**

Студент: Речинская Ангелина Юрьевна

Группа: М80-206Б-20

Вариант: 8

Преподаватель: Соколов Андрей Алексеевич

Дата: 21.12.2021

Оценка: 5

Подпись: \_\_\_\_\_\_\_\_\_\_\_

Москва, 2021

**Постановка задачи**

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.

8. Есть К массивов одинаковой длины. Необходимо сложить эти массивы. Необходимо предусмотреть стратегию, адаптирующуюся под количество массивов и их длину (по количеству операций)

**Листинг программы**

**lab3.c**

#include <pthread.h>

#include <stdlib.h>

#include <mcheck.h>

#include <stdio.h>

#include <stdbool.h>

#include <stdint.h>

#include <inttypes.h>

typedef struct{

int cnt;

int \*arr;

} pthrData;

int\* Sum;

#define memory 724

#define thread\_size 4

void\* threadFunc(void\* thread\_data){

pthrData \*data = (pthrData\*) thread\_data;

for(int i=0;i<data->cnt;i++){

Sum[i]=Sum[i]+data->arr[i];

}

return NULL;

}

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

if(argc!=2){

fprintf(stderr, "Неверное количество аргументов\n");

return 1;

}

int n,m;

scanf("%d%d",&n,&m);

long long p=atoi(argv[1]);

if(p<1){

fprintf(stderr, "Количество памяти не может быть меньше единицы\n");

return 2;

}

p=p-memory-thread\_size\*(n-1);

if(p<0){

fprintf(stderr, "Количество памяти слишком мало\n");

return 3;

}

int Arr[n][m];

for(int i=0;i<n;i++){

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

scanf("%d",&Arr[i][j]);

}

}

pthread\_t threads[n];

pthrData threadData[n];

Sum=malloc(sizeof(int)\*m);

for(int i=0;i<m;i++){

Sum[i]=0;

}

for(int i=0;i<n;i++){

threadData[i].cnt=m;

threadData[i].arr=malloc(sizeof(int)\*m);

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

threadData[i].arr[j]=Arr[i][j];

}

pthread\_create(&(threads[i]), NULL, threadFunc, &threadData[i]);

}

for(int i=0;i<n;i++){

pthread\_join(threads[i], NULL);

}

for(int i=0;i<m;i++){

printf("%d ",Sum[i]);

}

printf("\n");

}

**Примеры работы**

salutik@salutik:~/un/OS/lab3$ ./lab3 800

4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

1 2 3 4 5

4 8 12 16 20

**Вывод**

В процессе работы над лабораторной я научилась основам работы потоками в Си. К сожалению, в процессе работы не возникло потребности использования мьютексов, семафоров и т. п., так что нельзя сказать, что материал был освоен полностью на практике.