Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу
«Операционные системы»

Студент: Казацнев Данила Игоревич
Группа: М8О-207Б-21
Вариант: 1
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2022

Содержание

Репозиторий

https://github.com/thgdanilaya/mai_os_labs/lab2

Постановка задачи

Цель работы

Целью является приобретение практических навыков в:

Управление потоками в ОС

Обеспечение синхронизации между потоками

Задание

Отсортировать массив целых чисел при помощи битонической сортировки

Общий метод и алгоритм решения

На вход подается кол-во потоков, потом такое же кол-во массивов чисел(строго степень двойки).

Дальше сортируется методом битоннической сортировки

Исходный код

```
#include "time.h"
#include "stdio.h"
#include "stdlib.h"
#include "pthread.h"

#define MAX 10

int up = 1;
int down = 0;

void sswap(int *num1, int *num2) {
    int temp;

    temp = *num1;
    *num1 = *num2;
    *num2 = temp;
}

void compare(int *arr, int i, int j, int dir) {
    if (dir == (arr[i] > arr[j])) {
        sswap(&arr[i], &arr[j]);
    }
}

void bitonicmerge(int *arr, int low, int c, int dir) {
    int k;
    int i;

    if (c > 1) {
        k = c / 2;
        i = low;
        while (i < low + k) {
            compare(arr, i, i + k, dir);
            i++;
        }

        bitonicmerge(arr, low, k, dir);
```

```
        bitonicmerge(arr, low + k, k, dir);
    }
}

void recbitonic(int *arr, int low, int v, int dir) {
    int k;

    if (v > 1) {
        k = v / 2;
        recbitonic(arr, low, k, up);
        recbitonic(arr, low + k, k, down);
        bitonicmerge(arr, low, v, dir);
    }
}


void *routine(void *arg) {
    int *mass = (int *) arg;
    recbitonic(mass, 0, MAX, up);
    for (int i = 0; i < MAX; i++)
        printf("%d ", mass[i]);
    printf("\n");
    free(arg);
    return NULL;
}

int main(int argc, char *argv[]) {
    int n = atoi(argv[1]);
    pthread_t pid[n];

    int data[n][MAX];
    printf("Enter %d massives!\n", n);
    for (int i = 0; i < n; i++) {
        for (int u = 0; u < MAX; u++)
            scanf("%d", data[i] + u);
    }

    double start = clock();;

    if (argc == 2) {
        for (int i = 0; i < n; i++) {
            int *data2 = malloc(sizeof(int) * MAX);
            for (int u = 0; u < MAX; u++) {
                data2[u] = data[i][u];
            }
            if (pthread_create(&pid[i], NULL, &routine, data2) != 0) {
                perror("Couldn't create a thread\n");
                return 1;
            }
            printf("Thread %d has started\n", i + 1);
        }
        for (int i = 0; i < n; i++) {
            if (pthread_join(pid[i], NULL) != 0) {
                return 2;
            }
            printf("Thread has finished execution!\n");
        }
    } else {
        printf("Please enter an appropriate program key !\n");
    }
    printf("Count of threads: %d\n", n);
    printf("The program ran for %.4lf seconds\n", (clock() - start) /
(CLOCKS_PER_SEC));
```

```
    return 0;
}
```

Демонстрация работы программы

```
danilaya@DESKTOP-JFEGEK0:/mnt/c/Users/frede/CLionProjects/oslab3$ ./lab2 4
Enter 4 massives!
9 8 7 6
6 3 -8 9
1 2 3 4
5 0 -2 1
Thread 1 has started
6 7 8 9
Thread 2 has started
-8 3 6 9
Thread 3 has started
1 2 3 4
Thread 4 has started
Thread has finished execution!
Thread has finished execution!
Thread has finished execution!
-2 0 1 5
Thread has finished execution!
Count of threads: 4
The program ran for 0.0009 seconds
```

Выводы

Я приобрел навыки в управлении потоками и обеспечении синхронизации между потоками.