МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ

ФЕДЕРАЦИИ

Московский Авиационный Институт

(Национальный Исследовательский Университет)

Институт №8 "Компьютерные науки и прикладная математика"

Кафедра 806 "Вычислительная математика и программирование"

Лабораторная работа №2

По курсу «Операционные системы»

Студент: Власко М. М.

Группа: М8О-208Б-23

Преподаватель: Живалев Е. А.

Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2024

**Тема:** Управление потоками и синхронизация в ОС

**Цель работы:** Целью работы является приобретение практических навыков в:

* Управлении потоками в операционной системе.
* Организации синхронизации между потоками для эффективного использования многопоточности.

**Вариант:** 3. Отсортировать массив целых чисел при помощи параллельной сортировки слиянием.

**Задачи:**

1. Разработать программу на языке Си, реализующую многопоточную сортировку массива целых чисел методом слияния.
2. Ограничить максимальное количество одновременно работающих потоков с использованием заданного параметра.
3. Обеспечить корректную синхронизацию потоков с помощью стандартных средств операционной системы.
4. Провести исследование зависимости ускорения и эффективности алгоритма от входных данных и количества потоков.

**Описание решения:** Программное решение представляет собой многопоточную реализацию сортировки массива методом слияния. Основные компоненты программы:

* **Управление потоками:** Для создания потоков используется библиотека pthread. Максимальное количество одновременно работающих потоков задается пользователем в виде параметра запуска программы.
* **Синхронизация потоков:** Для управления количеством активных потоков используется мьютекс и глобальная переменная countOfActiveThreads. Увеличение и уменьшение счетчика активных потоков синхронизировано с помощью мьютекса.
* **Алгоритм сортировки:** Сортировка массива выполняется рекурсивно. При каждом делении массива создается новый поток, если количество активных потоков меньше заданного максимума. В противном случае обработка выполняется в текущем потоке.

Программа функционирует следующим образом:

1. Пользователь задает максимальное количество потоков и тип ввода данных (вручную или случайная генерация).
2. Если выбран ручной ввод, пользователь вводит размер массива и его элементы. В случае случайной генерации массив заполняется случайными числами.
3. Основной поток вызывает функцию сортировки, передавая в нее данные о массиве.
4. В процессе сортировки массив делится на части, которые обрабатываются либо в новых потоках, либо в текущем потоке, в зависимости от текущей загрузки.
5. После завершения сортировки выводится отсортированный массив.

**Репозиторий:** https://github.com/MMVlasko/mai-os-lab-work/tree/main/LW2

**Исходный код:** Программное обеспечение состоит из следующих файлов:

1. **main.c:** Инициализация программы, ввод данных и запуск сортировки.
2. **utils.c:** Реализация функций сортировки слиянием и управления потоками.

Пример кода:

// Функция слияния массивов

void Merge(int \*array, const int left, const int mid, const int right) {

int leftSize = mid - left + 1;

int rightSize = right - mid;

int \*leftArray = malloc(leftSize \* sizeof(int));

int \*rightArray = malloc(rightSize \* sizeof(int));

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

leftArray[i] = array[left + i];

}

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

rightArray[i] = array[mid + 1 + i];

}

int i = 0, j = 0, k = left;

while (i < leftSize && j < rightSize) {

if (leftArray[i] <= rightArray[j]) {

array[k++] = leftArray[i++];

} else {

array[k++] = rightArray[j++];

}

}

while (i < leftSize) {

array[k++] = leftArray[i++];

}

while (j < rightSize) {

array[k++] = rightArray[j++];

}

free(leftArray);

free(rightArray);

}

**Пример работы:**

./lab2

Использование: ./lab2 <максимальное количество потоков> <i/r>

i - ввести произвольный массив

r - сгенерировать случайный массив

./lab2 20 i

Введите количество элементов в массиве: 5

Введите элементы массива:

5

4

3

2

1

Отсортированный массив:

1

2

3

4

5

**Вывод:** В ходе выполнения лабораторной работы была реализована многопоточная сортировка массива методом слияния. Программа корректно ограничивает количество одновременно работающих потоков, обеспечивая при этом эффективное использование многопоточности.

Результаты исследования зависимости ускорения и эффективности алгоритма от количества потоков и объема данных показали, что увеличение числа потоков до определенного предела ускоряет выполнение программы. Однако при чрезмерном увеличении числа потоков эффективность снижается из-за накладных расходов на управление потоками. Полученные результаты соответствуют теоретическим ожиданиям.