

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: Григорьев Т. А.

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: 1 Отсортировать массив целых чисел при помощи битонической сортировки

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 битонической сортировки массива. 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/timofeez/os-labs/tree/main/02>

Исходный код: Программное обеспечение состоит из следующих файлов:

1. **main.c:** Инициализация программы, ввод данных и запуск сортировки.
2. **bitonic_data.c:** Определение структуры битонических данных, различные методы манипуляции с ними
3. **bitonic_sort.c:** Все функции, непосредственно связанные с сортировкой
4. **globals.c** Определение глобальных переменных

Пример кода:

```
// Функция битонического слияния

void bitonic_merge(int *array, int low, int count, int direction) {

    if (count > 1) {

        int mid = count / 2;

        for (int i = low; i < low + mid; i++) {

            compare_and_swap(array, i, i + mid, direction);

        }

        bitonic_merge(array, low, mid, direction);

        bitonic_merge(array, low + mid, mid, direction);

    }

}
```

```

}

void* bitonic_sort_thread(void* arg) {

    BitonicData* data = (BitonicData*)arg;

    bitonic_sort_recursive(data->array, data->low, data->count, data->direction);

    pthread_mutex_lock(&mutex);

    countOfActiveThreads--;

    pthread_mutex_unlock(&mutex);

    free(data);

    return NULL;
}

```

} 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 рамках выполнения лабораторной работы была разработана программа, выполняющая сортировку массива при помощи битонической сортировки с использованием многопоточности. Реализация предусматривает ограничение числа одновременно работающих потоков, что позволяет эффективно использовать ресурсы многопроцессорной системы.

В ходе исследования была изучена зависимость ускорения и эффективности работы алгоритма от количества потоков и размера данных. Результаты показали, что увеличение числа потоков способствует ускорению программы до определенного предела. Однако при избыточном увеличении количества потоков эффективность начинает снижаться из-за дополнительных затрат на управление потоками. Полученные данные подтвердили теоретические предположения.