

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»

Группа: М8О-210Б-23

Студент: Коваленко Д.А

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 17.11.24

Москва, 2024

Постановка задачи

Вариант 3.

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработки использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение максимального количества потоков, работающих в один момент времени, 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 входных данных и количества потоков. Получившиеся результаты необходимо объяснить.

Отсортировать массив целых чисел при помощи параллельной сортировки слиянием

Общий метод и алгоритм решения

Использованные системные вызовы:

- `pthread_join()` – ожидание завершения потока.
- `pthread_mutex_init()` – инициализация мьютекса.
- `pthread_mutex_lock()` – блокировка мьютекса.
- `pthread_mutex_unlock()` – разблокировка мьютекса.
- `pthread_mutex_destroy()` – уничтожение мьютекса.
- `gettimeofday()` – получение текущего времени.
- `write()` – запись данных в стандартный вывод (консоль).
- **`pthread_create()`** – создание нового потока.

Описание алгоритма:

Алгоритм является реализацией классической сортировки слиянием, однако с повышением производительности за счёт применения параллельного разделения массива на части и их обратного слияния. Для работы с потоками и синхронизации их работы используются `pthread`, мьютексы и семафоры.

Описание кода:

Вывод в stdout:

Для вывода данных используется функция `writeToStdout`, которая выполняет прямой системный вызов `write` для записи в `stdout`. Это позволяет избежать использования библиотеки `stdio`, обеспечивая большую прозрачность и упрощая отладку.

Вывод массива:

Вывод массива в `stdout` обернут в мьютекс (`pthread_mutex_lock`), что исключает рассогласованные выходы при работе нескольких потоков.

Сортировка слиянием:

- Функция `mergeArrays` сливает два отсортированных подмассива в общий.
- Функция `mergeSort` выполняет рекурсивную сортировку, разделяя массив на подмассивы и запуская отдельные потоки для каждой части.
- Для ожидания завершения потоков используется `pthread_join`.

Код программы

main/main.cpp

```
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <pthread.h>
#include <semaphore.h>
#include <sys/time.h>
#include <unistd.h>
#include <vector>

pthread_mutex_t ioMutex;
sem_t threadLimit;

struct ThreadData {
    std::vector<int> *array;
    int left;
    int right;
};
```

```
void writeToStdout(const char *str) { write(STDOUT_FILENO, str, strlen(str)); }
```

```
void printArray(const std::vector<int> &array) {  
    pthread_mutex_lock(&ioMutex);  
    for (size_t i = 0; i < array.size(); ++i) {  
        char buffer[16];  
        snprintf(buffer, sizeof(buffer), "%d ", array[i]);  
        writeToStdout(buffer);  
    }  
    writeToStdout("\n");  
    pthread_mutex_unlock(&ioMutex);  
}
```

```
void mergeArrays(std::vector<int> &array, int left, int mid, int right) {  
    int n1 = mid - left + 1;  
    int n2 = right - mid;
```

```
    std::vector<int> leftArray(n1), rightArray(n2);  
    for (int i = 0; i < n1; ++i)  
        leftArray[i] = array[left + i];  
    for (int i = 0; i < n2; ++i)  
        rightArray[i] = array[mid + 1 + i];
```

```
    int i = 0, j = 0, k = left;  
    while (i < n1 && j < n2) {  
        if (leftArray[i] <= rightArray[j]) {  
            array[k] = leftArray[i];  
            ++i;  
        } else {  
            array[k] = rightArray[j];  
            ++j;  
        }  
        ++k;  
    }
```

```
    while (i < n1) {  
        array[k] = leftArray[i];
```

```
    ++i;  
    ++k;  
}
```

```
while (j < n2) {  
    array[k] = rightArray[j];  
    ++j;  
    ++k;  
}  
}
```

```
void *mergeSort(void *arg) {  
    ThreadData *data = (ThreadData *)arg;  
    std::vector<int> &array = *(data->array);  
    int left = data->left;  
    int right = data->right;
```

```
    char debugBuffer[64];  
    if (left >= right) {  
        sem_post(&threadLimit);  
        return nullptr;  
    }
```

```
    int mid = left + (right - left) / 2;
```

```
    pthread_t leftThread, rightThread;  
    ThreadData leftData = {&array, left, mid};  
    ThreadData rightData = {&array, mid + 1, right};
```

```
    if (pthread_create(&leftThread, nullptr, mergeSort, &leftData) != 0) {  
        writeToStdout(  
            "Failed to create left thread, falling back to single-threaded\n");  
        mergeSort(&leftData);  
        sem_post(&threadLimit);  
    }
```

```
    if (pthread_create(&rightThread, nullptr, mergeSort, &rightData) != 0) {
```

```

        writeToStdout(
            "Failed to create right thread, falling back to single-threaded\n");
        mergeSort(&rightData);
        sem_post(&threadLimit);
    }

    pthread_join(leftThread, nullptr);
    pthread_join(rightThread, nullptr);
    mergeArrays(array, left, mid, right);

    sem_post(&threadLimit);
    return nullptr;
}

int main(int argc, char **argv) {
    if (argc != 3) {
        writeToStdout("Usage: ./mergeSort <maxThreads> <arraySize>\n");
        return 1;
    }

    int maxThreads = std::atoi(argv[1]);
    int arraySize = std::atoi(argv[2]);

    if (maxThreads < 1 || arraySize < 1) {
        writeToStdout("Both maxThreads and arraySize must be positive integers.\n");
        return 1;
    }

    pthread_mutex_init(&ioMutex, nullptr);
    sem_init(&threadLimit, 0, maxThreads);

    std::srand(time(nullptr));
    std::vector<int> array(arraySize);

    for (int &x : array)
        x = std::rand() % 100;

```

```

writeToStdout("Unsorted array:\n");
printArray(array);

ThreadData initialData = {&array, 0, arraySize - 1};

struct timeval start, end;
gettimeofday(&start, nullptr);

sem_wait(&threadLimit);
mergeSort(&initialData);

gettimeofday(&end, nullptr);

double elapsedTime =
    (end.tv_sec - start.tv_sec) + (end.tv_usec - start.tv_usec) / 1e6;

writeToStdout("Sorted array:\n");
printArray(array);

char buffer[64];
snprintf(buffer, sizeof(buffer), "Time elapsed: %.6f seconds\n", elapsedTime);
writeToStdout(buffer);

pthread_mutex_destroy(&ioMutex);
sem_destroy(&threadLimit);

return 0;
}

```

Протокол работы программы

```
strace ./mergeSort 1 20
```

```
execve("./mergeSort", [ "./mergeSort", "1", "20"], 0x7ffc707a16c0 /* 89 vars */) = 0
```

```
brk(NULL) = 0x600a3facc000
```

```
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc6432afa0) = -1 EINVAL (Invalid argument)
```

```
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|
MAP_ANONYMOUS, -1, 0) = 0x7563329bf000
```

```
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
```

```
openat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/glibc-
hwcaps/x86-64-v3/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
```

```
newfstatat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/glibc-
hwcaps/x86-64-v3", 0x7ffc6432a1c0, 0) = -1 ENOENT (No such file or directory)
```

```
openat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/glibc-
hwcaps/x86-64-v2/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No
such file or directory)
```

```
newfstatat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/glibc-
hwcaps/x86-64-v2", 0x7ffc6432a1c0, 0) = -1 ENOENT (No such file or directory)
```

```
openat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/tls/x86_64/x86_64/libstdc+
+.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
```

```
newfstatat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/tls/x86_64/x86_64",
0x7ffc6432a1c0, 0) = -1 ENOENT (No such file or directory)
```

```
openat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/tls/x86_64/libstdc++.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
```

```
newfstatat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/tls/x86_64", 0x7ffc6432a1c0, 0) =
-1 ENOENT (No such file or directory)
```

```
openat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/tls/x86_64/libstdc++.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
```

```
newfstatat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/tls/x86_64", 0x7ffc6432a1c0, 0) =
-1 ENOENT (No such file or directory)
```

```
openat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/tls/libstdc+
+.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
```

```
newfstatat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/tls",
0x7ffc6432a1c0, 0) = -1 ENOENT (No such file or directory)
```

```
openat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/x86_64/x86_64/libstdc++.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
```



```

newfstatat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/x86_64",
0x7ffc6432a1c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/x86_64/libstdc++.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/x86_64",
0x7ffc6432a1c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD,
"/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/x86_64/libstdc++.so.6",
O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/x86_64",
0x7ffc6432a1c0, 0) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/libstdc+
+.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

newfstatat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri",
{st_mode=S_IFDIR|0755, st_size=288, ...}, 0) = 0

openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=105007, ...},
AT_EMPTY_PATH) = 0

mmap(NULL, 105007, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7563329a5000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libstdc++.so.6", O_RDONLY|
O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) =
832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2260296, ...},
AT_EMPTY_PATH) = 0

mmap(NULL, 2275520, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3,
0) = 0x756332600000

mprotect(0x75633269a000, 1576960, PROT_NONE) = 0

mmap(0x75633269a000, 1118208, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9a000) =
0x75633269a000

```

mmap(0x7563327ab000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1ab000) = 0x7563327ab000

mmap(0x75633281b000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x21a000) = 0x75633281b000

mmap(0x756332829000, 10432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x756332829000

close(3) = 0

openat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=125488, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 127720, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x756332985000

mmap(0x756332988000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x756332988000

mmap(0x75633299f000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000) = 0x75633299f000

mmap(0x7563329a3000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x7563329a3000

close(3) = 0

openat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\1237\2\0\0\0\0"..., 832) = 832

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0"..., 784, 64) = 784

pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0"..., 48, 848) = 48

pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\17\357\204\3\$\fl221\2039x\324\224\323\236S"..., 68, 896) = 68

newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 0

pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784

mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x756332200000

mprotect(0x756332228000, 2023424, PROT_NONE) = 0

mmap(0x756332228000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x756332228000

mmap(0x7563323bd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7563323bd000

mmap(0x756332416000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x756332416000

mmap(0x75633241c000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x75633241c000

close(3) = 0

openat(AT_FDCWD, "/snap/alacritty/140/usr/lib/x86_64-linux-gnu/dri/libm.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x75633289e000

mmap(0x7563328ac000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7563328ac000

mmap(0x756332928000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x756332928000

mmap(0x756332983000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x756332983000

close(3) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75633289c000

arch_prctl(ARCH_SET_FS, 0x75633289d3c0) = 0

set_tid_address(0x75633289d690) = 61591

set_robust_list(0x75633289d6a0, 24) = 0

rseq(0x75633289dd60, 0x20, 0, 0x53053053) = 0

mprotect(0x756332416000, 16384, PROT_READ) = 0

mprotect(0x756332983000, 4096, PROT_READ) = 0

mprotect(0x7563329a3000, 4096, PROT_READ) = 0

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x75633289a000

mprotect(0x75633281b000, 45056, PROT_READ) = 0

mprotect(0x600a3f025000, 4096, PROT_READ) = 0

mprotect(0x7563329f9000, 8192, PROT_READ) = 0

prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0

munmap(0x7563329a5000, 105007) = 0

getrandom("\xf7\xdb\xf2\x43\x74\xff\xd8\xb2", 8, GRND_NONBLOCK) = 8

brk(NULL) = 0x600a3facc000

brk(0x600a3faed000) = 0x600a3faed000

write(1, "Unsorted array:\n", 16Unsorted array:

) = 16

write(1, "41 ", 341) = 3

write(1, "44 ", 344) = 3

```

write(1, "56 ", 356 )           = 3
write(1, "79 ", 379 )           = 3
write(1, "50 ", 350 )           = 3
write(1, "74 ", 374 )           = 3
write(1, "12 ", 312 )           = 3
write(1, "1 ", 21 )              = 2
write(1, "38 ", 338 )           = 3
write(1, "62 ", 362 )           = 3
write(1, "7 ", 27 )              = 2
write(1, "7 ", 27 )              = 2
write(1, "66 ", 366 )           = 3
write(1, "78 ", 378 )           = 3
write(1, "54 ", 354 )           = 3
write(1, "77 ", 377 )           = 3
write(1, "32 ", 332 )           = 3
write(1, "64 ", 364 )           = 3
write(1, "76 ", 376 )           = 3
write(1, "92 ", 392 )           = 3
write(1, "\n", 1
)                                = 1

```

```
gettimeofday({tv_sec=1734957902, tv_usec=196593}, NULL) = 0
```

```
rt_sigaction(SIGRT_1, {sa_handler=0x756332291870, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO,
sa_restorer=0x756332242520}, NULL, 8) = 0
```

```
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
```

```
mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|
MAP_STACK, -1, 0) = 0x756331800000
```

```
mprotect(0x756331801000, 8388608, PROT_READ|PROT_WRITE) = 0
```

rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

**clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|
CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x756332000910, parent_tid=0x756332000910, exit_signal=0,
stack=0x756331800000, stack_size=0x7fff00, tls=0x756332000640} =>
{parent_tid=[61592]}, 88) = 61592**

rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

**mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|
MAP_STACK, -1, 0) = 0x756330400000**

mprotect(0x756330401000, 8388608, PROT_READ|PROT_WRITE) = 0

rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

**clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|
CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|
CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x756330c00910, parent_tid=0x756330c00910, exit_signal=0,
stack=0x756330400000, stack_size=0x7fff00, tls=0x756330c00640} =>
{parent_tid=[61595]}, 88) = 61595**

rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

**futex(0x756332000910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,
61592, NULL, FUTEX_BITSET_MATCH_ANY) = 0**

futex(0x7563329fc108, FUTEX_WAIT_PRIVATE, 2, NULL) = 0

munmap(0x756329800000, 8392704) = 0

futex(0x7563329fc108, FUTEX_WAKE_PRIVATE, 1) = 1

**futex(0x756330c00910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,
61595, NULL, FUTEX_BITSET_MATCH_ANY) = 0**

munmap(0x756322c00000, 8392704) = 0

gettimeofday({tv_sec=1734957902, tv_usec=201236}, NULL) = 0

write(1, "Sorted array:\n", 14Sorted array:

) = 14

write(1, "1 ", 21) = 2

write(1, "7 ", 27) = 2

write(1, "7 ", 27) = 2

```

write(1, "12 ", 312 )           = 3
write(1, "32 ", 332 )           = 3
write(1, "38 ", 338 )           = 3
write(1, "41 ", 341 )           = 3
write(1, "44 ", 344 )           = 3
write(1, "50 ", 350 )           = 3
write(1, "54 ", 354 )           = 3
write(1, "56 ", 356 )           = 3
write(1, "62 ", 362 )           = 3
write(1, "64 ", 364 )           = 3
write(1, "66 ", 366 )           = 3
write(1, "74 ", 374 )           = 3
write(1, "76 ", 376 )           = 3
write(1, "77 ", 377 )           = 3
write(1, "78 ", 378 )           = 3
write(1, "79 ", 379 )           = 3
write(1, "92 ", 392 )           = 3
write(1, "\n", 1
)                                = 1

```

```

write(1, "Time elapsed: 0.004643 seconds\n", 31)
Time elapsed: 0.004643
seconds
) = 31

```

```

exit_group(0)                   = ?

```

```

+++ exited with 0 +++

```

Вывод:

Работа над заданием позволила углубиться в особенности многопоточного программирования, что является важным и востребованным навыком. Основные сложности возникли на этапе правильного распределения задач между потоками и

синхронизации доступа к данным. Однако полученные результаты показывают, что даже при наличии накладных расходов многопоточность позволяет достичь ускорения. С увеличением количества потоков эффективность снижается, что указывает на предел масштабируемости алгоритма в данной реализации.

Число потоков	Время исполнения (с)	Ускорение	Эффективность
1	0.006140	1	1
2	0.004554	1.35	0.67
4	0.004262	1.44	0.36
8	0.004700	1.31	0.16
16	0.004314	1.42	0.09