

**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»**

Выполнил: Д. А. Кузнецов  
Группа: М8О-207БВ-24  
Преподаватель: Е. С. Миронов

Москва, 2025

## **Условие**

### **Цель работы:**

Приобретение практических навыков в:

- Управление потоками в ОС
- Обеспечение синхронизации между потоками

### **Задание:**

Составить программу на языке С++, 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.

### **Вариант: 3**

Отсортировать массив целых чисел при помощи параллельной сортировки слиянием

## **Метод решения**

Алгоритм решения задачи:

1. Пользователь запускает программу с параметром –max-threads N.
2. Создается объект ParallelMergeSort с максимальным числом потоков N.
3. Генерируется случайный массив заданным размером элементов.
4. Вызывается метод Sort() основного массива.
5. Инициализируется счетчик активных потоков.
6. Запускается рекурсивная параллельная сортировка.
7. Попытка создания нового потока. В случае неудачи - сортируем подмассив последовательно.
8. Для маленьких подмассивов или при недоступности потоков рекурсивно делим массив пополам, сортируем левую и правую части, выполняем слияние остоортированных частей.
9. Когда все рекурсивные вызовы завершены: основной поток завершает выполнение, выводится время сортировки, программа завершает работу.

Архитектура программы:

```
lab2/
  build/
  include/
    sort.h
    utils.h
  src/
    sort.cpp
    utils.cpp
  CMakeLists.txt
  main.cpp
```

Ссылки:

- <https://en.cppreference.com/w/cpp/chrono.html>
- <https://en.cppreference.com/w/cpp/thread/thread.html>
- [https://en.cppreference.com/w/cpp/thread/thread/lock\\_guard.html](https://en.cppreference.com/w/cpp/thread/thread/lock_guard.html)
- <https://en.cppreference.com/w/cpp/numeric/random/rand.html>
- [https://en.cppreference.com/w/cpp/numeric/random/mersenne\\_twister\\_engine.html](https://en.cppreference.com/w/cpp/numeric/random/mersenne_twister_engine.html)
- <https://en.cppreference.com/w/cpp/algorithm.html>
- <https://en.cppreference.com/w/cpp/header/cstdlib.html>

## Описание программы

main.cpp — точка входа в программу.

include/sort.h — объявление класса ParallelMergeSort

src/sort.cpp — реализация параллельной сортировки

- ParallelMergeSort::ParallelMergeSort(int maxThreads); — конструктор.
- void ParallelMergeSort::ParallelSort(std::vector<int> arr, int left, int right); — рекурсивная параллельная сортировка.
- void ParallelMergeSort::SequentialSort(std::vector<int> arr, int left, int right); — последовательная сортировка.
- void ParallelMergeSort::Merge(std::vector<int> arr, int left, int mid, int right); — слияние отсортированных массивов.

include/utils.h — объявление вспомогательной функции.

src/utils.cpp — реализация вспомогательной функции.

- `std::vector<int> GenerateRandomArray(size_t size);` — генерация случайного массива.

## Результаты

Программа получает на вход один параметр: максимальное число потоков(`-max-threads N`) После проверки корректности аргументов она запускает параллельную сортировку слиянием массива из 10000000 элементов. Алгоритм динамически распределяет работу между потоками, создавая новые потоки только когда это эффективно и не превышает заданный лимит.

По завершении всех вычислений программа выводит в стандартный поток вывода:

- размер отсортированного массива,
- максимальное число используемых потоков,
- время выполнения в миллисекундах,
- статус проверки корректности сортировки.

Результатом работы является отсортированный по возрастанию массив, корректность которого проверяется автоматически. Алгоритм гарантирует правильную сортировку благодаря корректной реализации слияния отсортированных подмассивов и атомарному управлению потоками.

Программа корректно обрабатывает ошибочные ситуации (некорректные аргументы, недопустимые значения потоков) и завершается с соответствующим кодом ошибки. В случае корректного запуска все потоки завершаются штатно, ресурсы освобождаются автоматически, а результат сортировки проходит валидацию.

График зависимости времени выполнения от количества используемых потоков приведен на рисунке 1 (размер массива = 10000000 элементов).

Анализ показывает, что при увеличении числа потоков от 1 до 8 наблюдается значительное сокращение времени выполнения, что свидетельствует о хорошей параллелизумости алгоритма сортировки слиянием. Ускорение достигает примерно 3.5× при использовании 8 потоков по сравнению с однопоточной версией.

Оптимальная производительность достигается при 4-8 потоках, после чего дальнейшее увеличение числа потоков приводит к уменьшению прироста производительности. Это объясняется накладными расходами на создание и синхронизацию потоков, а также ограничениями памяти и кэша процессора.

При использовании максимального числа потоков (12 и более) наблюдается насыщение производительности, где дополнительное распараллеливание не дает существенного выигрыша из-за возрастающих затрат на коммуникацию между потоками и конкуренции за ресурсы памяти.

Таким образом, для задачи параллельной сортировки слиянием оптимальное число потоков составляет 4-8 для массива из 10000000 элементов. Это связано с балансом между вычислительной сложностью подзадач и накладными расходами на управление потоками. Алгоритм эффективно использует доступные вычислительные ресурсы и демонстрирует хорошую масштабируемость в многопоточной среде.



Рис. 1: График зависимости времени выполнения от количества используемых потоков.

## Выводы

В ходе выполнения лабораторной работы были приобретены практические навыки в организации многопоточной сортировки данных, синхронизации потоков и эффективного управления вычислительными ресурсами.

Была разработана и отлажена программа на языке C++, реализующая параллельную сортировку слиянием для больших массивов данных. Программа использует стандартные средства многопоточности C++ (`std::thread`, `std::atomic`), что обеспечивает корректную работу на различных операционных системах и поддерживает кроссплатформенность.

В результате работы программа динамически распределяет вычислительную нагрузку между потоками, создавая новые потоки только когда это эффективно и не превышает заданный пользователем лимит. Алгоритм реализует стратегию "разделяй и властвуй" рекурсивно разделяя массив на подмассивы и параллельно обрабатывая их в отдельных потоках.

Была обеспечена безопасная работа с разделяемыми ресурсами: для контроля количества активных потоков используется атомарная переменная (`std::atomic`), что исключает гонки данных и гарантирует корректность работы в многопоточной среде.

Были обработаны возможные ошибки ввода (некорректные аргументы командной строки, недопустимые значения потоков), а также обеспечена корректная инициализация и завершение всех потоков. Программа включает проверку корректности сортировки для верификации результатов.

## Исходная программа

```
1 #pragma once
2
3 #include <vector>
4 #include <atomic>
5
6 class ParallelMergeSort {
7 public:
8     explicit ParallelMergeSort(int maxThreads);
9     void Sort(std::vector<int>& arr);
10
11 private:
12     void Merge(std::vector<int>& arr, int left, int mid, int right);
13     void SequentialSort(std::vector<int>& arr, int left, int right);
14     void ParallelSort(std::vector<int>& arr, int left, int right);
15
16     int maxThreads_;
17     std::atomic<int> activeThreads_{1};
18 };
```

Листинг 1: sort.h

```
1 #include "sort.h"
2 #include <thread>
3 #include <vector>
4
5 ParallelMergeSort::ParallelMergeSort(int maxThreads) : maxThreads_(maxThreads) {
6     if (maxThreads_ < 1) maxThreads_ = 1;
7 }
8
9 void ParallelMergeSort::Merge(std::vector<int>& arr, int left, int mid, int right) {
10    std::vector<int> leftArr(arr.begin() + left, arr.begin() + mid + 1);
11    std::vector<int> rightArr(arr.begin() + mid + 1, arr.begin() + right + 1);
12
13    int i = 0;
14    int j = 0;
15    int k = left;
16
17    while (i < static_cast<int>(leftArr.size()) && j < static_cast<int>(rightArr.size())
18          ()) {
19        if (leftArr[i] <= rightArr[j]) {
20            arr[k++] = leftArr[i++];
21        } else {
22            arr[k++] = rightArr[j++];
23        }
24    }
25
26    while (i < static_cast<int>(leftArr.size())) {
27        arr[k++] = leftArr[i++];
28    }
29
30    while (j < static_cast<int>(rightArr.size())) {
31        arr[k++] = rightArr[j++];
32    }
33
34 void ParallelMergeSort::SequentialSort(std::vector<int>& arr, int left, int right) {
```

```

35     if (left >= right) {
36         return;
37     }
38     int mid = left + (right - left) / 2;
39     SequentialSort(arr, left, mid);
40     SequentialSort(arr, mid + 1, right);
41     Merge(arr, left, mid, right);
42 }
43
44 void ParallelMergeSort::ParallelSort(std::vector<int>& arr, int left, int right) {
45     if (left >= right) return;
46
47     int mid = left + (right - left) / 2;
48
49     int current = activeThreads_.load();
50     bool spawned = false;
51     while (current < maxThreads_) {
52         if (activeThreads_.compare_exchange_weak(current, current + 1)) {
53             std::thread t([this, &arr, left, mid]() {
54                 ParallelSort(arr, left, mid);
55                 activeThreads_.fetch_sub(1);
56             });
57             ParallelSort(arr, mid + 1, right);
58             t.join();
59             spawned = true;
60             break;
61         }
62     }
63
64     if (spawned) {
65         Merge(arr, left, mid, right);
66     } else {
67         SequentialSort(arr, left, right);
68     }
69 }
70
71 void ParallelMergeSort::Sort(std::vector<int>& arr) {
72     if (arr.size() <= 1) {
73         return;
74     }
75     activeThreads_.store(1);
76     ParallelSort(arr, 0, static_cast<int>(arr.size()) - 1);
77 }

```

Листинг 2: sort.cpp

```

1 #pragma once
2
3 #include <vector>
4 #include <string>
5
6 class Utils {
7 public:
8     static std::vector<int> GenerateRandomArray(size_t size);
9 };

```

Листинг 3: utils.h

```

1 #include <utils.h>
2 #include <random>
3
4 std::vector<int> Utils::GenerateRandomArray(size_t size) {
5     std::vector<int> arr(size);
6     std::random_device rd;
7     std::mt19937 gen(rd());
8     std::uniform_int_distribution<> dis(1, 1000000);
9     for (auto& x : arr) x = dis(gen);
10    return arr;
11 }

```

Листинг 4: utils.cpp

```

1 #include <iostream>
2 #include <vector>
3 #include <chrono>
4 #include <string>
5 #include <cstdlib>
6 #include "sort.h"
7 #include "utils.h"
8
9 void PrintUsage(const char* prog) {
10     std::cerr << "Usage: " << prog << " --max-threads N\n";
11     std::cerr << " N must be a positive integer (e.g., 1, 2, 4, 8).\n";
12 }
13
14 int main(int argc, char* argv[]) {
15     if (argc != 3) {
16         PrintUsage(argv[0]);
17         return 1;
18     }
19
20     int maxThreads = -1;
21
22     if (std::string(argv[1]) == "--max-threads") {
23         maxThreads = std::atoi(argv[2]);
24     } else {
25         std::cerr << "Error: expected --max-threads as first argument.\n";
26         PrintUsage(argv[0]);
27         return 1;
28     }
29
30     if (maxThreads <= 0) {
31         std::cerr << "Error: --max-threads must be a positive integer.\n";
32         return 1;
33     }
34
35     const size_t ARRAY_SIZE = 1000000;
36     std::cout << "Generating random array of " << ARRAY_SIZE << " elements...\n";
37     std::vector<int> arr = Utils::GenerateRandomArray(ARRAY_SIZE);
38
39     std::cout << "Max threads allowed: " << maxThreads << "\n";
40     auto start = std::chrono::high_resolution_clock::now();
41     ParallelMergeSort sorter(maxThreads);
42     sorter.Sort(arr);
43     auto end = std::chrono::high_resolution_clock::now();
44

```

```

45     auto duration = std::chrono::duration_cast<std::chrono::milliseconds>(end - start)
46     ;
47     std::cout << "Sorting completed in " << duration.count() << " ms\n";
48 }
49 }
```

Листинг 5: main.cpp

## Strace

```

execve("./main", ["../main"], 0x7ffd0636d670 /* 37 vars */) = 0
brk(NULL)                               = 0x5c242a69c000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff336d94d0) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x72ea2fb48000
access("/etc/ld.so.preload", R_OK)        = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/glibc-hwcaps/x86_64", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/glibc-hwcaps/x86_64", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/glibc-hwcaps/x86_64", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/glibc-hwcaps/x86_64", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64/libc", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64/libc", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64/libc.so", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64/libc.so", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64/libc++.so", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64/libc++.so", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64/libstdc++.so", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/tls/haswell/x86_64/libstdc++.so", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/haswell/x86_64/libc", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/haswell/x86_64/libc", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/haswell/x86_64/libc.so", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/haswell/x86_64/libc.so", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/haswell/x86_64/libstdc++.so", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/haswell/x86_64/libstdc++.so", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/x86_64/libstdc++.so", O_RDONLY) = -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD, "/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/x86_64/libstdc++.so", {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}, {st_size=0, st_mode=S_IFREG|0644, st_nlink=1}) = -1 ENOENT (No such file or directory)
```

```
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/x86_64",0x7
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib/libstdc++.so.6"
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_ros_com/lib",{st_mode=S_
= 0
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/glibc-hwcaps/x86-6
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/glibc-hwcaps/x
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/glibc-hwcaps/x86-6
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/glibc-hwcaps/x
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/tls/haswell/x86_64
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/tls/haswell/x8
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/tls/haswell/libstd
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/tls/haswell",0
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/tls/x86_64/libstdc
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/tls/x86_64",0x
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/tls/libstdc++.so.6
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/tls",0x7fff336
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/haswell/x86_64/lib
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/haswell/x86_64
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/haswell/libstdc++.
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/haswell",0x7ff
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/x86_64/libstdc++.
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/x86_64",0x7ff
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib/libstdc++.so.6",0_
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/clydegriffiths/ros2_ws/install/px4_msgs/lib",{st_mode=S_IF_
= 0
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/glibc-hwcaps/x86-64-v3/libs
```

```
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/glibc-hwcaps/x86-64-v3"
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/glibc-hwcaps/x86-64-v2/libst
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/glibc-hwcaps/x86-64-v2"
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/tls/haswell/x86_64/libstdc+
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/tls/haswell/x86_64",0x7
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/tls/haswell/libstdc++.so.6"
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/tls/haswell",0x7fff336d
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/tls/x86_64/libstdc++.so.6",
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/tls/x86_64",0x7fff336d8
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/tls/libstdc++.so.6",0_RDONLY
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/tls",0x7fff336d86f0,0)
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/haswell/x86_64/libstdc++.so
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/haswell/x86_64",0x7fff3
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/haswell/libstdc++.so.6",0_R
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/haswell",0x7fff336d86f0
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/x86_64/libstdc++.so.6",0_RDONLY
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/x86_64",0x7fff336d86f0
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib/libstdc++.so.6",0_RDONLY|0_O
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/opt/rviz_ogre_vendor/lib",{st_mode=S_IFDIR|0755,
= 0
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v3/libstdc+
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v3",0x7
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v2/libstdc+
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/glibc-hwcaps/x86-64-v2",0x7
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/tls/haswell/x86_64/libstdc++.so
```

```
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/tls/haswell/x86_64",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/tls/haswell/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/tls/haswell",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/tls/x86_64/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/tls/x86_64",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/tls/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/tls",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/haswell/x86_64/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/haswell/x86_64",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/haswell/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/haswell",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/x86_64/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/x86_64",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/x86_64-linux-gnu",{st_mode=S_IFDIR|0755,st_size=0
openat(AT_FDCWD,"/opt/ros/humble/lib/glibc-hwcaps/x86-64-v3/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/glibc-hwcaps/x86-64-v3",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/glibc-hwcaps/x86-64-v2/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/glibc-hwcaps/x86-64-v2",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/tls/haswell/x86_64/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/tls/haswell/x86_64",0x7fff336d86f0,0,_R
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/tls/haswell/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/opt/ros/humble/lib/tls/haswell",0x7fff336d86f0,0) = -1
ENOENT (No such file or directory)
openat(AT_FDCWD,"/opt/ros/humble/lib/tls/x86_64/libstdc++.so.6",0_RDONLY|0_CLOEXEC)
```







```
futex(0x72ea2fa2977c,FUTEX_WAKE_PRIVATE,2147483647) = 0
write(2,"Usage: ",7Usage: ) = 7
write(2,"./main",6./main) = 6
write(2," --max-threads N\n",17 --max-threads N
) = 17
write(2," N must be a positive integer (...),51 N must be a positive integer
(e.g.,1,2,4,8).
) = 51
exit_group(1) = ?
+++ exited with 1 +++
```