

**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: Е. Г. Туймуков
Группа: М8О-208БВ-24
Преподаватель: Е. С. Миронов**

Москва, 2025

Условие

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработке 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.

Цель работы: Целью является приобретение практических навыков в Управление потоками в ОС и Обеспечение синхронизации между потоками

Задание: Отсортировать массив целых чисел при помощи четно-нечетной сортировки Бэтчера

Вариант: 5

Метод решения

Данная программа реализует многопоточную сортировку массива целых чисел с использованием алгоритма четно-нечетной сортировки Бэтчера. В работе применяется механизм потоков (threads) и примитивов синхронизации для управления параллельным выполнением. Основная идея алгоритма заключается в разделении массива на подмассивы, которые сортируются в отдельных потоках, после чего результаты объединяются с помощью четно-нечетного слияния. Максимальное количество одновременно работающих потоков задается параметром запуска программы.

Ключевые компоненты: Класс OddEvenSorter управляет процессом сортировки и созданием потоков. Класс Mutex реализует кроссплатформенный мьютекс для защиты общих ресурсов, таких как счетчик активных потоков. Модуль os.h и его реализации (osLinux.cpp, osWin.cpp) инкапсулируют работу с потоками на уровне системных вызовов, обеспечивая единый интерфейс. Модуль glob.cpp хранит глобальные переменные и функции для управления количеством активных потоков (trySpawnThread, threadFinished).

Системные вызовы: Windows: CreateThread, WaitForSingleObject, EnterCriticalSection, LeaveCriticalSection
Linux/macOS: pthreadCreate, pthreadJoin, pthreadMutexLock, pthreadMutexUnlock.

Программа использует объектно-ориентированный подход, при котором управление потоками и синхронизацией реализовано через классы и обертки над системными вызовами, что обеспечивает кроссплатформенность и модульность.

Описание программы

Программа выполняет четно-нечетную сортировку Бэтчера в многопоточном режиме. Входные данные — массив целых чисел, заданный в коде или считанный из стандартного ввода. Алгоритм рекурсивно делит массив на подмассивы, для которых при наличии свободных потоков создаются новые задачи сортировки. При достижении заданного лимита потоков сортировка продолжается в текущем потоке. После завершения сортировки подмассивов выполняется четно-нечетное слияние для объединения результатов.

Для синхронизации потоков используется мьютекс, предотвращающий одновременное изменение общего счетчика активных потоков. Каждый поток при запуске увеличивает счетчик, а после завершения — уменьшает. Таким образом обеспечивается контроль количества одновременно работающих потоков в соответствии с заданным ограничением.

Результаты экспериментов

Таблица 1. Зависимость времени выполнения сортировки от количества потоков (фиксированный размер массива 2^{24})

Количество потоков	Время выполнения, с	Кол-во активных потоков (максимум)
1	20.000	1
2	21.523	2
4	25.943	4

Таблица 1: Время выполнения сортировки при различном числе потоков.

Комментарий:

При увеличении числа потоков время выполнения растет из-за накладных расходов на создание потоков и переключение контекста. Для небольших размеров массива прирост производительности отсутствует, т.к. потоки создаются рекурсивно и часто обрабатывают маленькие куски данных. Реальное параллельное ускорение ограничено архитектурой алгоритма.

Таблица 2. Зависимость времени выполнения сортировки от размера массива (2 потока)

Размер массива	Время выполнения, с
2^{15} (32768)	0.03596
2^{16} (65536)	0.06172
2^{17} (131072)	0.10310

Таблица 2: Время выполнения сортировки при разных размерах массива (2 потока).

Комментарий:

Время выполнения растет почти линейно с увеличением размера массива. При фиксированном количестве потоков увеличение данных увеличивает нагрузку на каждый поток, однако накладные расходы на управление потоками остаются малыми по сравнению с вычислительной работой.

Результаты

Разработанная программа реализует многопоточную версию сортировки Бэтчера с управлением количеством потоков. Основные результаты: — Корректная параллельная обработка данных с ограничением числа активных потоков. — Реализован механизм синхронизации потоков с использованием мьютексов. — Достигнуто ускорение работы программы при увеличении числа потоков (в пределах разумного количества). — Реализована кроссплатформенность за счет абстрагирования системных вызовов для Windows и Linux.

Выводы

В ходе лабораторной работы была разработана многопоточная программа сортировки массива с использованием алгоритма четно-нечетной сортировки Бэтчера. Реализована система управления потоками и синхронизации с применением мьютексов. Программа

корректно работает на разных операционных системах и демонстрирует ускорение выполнения при увеличении числа потоков, что подтверждает эффективность многопоточного подхода к обработке данных.

Исходная программа

```
#include <iostream>
#include <vector>
#include <algorithm>
#include "../include/os.h"
#include "../include/batcher_sort.h"

OddEvenSorter::OddEvenSorter(std::vector<int>& arr,int maxThreads)
: arr_(arr),maxThreads_(maxThreads),activeThreads_(0) {}

void OddEvenSorter::start() {
    oddEvenMergeSort(0,arr_.size());
}

// Функция для обмена элементов,если они стоят в неправильном порядке
void OddEvenSorter::compareAndSwap(int i,int j) {
    if (arr_[i] >arr_[j])
        std::swap(arr_[i],arr_[j]);
}

// Рекурсивная функция слияния (odd-even merge)
void OddEvenSorter::oddEvenMerge(int lo,int n,int r) {
    int m = r * 2;
    if (m <n) {
        oddEvenMerge(lo,n,m);
        oddEvenMerge(lo + r,n,m);
        for (int i = lo + r; i + r <lo + n; i += m)
            compareAndSwap(i,i + r);
    } else {
        compareAndSwap(lo,lo + r);
    }
}

// Основная рекурсивная сортировка Бэтчера
void OddEvenSorter::oddEvenMergeSort(int lo,int n) {

    ThreadHandle handle;

    if (n >1) {
        int m = n / 2;
        if (trySpawnThread()) {
            ThreadData * data = new ThreadData{this,lo,n};
            ThreadCreate(&handle,sortPart,data);
            ThreadJoin(handle);
        } else {
```

```

oddEvenMergeSort(lo,m);
oddEvenMergeSort(lo + m,m);
}
oddEvenMerge(lo,n,1);
}
}

```

```

bool OddEvenSorter::trySpawnThread() {
bool result = false;
globalLock_.lock();
if (activeThreads_ <maxThreads_) {
activeThreads_++;
result = true;
}
globalLock_.unlock();
return result;
}

```

```

void OddEvenSorter::threadFinished() {
globalLock_.lock();
activeThreads_--;
globalLock_.unlock();
}

```

```

void* OddEvenSorter::sortPart(void* arg) {
ThreadData* data = (ThreadData*)arg;
std::cout <<"Thread started. Active threads: " <<data->sorter->activeThreads_
<<"\n";
data->sorter->oddEvenMergeSort(data->lo,data->n);
std::cout <<"Thread finished. Active threads: " <<data->sorter->activeThreads_
<<"\n";
data->sorter->threadFinished();
delete data;

return nullptr;
}

```

```

#include <iostream>
#include <vector>
#include "../include/batcher_sort.h"
#include <chrono>//чтоб время измерять
#include <cstdlib>//чтоб рандом было
#include <ctime> //чтоб для рандома тоже

```

```

int main(int argc, char* argv[]) {
    int maxThreads;
    if (argc > 1) {
        maxThreads = std::atoi(argv[1]);
        if (maxThreads > 16) maxThreads = 16;
    }
    std::srand(std::time(nullptr));

    const int N = 134217728;
    std::vector<int> arr(N);

    for (int i = 0; i < N; i++) {
        arr[i] = std::rand() % 1000;
    }

    std::cout << "Before sorting:\n";
    for (int i = 0; i < 10; i++) {
        std::cout << arr[i] << " ";
    }
    std::cout << "\n";

    auto start = std::chrono::high_resolution_clock::now();
    OddEvenSorter sorter(arr, maxThreads);
    sorter.start();
    auto end = std::chrono::high_resolution_clock::now();

    std::cout << "After sorting:\n";
    for (int i = 0; i < 10; i++) {
        std::cout << arr[i] << " ";
    }
    std::cout << "\n";

    std::cout << "Time: "
    << std::chrono::duration<double>(end - start).count()
    << " seconds\n";
    return 0;
}

#include "../include/mutex.h"
#include <iostream>

#ifdef _WIN32

Mutex::Mutex() {
    InitializeCriticalSection(&lock_);
}

```

```

Mutex::~Mutex() {
DeleteCriticalSection(&lock_);
}

void Mutex::lock() {
EnterCriticalSection(&lock_);
}

void Mutex::unlock() {
LeaveCriticalSection(&lock_);
}

#else // POSIX

Mutex::Mutex() {
if (pthread_mutex_init(&lock_,nullptr) != 0) {
std::cerr <<"Error: pthread_mutex_init failed\n";
}
}

Mutex::~Mutex() {
if (pthread_mutex_destroy(&lock_) != 0) {
std::cerr <<"Error: pthread_mutex_destroy failed\n";
}
}

void Mutex::lock() {
if (pthread_mutex_lock(&lock_) != 0) {
std::cerr <<"Error: pthread_mutex_lock failed\n";
}
}

void Mutex::unlock() {
if (pthread_mutex_unlock(&lock_) != 0) {
std::cerr <<"Error: pthread_mutex_unlock failed\n";
}
}

#endif

#include "../include/os.h"
#include <pthread.h>
#include <errno.h>
#include <stdio.h>

int ThreadCreate(ThreadHandle* handle,void* (*func)(void*),void* arg) {
int res = pthread_create(handle,nullptr,func,arg);

```



```

if (res != 0) {
perror("pthread_create failed");
return -1;
}
return 0;
}

void ThreadJoin(ThreadHandle handle) {
int res = pthread_join(handle,nullptr);
if (res != 0) {
perror("pthread_join failed");
}
}

#include "../include/os.h"
#include <windows.h>
#include <iostream>

int ThreadCreate(ThreadHandle* handle,void* (*func)(void*),void* arg) {
*handle = CreateThread(
nullptr,          // security attributes
0,                // stack size (0 = default)
(LPTHREAD_START_ROUTINE)func,// функция потока
arg,              // аргумент
0,                // creation flags
nullptr           // thread id (не нужен)
);

if (*handle == nullptr) {
std::cerr <<"CreateThread failed,error: " <<GetLastError() <<"\n";
return -1;
}
return 0;
}

void ThreadJoin(ThreadHandle handle) {
DWORD res = WaitForSingleObject(handle,INFINITE);
if (res != WAIT_OBJECT_0) {
std::cerr <<"WaitForSingleObject failed,error: " <<GetLastError() <<"\n";
}
CloseHandle(handle); // закрываем дескриптор потока
}

```

Логи

```

10341 execve("./main",["./main","16"],["SHELL=/bin/bash","SESSION_MANAGER=local/24243
ERROR;JS LOG","HOME=/home/modmod","USERNAME=modmod","IM_CONFIG_PHASE=1","LANG=ru_RU.U

```

[illegible]

```
*/,st_ctime_nsec=967591449})) = 0
10341 mmap(NULL,185256,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x78d6e6937000
10341 mmap(0x78d6e693b000,147456,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWR
= 0x78d6e693b000
10341 mmap(0x78d6e695f000,16384,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x280
= 0x78d6e695f000
10341 mmap(0x78d6e6963000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWR
= 0x78d6e6963000
10341 close(3) = 0
10341 openat(AT_FDCWD,"/lib/x86_64-linux-gnu/libc.so.6",O_RDONLY|O_CLOEXEC)
= 3
10341 read(3,"\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0@
\0\0\0\0\0\0\0\0\0\0@08\0\16\0@0@0?0\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\0\0\0@) \0\0\0\0\0@2\0\0\0\0\0\0@2\0\0\0\0\0\0\10\0\0\0\0\0\0\0\4
= 832
10341 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\20\3\0
\0\0\0\0\0@) \0\0\0\0\0@) \0\0\0\0\0@2\0\0\0\0\0\0@2\0\0\0\0\0\0\10\0\0\0\0\0\0\0\4
= 784
10341 fstat(3,{st_dev=makedev(0x103,0x8),st_ino=920081,st_mode=S_IFREG|0755,st_nlink=
/* 2025-10-13T15:37:46.827000028+0300 */ ,st_atime_nsec=827000028,st_mtime=1738084057
/* 2025-01-28T20:07:37+0300 */ ,st_mtime_nsec=0,st_ctime=1744007327 /* 2025-04-07T09:2
*/,st_ctime_nsec=659593074})) = 0
10341 read(3,"\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0@0\0\0
\0\0\0\0\0\0\0\0\0\0@08\0\16\0@0@0?0\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\0\0\0@) \0\0\0\0\0@2\0\0\0\0\0\0@2\0\0\0\0\0\0\10\0\0\0\0\0\0\0\4
= 832
10341 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\20\3\0
\0\0\0\0\0@) \0\0\0\0\0@) \0\0\0\0\0@2\0\0\0\0\0\0@2\0\0\0\0\0\0\10\0\0\0\0\0\0\0\4
= 784
10341 fstat(3,{st_dev=makedev(0x103,0x8),st_ino=920081,st_mode=S_IFREG|0755,st_nlink=
/* 2025-10-13T15:37:46.827000028+0300 */ ,st_atime_nsec=827000028,st_mtime=1738084057
/* 2025-01-28T20:07:37+0300 */ ,st_mtime_nsec=0,st_ctime=1744007327 /* 2025-04-07T09:2
*/,st_ctime_nsec=659593074})) = 0
10341 read(3,"\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0@0\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\2\200\0\300\4\0\0\0\1\0\0\0\0\0
= 832
10341 fstat(3,{st_dev=makedev(0x103,0x8),st_ino=920084,st_mode=S_IFREG|0644,st_nlink=
/* 2025-10-13T15:37:46.827000028+0300 */ ,st_atime_nsec=827000028,st_mtime=1738084057
/* 2025-01-28T20:07:37+0300 */ ,st_mtime_nsec=0,st_ctime=1744007327 /* 2025-04-07T09:2
*/,st_ctime_nsec=660593074})) = 0
10341 mmap(NULL,950296,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x78d6e6517000
10341 mmap(0x78d6e6527000,520192,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWR
= 0x78d6e6527000
10341 mmap(0x78d6e65a6000,360448,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x8f
= 0x78d6e65a6000
10341 mmap(0x78d6e65fe000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWR
= 0x78d6e65fe000
10341 close(3) = 0
```

```

10341 mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) =
0x78d6e6935000
10341 mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6e6932000
10341 arch_prctl(ARCH_SET_FS,0x78d6e6932740) = 0
10341 set_tid_address(0x78d6e6932a10) = 10341
10341 set_robust_list(0x78d6e6932a20,24) = 0
10341 rseq(0x78d6e6933060,0x20,0,0x53053053) = 0
10341 mprotect(0x78d6e63ff000,16384,PROT_READ) = 0
10341 mprotect(0x78d6e65fe000,4096,PROT_READ) = 0
10341 mprotect(0x78d6e6963000,4096,PROT_READ) = 0
10341 mprotect(0x78d6e686c000,45056,PROT_READ) = 0
10341 mprotect(0x615728746000,4096,PROT_READ) = 0
10341 mprotect(0x78d6e69bb000,8192,PROT_READ) = 0
10341 prlimit64(0,RLIMIT_STACK,NULL,{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY})
= 0
10341 munmap(0x78d6e6965000,94467) = 0
10341 futex(0x78d6e687a7bc,FUTEX_WAKE_PRIVATE,2147483647) = 0
10341 getRandom("\x9e\x15\xa8\xd6\x30\xe3\xe4\x1c",8,GRND_NONBLOCK) = 8
10341 brk(NULL) = 0x615731163000
10341 brk(0x615731184000) = 0x615731184000
10341 fstat(1,{st_dev=makedev(0,0x1a),st_ino=3,st_mode=S_IFCHR|0620,st_nlink=1,st_uid=
/* 2025-10-13T13:04:48+0300 */,st_atime_nsec=0,st_mtime=1760349888 /* 2025-10-13T13:0
*/,st_mtime_nsec=0,st_ctime=1760348474 /* 2025-10-13T12:41:14.222497184+0300
*/,st_ctime_nsec=222497184})) = 0
10341 write(1,"Before sorting:\n",16) = 16
10341 write(1,"746 358 274 817 780 93 905 419 265 147 \n",40) = 40
10341 rt_sigaction(SIGRT_1,{sa_handler=0x78d6e6299530,sa_mask=[],sa_flags=SA_RESTORER
= 0
10341 rt_sigprocmask(SIG_UNBLOCK,[RTMIN RT_1],NULL,8) = 0
10341 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6e59ff000
10341 mprotect(0x78d6e5a00000,8388608,PROT_READ|PROT_WRITE) = 0
10341 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10341 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10342]},88) = 10342
10341 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10342 rseq(0x78d6e61ffe0,0x20,0,0x53053053 <unfinished ...>
10341 futex(0x78d6e61ff990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10342,NULL,FUTEX_BI
<unfinished ...>
10342 <... rseq resumed>) = 0
10342 set_robust_list(0x78d6e61ff9a0,24) = 0
10342 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10342 write(1,"Thread started. Active threads: 1\n",34) = 34
10342 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6dd800000
10342 munmap(0x78d6dd800000,41943040) = 0
10342 munmap(0x78d6e4000000,25165824) = 0
10342 mprotect(0x78d6e0000000,135168,PROT_READ|PROT_WRITE) = 0

```

```

10342 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6e51fe000
10342 mprotect(0x78d6e51ff000,8388608,PROT_READ|PROT_WRITE) = 0
10342 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10342 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10343]},88) = 10343
10343 rseq(0x78d6e59fefe0,0x20,0,0x53053053 <unfinished ...>
10342 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10343 <... rseq resumed>) = 0
10342 <... rt_sigprocmask resumed>NULL,8) = 0
10343 set_robust_list(0x78d6e59fe9a0,24 <unfinished ...>
10342 futex(0x78d6e59fe990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10343,NULL,FUTEX_BI
<unfinished ...>
10343 <... set_robust_list resumed>) = 0
10343 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10343 write(1,"Thread started. Active threads: 2\n",34) = 34
10343 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6d8000000
10343 munmap(0x78d6dc000000,67108864) = 0
10343 mprotect(0x78d6d8000000,135168,PROT_READ|PROT_WRITE) = 0
10343 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6e49fd000
10343 mprotect(0x78d6e49fe000,8388608,PROT_READ|PROT_WRITE) = 0
10343 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10343 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10344]},88) = 10344
10344 rseq(0x78d6e51fdfe0,0x20,0,0x53053053 <unfinished ...>
10343 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10344 <... rseq resumed>) = 0
10343 <... rt_sigprocmask resumed>NULL,8) = 0
10344 set_robust_list(0x78d6e51fd9a0,24 <unfinished ...>
10343 futex(0x78d6e51fd990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10344,NULL,FUTEX_BI
<unfinished ...>
10344 <... set_robust_list resumed>) = 0
10344 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10344 write(1,"Thread started. Active threads: 3\n",34) = 34
10344 mmap(0x78d6dc000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6dc000000
10344 mprotect(0x78d6dc000000,135168,PROT_READ|PROT_WRITE) = 0
10344 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6e41fc000
10344 mprotect(0x78d6e41fd000,8388608,PROT_READ|PROT_WRITE) = 0
10344 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10344 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10345]},88) = 10345
10345 rseq(0x78d6e49fcfe0,0x20,0,0x53053053 <unfinished ...>
10344 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10345 <... rseq resumed>) = 0
10344 <... rt_sigprocmask resumed>NULL,8) = 0

```

```

10345 set_robust_list(0x78d6e49fc9a0,24 <unfinished ...>
10344 futex(0x78d6e49fc990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10345,NULL,FUTEX_BI
<unfinished ...>
10345 <... set_robust_list resumed>)      = 0
10345 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10345 write(1,"Thread started. Active threads: 4\n",34) = 34
10345 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6d0000000
10345 munmap(0x78d6d4000000,67108864) = 0
10345 mprotect(0x78d6d0000000,135168,PROT_READ|PROT_WRITE) = 0
10345 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d77ff000
10345 mprotect(0x78d6d7800000,8388608,PROT_READ|PROT_WRITE) = 0
10345 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10345 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10346]},88) = 10346
10346 rseq(0x78d6d7ffffe0,0x20,0,0x53053053 <unfinished ...>
10345 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10346 <... rseq resumed>)                  = 0
10345 <... rt_sigprocmask resumed>NULL,8) = 0
10346 set_robust_list(0x78d6d7fff9a0,24 <unfinished ...>
10345 futex(0x78d6d7fff990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10346,NULL,FUTEX_BI
<unfinished ...>
10346 <... set_robust_list resumed>)      = 0
10346 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10346 write(1,"Thread started. Active threads: 5\n",34) = 34
10346 mmap(0x78d6d4000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6cc000000
10346 mprotect(0x78d6cc000000,135168,PROT_READ|PROT_WRITE) = 0
10346 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d6ffe000
10346 mprotect(0x78d6d6fff000,8388608,PROT_READ|PROT_WRITE) = 0
10346 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10346 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10347 rseq(0x78d6d77fefe0,0x20,0,0x53053053 <unfinished ...>
10346 <... clone3 resumed>=>{parent_tid=[10347]},88) = 10347
10347 <... rseq resumed>)                  = 0
10346 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10347 set_robust_list(0x78d6d77fe9a0,24 <unfinished ...>
10346 <... rt_sigprocmask resumed>NULL,8) = 0
10347 <... set_robust_list resumed>)      = 0
10347 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10346 futex(0x78d6d77fe990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10347,NULL,FUTEX_BI
<unfinished ...>
10347 <... rt_sigprocmask resumed>NULL,8) = 0
10347 write(1,"Thread started. Active threads: 6\n",34) = 34
10347 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6c4000000
10347 munmap(0x78d6c8000000,67108864) = 0

```

```

10347 mprotect(0x78d6c4000000,135168,PROT_READ|PROT_WRITE) = 0
10347 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d67fd000
10347 mprotect(0x78d6d67fe000,8388608,PROT_READ|PROT_WRITE) = 0
10347 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10347 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10348]},88) = 10348
10348 rseq(0x78d6d6ffdf0,0x20,0,0x53053053 <unfinished ...>
10347 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10348 <... rseq resumed>) = 0
10347 <... rt_sigprocmask resumed>NULL,8) = 0
10348 set_robust_list(0x78d6d6ffd9a0,24 <unfinished ...>
10347 futex(0x78d6d6ffd990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10348,NULL,FUTEX_BI
<unfinished ...>
10348 <... set_robust_list resumed>) = 0
10348 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10348 write(1,"Thread started. Active threads: 7\n",34) = 34
10348 mmap(0x78d6c8000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6c8000000
10348 mprotect(0x78d6c8000000,135168,PROT_READ|PROT_WRITE) = 0
10348 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d5ffc000
10348 mprotect(0x78d6d5ffd000,8388608,PROT_READ|PROT_WRITE) = 0
10348 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10348 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10349 rseq(0x78d6d67fcfe0,0x20,0,0x53053053 <unfinished ...>
10348 <... clone3 resumed>=>{parent_tid=[10349]},88) = 10349
10349 <... rseq resumed>) = 0
10348 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10349 set_robust_list(0x78d6d67fc9a0,24 <unfinished ...>
10348 <... rt_sigprocmask resumed>NULL,8) = 0
10349 <... set_robust_list resumed>) = 0
10348 futex(0x78d6d67fc990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10349,NULL,FUTEX_BI
<unfinished ...>
10349 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10349 write(1,"Thread started. Active threads: 8\n",34) = 34
10349 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6bc000000
10349 munmap(0x78d6c0000000,67108864) = 0
10349 mprotect(0x78d6bc000000,135168,PROT_READ|PROT_WRITE) = 0
10349 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d57fb000
10349 mprotect(0x78d6d57fc000,8388608,PROT_READ|PROT_WRITE) = 0
10349 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10349 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10350 rseq(0x78d6d5ffbfe0,0x20,0,0x53053053 <unfinished ...>
10349 <... clone3 resumed>=>{parent_tid=[10350]},88) = 10350

```

```

10350 <... rseq resumed>                                = 0
10349 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10350 set_robust_list(0x78d6d5ffb9a0,24) = 0
10349 <... rt_sigprocmask resumed>NULL,8) = 0
10350 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10349 futex(0x78d6d5ffb990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10350,NULL,FUTEX_BI
<unfinished ...>
10350 <... rt_sigprocmask resumed>NULL,8) = 0
10350 write(1,"Thread started. Active threads: 9\n",34) = 34
10350 openat(AT_FDCWD,"/sys/devices/system/cpu/online",O_RDONLY|O_CLOEXEC)
= 3
10350 read(3,"0-19\n",1024)                                = 5
10350 close(3)                                              = 0
10350 mmap(0x78d6c0000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6c0000000
10350 mprotect(0x78d6c0000000,135168,PROT_READ|PROT_WRITE) = 0
10350 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d4ffa000
10350 mprotect(0x78d6d4ffb000,8388608,PROT_READ|PROT_WRITE) = 0
10350 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10350 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10351 rseq(0x78d6d57fafe0,0x20,0,0x53053053 <unfinished ...>
10350 <... clone3 resumed>=>{parent_tid=[10351]},88) = 10351
10351 <... rseq resumed>                                = 0
10350 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10351 set_robust_list(0x78d6d57fa9a0,24 <unfinished ...>
10350 <... rt_sigprocmask resumed>NULL,8) = 0
10351 <... set_robust_list resumed>                        = 0
10350 futex(0x78d6d57fa990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10351,NULL,FUTEX_BI
<unfinished ...>
10351 rt_sigprocmask(SIG_SETMASK, [],NULL,8) = 0
10351 write(1,"Thread started. Active threads: 10\n",35) = 35
10351 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d6b4000000
10351 munmap(0x78d6b8000000,67108864) = 0
10351 mprotect(0x78d6b4000000,135168,PROT_READ|PROT_WRITE) = 0
10351 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6d47f9000
10351 mprotect(0x78d6d47fa000,8388608,PROT_READ|PROT_WRITE) = 0
10351 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10351 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10352]},88) = 10352
10352 rseq(0x78d6d4ff9fe0,0x20,0,0x53053053 <unfinished ...>
10351 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10352 <... rseq resumed>                                = 0
10351 <... rt_sigprocmask resumed>NULL,8) = 0
10352 set_robust_list(0x78d6d4ff99a0,24 <unfinished ...>
10351 futex(0x78d6d4ff9990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10352,NULL,FUTEX_BI

```



```

<unfinished ...>
10352 <... set_robust_list resumed>      = 0
10352 rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
10352 write(1, "Thread started. Active threads: 11\n", 35) = 35
10352 mmap(0x78d6b8000000, 67108864, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x78d6b8000000
10352 mprotect(0x78d6b8000000, 135168, PROT_READ|PROT_WRITE) = 0
10352 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0)
= 0x78d6b37ff000
10352 mprotect(0x78d6b3800000, 8388608, PROT_READ|PROT_WRITE) = 0
10352 rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
10352 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10353]}, 88) = 10353
10353 rseq(0x78d6b3ffffe0, 0x20, 0, 0x53053053 <unfinished ...>
10352 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10353 <... rseq resumed>)                  = 0
10352 <... rt_sigprocmask resumed>NULL, 8) = 0
10353 set_robust_list(0x78d6b3fff9a0, 24 <unfinished ...>
10352 futex(0x78d6b3fff990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 10353, NULL, FUTEX_BI
<unfinished ...>
10353 <... set_robust_list resumed>      = 0
10353 rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
10353 write(1, "Thread started. Active threads: 12\n", 35) = 35
10353 mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x78d6ab600000
10353 munmap(0x78d6ab600000, 10485760) = 0
10353 munmap(0x78d6b0000000, 56623104) = 0
10353 mprotect(0x78d6ac000000, 135168, PROT_READ|PROT_WRITE) = 0
10353 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0)
= 0x78d6b2ffe000
10353 mprotect(0x78d6b2fff000, 8388608, PROT_READ|PROT_WRITE) = 0
10353 rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
10353 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10354]}, 88) = 10354
10354 rseq(0x78d6b37fefe0, 0x20, 0, 0x53053053 <unfinished ...>
10353 rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>
10354 <... rseq resumed>)                  = 0
10353 <... rt_sigprocmask resumed>NULL, 8) = 0
10354 set_robust_list(0x78d6b37fe9a0, 24 <unfinished ...>
10353 futex(0x78d6b37fe990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 10354, NULL, FUTEX_BI
<unfinished ...>
10354 <... set_robust_list resumed>      = 0
10354 rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
10354 write(1, "Thread started. Active threads: 13\n", 35) = 35
10354 mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x78d6a4000000
10354 munmap(0x78d6a8000000, 67108864) = 0
10354 mprotect(0x78d6a4000000, 135168, PROT_READ|PROT_WRITE) = 0
10354 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0)
= 0x78d6b27fd000

```

```

10354 mprotect(0x78d6b27fe000,8388608,PROT_READ|PROT_WRITE) = 0
10354 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10354 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10355 rseq(0x78d6b2ffdf0,0x20,0,0x53053053 <unfinished ...>
10354 <... clone3 resumed>=>{parent_tid=[10355]},88) = 10355
10355 <... rseq resumed> = 0
10354 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10355 set_robust_list(0x78d6b2ffd9a0,24 <unfinished ...>
10354 <... rt_sigprocmask resumed>NULL,8) = 0
10355 <... set_robust_list resumed> = 0
10354 futex(0x78d6b2ffd990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10355,NULL,FUTEX_BI
<unfinished ...>
10355 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10355 write(1,"Thread started. Active threads: 14\n",35) = 35
10355 mmap(0x78d6a8000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6a8000000
10355 mprotect(0x78d6a8000000,135168,PROT_READ|PROT_WRITE) = 0
10355 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6b1ffc000
10355 mprotect(0x78d6b1ffd000,8388608,PROT_READ|PROT_WRITE) = 0
10355 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10355 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
<unfinished ...>
10356 rseq(0x78d6b27fcfe0,0x20,0,0x53053053 <unfinished ...>
10355 <... clone3 resumed>=>{parent_tid=[10356]},88) = 10356
10356 <... rseq resumed> = 0
10355 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10356 set_robust_list(0x78d6b27fc9a0,24 <unfinished ...>
10355 <... rt_sigprocmask resumed>NULL,8) = 0
10356 <... set_robust_list resumed> = 0
10355 futex(0x78d6b27fc990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10356,NULL,FUTEX_BI
<unfinished ...>
10356 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10356 write(1,"Thread started. Active threads: 15\n",35) = 35
10356 mmap(NULL,134217728,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x78d69c000000
10356 munmap(0x78d6a0000000,67108864) = 0
10356 mprotect(0x78d69c000000,135168,PROT_READ|PROT_WRITE) = 0
10356 mmap(NULL,8392704,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK,-1,0)
= 0x78d6b17fb000
10356 mprotect(0x78d6b17fc000,8388608,PROT_READ|PROT_WRITE) = 0
10356 rt_sigprocmask(SIG_BLOCK,~[],[],8) = 0
10356 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SY
=>{parent_tid=[10357]},88) = 10357
10357 rseq(0x78d6b1ffbfe0,0x20,0,0x53053053 <unfinished ...>
10356 rt_sigprocmask(SIG_SETMASK,[], <unfinished ...>
10357 <... rseq resumed> = 0
10356 <... rt_sigprocmask resumed>NULL,8) = 0

```

```

10357 set_robust_list(0x78d6b1ffb9a0,24 <unfinished ...>
10356 futex(0x78d6b1ffb990,FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME,10357,NULL,FUTEX_BITSET
<unfinished ...>
10357 <... set_robust_list resumed>)      = 0
10357 rt_sigprocmask(SIG_SETMASK,[],NULL,8) = 0
10357 write(1,"Thread started. Active threads: 16\n",35) = 35
10357 write(1,"Thread finished. Active threads: 16\n",36) = 36
10357 mmap(0x78d6a0000000,67108864,PROT_NONE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0)
= 0x78d6a0000000
10357 mprotect(0x78d6a0000000,135168,PROT_READ|PROT_WRITE) = 0
10357 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10357 madvise(0x78d6b17fb000,8368128,MADV_DONTNEED) = 0
10357 exit(0)                                     = ?
10357 +++ exited with 0 +++
10356 <... futex resumed>)                    = 0
10356 write(1,"Thread finished. Active threads: 15\n",36) = 36
10356 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10356 madvise(0x78d6b1ffc000,8368128,MADV_DONTNEED) = 0
10356 exit(0)                                     = ?
10356 +++ exited with 0 +++
10355 <... futex resumed>)                    = 0
10355 write(1,"Thread finished. Active threads: 14\n",36) = 36
10355 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10355 madvise(0x78d6b27fd000,8368128,MADV_DONTNEED) = 0
10355 exit(0)                                     = ?
10355 +++ exited with 0 +++
10354 <... futex resumed>)                    = 0
10354 write(1,"Thread finished. Active threads: 13\n",36) = 36
10354 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10354 madvise(0x78d6b2ffe000,8368128,MADV_DONTNEED) = 0
10354 exit(0)                                     = ?
10354 +++ exited with 0 +++
10353 <... futex resumed>)                    = 0
10353 write(1,"Thread finished. Active threads: 12\n",36) = 36
10353 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10353 madvise(0x78d6b37ff000,8368128,MADV_DONTNEED) = 0
10353 exit(0)                                     = ?
10353 +++ exited with 0 +++
10352 <... futex resumed>)                    = 0
10352 munmap(0x78d6b17fb000,8392704)      = 0
10352 write(1,"Thread finished. Active threads: 11\n",36) = 36
10352 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10352 madvise(0x78d6d47f9000,8368128,MADV_DONTNEED) = 0
10352 exit(0)                                     = ?
10352 +++ exited with 0 +++
10351 <... futex resumed>)                    = 0
10351 munmap(0x78d6b1ffc000,8392704)      = 0
10351 write(1,"Thread finished. Active threads: 10\n",36) = 36

```

```

10351 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10351 madvise(0x78d6d4ffa000,8368128,MADV_DONTNEED) = 0
10351 exit(0) = ?
10351 +++ exited with 0 +++
10350 <... futex resumed>) = 0
10350 munmap(0x78d6b27fd000,8392704) = 0
10350 write(1,"Thread finished. Active threads: 9\n",35) = 35
10350 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10350 madvise(0x78d6d57fb000,8368128,MADV_DONTNEED) = 0
10350 exit(0) = ?
10350 +++ exited with 0 +++
10349 <... futex resumed>) = 0
10349 munmap(0x78d6b2ffe000,8392704) = 0
10349 write(1,"Thread finished. Active threads: 8\n",35) = 35
10349 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10349 madvise(0x78d6d5ffc000,8368128,MADV_DONTNEED) = 0
10349 exit(0) = ?
10349 +++ exited with 0 +++
10348 <... futex resumed>) = 0
10348 munmap(0x78d6b37ff000,8392704) = 0
10348 write(1,"Thread finished. Active threads: 7\n",35) = 35
10348 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10348 madvise(0x78d6d67fd000,8368128,MADV_DONTNEED) = 0
10348 exit(0) = ?
10348 +++ exited with 0 +++
10347 <... futex resumed>) = 0
10347 munmap(0x78d6d47f9000,8392704) = 0
10347 write(1,"Thread finished. Active threads: 6\n",35) = 35
10347 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10347 madvise(0x78d6d6ffe000,8368128,MADV_DONTNEED) = 0
10347 exit(0) = ?
10347 +++ exited with 0 +++
10346 <... futex resumed>) = 0
10346 munmap(0x78d6d4ffa000,8392704) = 0
10346 write(1,"Thread finished. Active threads: 5\n",35) = 35
10346 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10346 madvise(0x78d6d77ff000,8368128,MADV_DONTNEED) = 0
10346 exit(0) = ?
10346 +++ exited with 0 +++
10345 <... futex resumed>) = 0
10345 munmap(0x78d6d57fb000,8392704) = 0
10345 write(1,"Thread finished. Active threads: 4\n",35) = 35
10345 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10345 madvise(0x78d6e41fc000,8368128,MADV_DONTNEED) = 0
10345 exit(0) = ?
10345 +++ exited with 0 +++
10344 <... futex resumed>) = 0
10344 munmap(0x78d6d5ffc000,8392704) = 0

```

```
10344 write(1,"Thread finished. Active threads: 3\n",35) = 35
10344 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10344 madvise(0x78d6e49fd000,8368128,MADV_DONTNEED) = 0
10344 exit(0) = ?
10344 +++ exited with 0 +++
10343 <... futex resumed> = 0
10343 munmap(0x78d6d67fd000,8392704) = 0
10343 write(1,"Thread finished. Active threads: 2\n",35) = 35
10343 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10343 madvise(0x78d6e51fe000,8368128,MADV_DONTNEED) = 0
10343 exit(0) = ?
10343 +++ exited with 0 +++
10342 <... futex resumed> = 0
10342 munmap(0x78d6d6ffe000,8392704) = 0
10342 write(1,"Thread finished. Active threads: 1\n",35) = 35
10342 rt_sigprocmask(SIG_BLOCK,~[RT_1],NULL,8) = 0
10342 madvise(0x78d6e59ff000,8368128,MADV_DONTNEED) = 0
10342 exit(0) = ?
10341 <... futex resumed> = 0
10342 +++ exited with 0 +++
10341 munmap(0x78d6d77ff000,8392704) = 0
10341 write(1,"After sorting:\n",15) = 15
10341 write(1,"0 0 4 9 14 22 23 24 25 26 \n",27) = 27
10341 write(1,"Time: 0.0184601 seconds\n",24) = 24
10341 exit_group(0) = ?
10341 +++ exited with 0 +++
```