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-214Б-23

Студент: Гайдуков А.В.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 22.11.24

Москва, 2024

Постановка задачи

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.

Вариант задания 1. Отсортировать массив целых чисел при помощи битонической сортировки

Общий метод и алгоритм решения

Использованные системные вызовы:

- ssize_t write(int fd, const void *buf, size_t count); - Записывает `count` байтов из буфера `buf` в файл, связанный с файловым дескриптором `fd`.
- ssize_t read(int fd, void *buf, size_t count); - Читает до `count` байтов из файла, связанного с файловым дескриптором `fd`, и сохраняет их в буфере `buf`.
- int open(const char *pathname, int flags, mode_t mode); - Открывает файл по указанному пути `pathname` с заданными флагами `flags` и режимом доступа `mode`. Возвращает файловый дескриптор.
- int close(int fd); - Закрывает файловый дескриптор `fd`.
- void *malloc(size_t size); - Выделяет блок памяти размером `size` байт и возвращает указатель на начало блока.
- void free(void *ptr); - Освобождает блок памяти, на который указывает `ptr`, ранее выделенный с помощью `malloc`.
- int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine) (void *), void *arg); - Создает новый поток, который начинает выполнение с функции `start_routine`, передавая ей аргумент `arg`.
- int pthread_join(pthread_t thread, void **retval); - Ожидает завершения потока `thread` и возвращает его результат через `retval`.
- int clock_gettime(clockid_t clk_id, struct timespec *tp); - Получает текущее время для указанных часов `clk_id` и сохраняет его в структуре `timespec`.

Программа реализует битоническую сортировку (Bitonic Sort) для массива целых чисел. Она включает в себя как однопоточную, так и многопоточную версии алгоритма.

Основные шаги программы:

1. **Генерация случайного массива:**

   - Программа генерирует случайный массив целых чисел заданной длины (степень двойки) в указанном диапазоне.

2. **Сортировка массива:**

   - Программа выполняет битоническую сортировку двумя способами:

- Рекурсивная сортировка: Однопоточная версия, использующая рекурсивные вызовы для сортировки.

- Многопоточная сортировка: Использует `pthreads` для параллельного выполнения сортировки на нескольких потоках.

3. **Измерение времени выполнения:**

- Программа измеряет время выполнения как рекурсивной, так и многопоточной сортировки.

4. **Сохранение результатов:**

- Отсортированные массивы сохраняются в файлы `recursion.txt` и `multiThread.txt`.

Код программы

bitSort.cpp

```
#include <pthread.h>
#include <string.h>
#include <ctype.h>
#include <unistd.h>


void compare_and_swap(int *arr, int i, int j, int dir) {
    if (dir == (arr[i] > arr[j])) {
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
}

void bitonic_merge(int *arr, int low, int cnt, int dir) {
    if (cnt > 1) {
        for (int i = low; i < low + cnt / 2; i++) {
            compare_and_swap(arr, i, i + cnt / 2, dir);
        }
        bitonic_merge(arr, low, cnt / 2, dir);
        bitonic_merge(arr, low + cnt / 2, cnt / 2, dir);
    }
}

void bitonic_sort(int *arr, int low, int cnt, int dir) {
    if (cnt > 1) {
        bitonic_sort(arr, low, cnt / 2, 1);
        bitonic_sort(arr, low + cnt / 2, cnt / 2, 0);
        bitonic_merge(arr, low, cnt, dir);
    }
}
```

```
void bitsort(int *arr, int n, int up) {
    bitonic_sort(arr, 0, n, up);
}
```

intstr.cpp

```
#include <pthread.h>
#include <string.h>
#include <ctype.h>
#include <unistd.h>
#include <random>
#include <math.h>

#include "intstr.h"


int hasOnlyNums(char* num){
    for (int i = 0; i < strlen(num) && num[i] != '\n'; i++)
    {
        if(!isdigit(num[i]) || (num[i] == '-' && i != 0)){
            return 0;
        }
    }
    return 1;
}

char* double_to_str(double num, char* str){
    int integer_num = (int)num;
    str = int_to_str(integer_num, str);
    num -= integer_num;
    num *= 100000000;
    integer_num = (int)num;

    char rev_num[100];
    int len = 0;
    for (int i = 0; integer_num != 0; i++)
    {
        rev_num[i] = '0' + integer_num % 10;
        integer_num /= 10;
        len = i + 1;
    }

    char* tmp;
    tmp = (char*)realloc(str, sizeof(char) * (sizeof(str) + len + 1));
    if(tmp == NULL){
        free(str);
        return NULL;
```

```
    }

    str = tmp;
    str[strlen(str)] = '.';
    int stln = strlen(str);
    for (int i = len - 1; i >= 0; i--, stln++)
    {
        str[stln] = rev_num[i];
    }
    str[stln] = '\0';
    return str;
}

int str_to_int(char* num){
    int res = 0, beg = 0;
    if(!hasOnlyNums(num)){
        return 0;
    }
    else if(num[0] == '-'){
        ++beg;
    }
    for (int i = beg; i < strlen(num) && num[i] != '\n'; i++)
    {
        res = (res * 10) + num[i] - '0';
    }
    return res;
}

char* int_to_str(int number, char* string){
    char rev_num[100];
    int len = 0;
    for (int i = 0; number != 0; i++)
    {
        rev_num[i] = '0' + number % 10;
        number /= 10;
        len = i + 1;
    }
    string = (char*)malloc(sizeof(char) * (len + 1));
    if(string == NULL){
        return " ";
    }
    for (int i = len - 1, j = 0; i >= 0; i--, j++)
    {
        string[j] = rev_num[i];
    }
    string[len] = '\0';
    return string;
}
```

main.cpp

```
#include <pthread.h>
#include <string.h>
#include <ctype.h>
#include <unistd.h>
#include <random>
#include <math.h>
#include <time.h>
#include <sys/sysinfo.h>
#include <sys/resource.h>

#include "intstr.h"
#include "bitSort.h"
#include "myio.h"
#include "multiThread.h"




int get_random_array(int*& array, int num, int lb, int rb){
    int amount = (1 << num);
    array = (int*)malloc(sizeof(int) * amount);
    if(array == NULL){
        return 0;
    }
    std::random_device rd;
    std::mt19937 gen(rd());
    std::uniform_int_distribution<> dis(lb, rb);
    for (int i = 0; i < amount; i++)
    {
        array[i] = dis(gen);
    }
    return 1;
}

//Format: ./<function> <lengthOfArray(the degree of two)> <lb> <rb>
int main(int argc, char* argv[]){
    if(argc != 4){
        my_write("Wrong amount of arguments!\n");
        return -1;
    }
    else if(!hasOnlyNums(argv[1]) || !hasOnlyNums(argv[2]) || !hasOnlyNums(argv[3])){
        my_write("Wrong type of argument! All arguments must be integer!\n");
        return -1;
    }
    int amount = str_to_int(argv[1]), lb = str_to_int(argv[2]), rb =
str_to_int(argv[3]);
```

```
    if(lb > rb || amount <= 0){
        my_write("Border | amount error!\n");
        return -1;
    }

    int *arrayForRec, *arrayForLin;

    get_random_array(arrayForRec, amount, lb, rb);
    get_random_array(arrayForLin, amount, lb, rb);

    amount = (1 << amount);
    my_write("How many threads do you want to use?\n");

    int numCPU = sysconf(_SC_NPROCESSORS_ONLN);

    my_write("Currently the amount of CPUs that are avaluable is ");
    char* chr;
    my_write(int_to_str(numCPU, chr));
    free(chr);
    my_write("\n");
    char buf[BUFSIZ];
    my_read(buf);
    if(hasOnlyNums(buf) && str_to_int(buf) > 0){
        //for recursion
        {
            struct timespec start, stop;
            clock_gettime(CLOCK_REALTIME, &start);
            bitsort(arrayForRec, amount, 1);
            clock_gettime(CLOCK_REALTIME, &stop);
            char *str;
            str = double_to_str((stop.tv_sec - start.tv_sec) * 1e6 + (stop.tv_nsec -
start.tv_nsec) / 1e3, str);

            my_write("Time for recursion: ");
            my_write(str);
            my_write("\n");
            free(str);
        }
        //for multi-thread
        {
            struct timespec start, stop;
            clock_gettime(CLOCK_REALTIME, &start);

            bitonicSort2(arrayForLin, amount, str_to_int(buf), 1);

            clock_gettime(CLOCK_REALTIME, &stop);
            char *str;
            str = double_to_str((stop.tv_sec - start.tv_sec) * 1e6 + (stop.tv_nsec -
start.tv_nsec) / 1e3, str);
```

```
        my_write("Time for multi-thread: ");
        my_write(str);
        my_write("\n");
        free(str);
    }

    char fileName1[100], fileName2[100];
    strcpy(fileName1, "recursion.txt");
    strcpy(fileName2, "multiThread.txt");

    int descriptor1, descriptor2;
    descriptor1 = file_open(fileName1);
    for (int i = 0; i < amount; i++)
    {
        char* str = int_to_str(arrayForRec[i], str);
        write(descriptor1, str, strlen(str));
        if(i % 15 == 0){
            write(descriptor1, "\n", 1);
        }
        else{
            write(descriptor1, " ", 1);
        }
        free(str);
    }
    file_close(descriptor1);

    descriptor2 = file_open(fileName2);
    for (int i = 0; i < amount; i++)
    {
        char* str = int_to_str(arrayForLin[i], str);
        write(descriptor2, str, strlen(str));
        if(i % 15 == 0){
            write(descriptor2, "\n", 1);
        }
        else{
            write(descriptor2, " ", 1);
        }
        free(str);
    }
    file_close(descriptor2);

    my_write("\nOK\n");
    free(arrayForLin);
    free(arrayForRec);
    return 0;
}
else{
    my_write("Wrong amount of CPUs\n");
    return -1;
```

```
    }
    free(arrayForLin);
    free(arrayForRec);
    return 0;
}
```

multiThread.cpp

```
#include <pthread.h>
#include <string.h>
#include <ctype.h>
#include <unistd.h>
#include <random>
#include <math.h>

#include "myio.h"
#include "intstr.h"

struct PTHREAD_DATA{
    int *paddedValues;
    unsigned int threadId;
    unsigned int chunkSize;
    unsigned int mergeStep;
    unsigned int bitonicSequenceSize;
};

void swap(int *a, int *b){
    int tmp=*a;
    *a=*b;
    *b=tmp;
}

void reverse(int* first, int* last)
{
  while (1)
    if( first == last || first == --last )
      return;
    else{
        swap( first, last);
        ++first;
    }
}

void compareAndSwap(int paddedValues[], unsigned int threadId,
bitonicSequenceSize)unsigned int chunkSize, unsigned int mergeStep, unsigned int
{
    unsigned int startIndex = threadId * chunkSize;
    unsigned int endIndex = (threadId + 1) * chunkSize;
```

```
    // Process the chunk assigned to this thread
    for (unsigned int currentIndex = startIndex; currentIndex < endIndex;
currentIndex++)
    {
        // Find the element to compare with
        unsigned int compareIndex = currentIndex ^ mergeStep;

        // Only compare if the compareIndex is greater (to avoid duplicate swaps)
        if (compareIndex > currentIndex)
        {
            bool shouldSwap = false;

            // Determine if we should swap based on the current subarray's sorting
direction
            if ((currentIndex & bitonicSequenceSize) == 0)  // First half of subarray
(ascending)
            {
                shouldSwap = (paddedValues[currentIndex] > paddedValues[compareIndex]);
            }
            else  // Second half of subarray (descending)
            {
                shouldSwap = (paddedValues[currentIndex] < paddedValues[compareIndex]);
            }

            // Perform the swap if necessary
            if (shouldSwap)
            {
                swap(&paddedValues[currentIndex], &paddedValues[compareIndex]);
            }
        }
    }
}

void *thread_func(void* arg)
{
  struct PTHREAD_DATA* data=(struct PTHREAD_DATA*)arg;
  if(data){
    compareAndSwap(data->paddedValues, data->threadId, data->chunkSize,
data->mergeStep, data->bitonicSequencesize);
    free(data);
  }
  return NULL;
}

void bitonicSort2(int values[], unsigned int arrayLength, unsigned int numThreads, int
sortOrder)
{
    // Step 1: Pad the array to the next power of 2
    unsigned int paddedLength = arrayLength; //(already padded according to the main,
the length is a degree of two)
```

```
    // Step 2: Determine chunk size for each thread
    unsigned int chunkSize = paddedLength / numThreads;
    int *paddedValues=values;
    // Step 3: Iteratively build and merge bitonic sequences
    // Outer loop: controls the size of bitonic sequences
    for (unsigned int bitonicSequenceSize = 2; bitonicSequenceSize <= paddedLength;
bitonicSequenceSize *= 2)
    {
        // Middle loop: controls the size of sub-sequences being merged
        for (unsigned int mergeStep = bitonicSequenceSize / 2; mergeStep > 0; mergeStep
/= 2)
        {
            // Step 4: Use multiple threads to compare and swap elements in parallel
            pthread_t *p_thread = (pthread_t*)malloc(numThreads*sizeof(pthread_t));

            if(p_thread!=NULL){


                // Thread creation loop
                for (unsigned int threadId = 0; threadId < numThreads; threadId++)
                {
                    struct PTHREAD_DATA *pdt = (struct PTHREAD_DATA
*)malloc(sizeof(struct PTHREAD_DATA)); //free in thread
                    if(pdt!=NULL){
                        pdt->paddedValues=paddedValues;
                        pdt->threadId=threadId;
                        pdt->chunkSize=chunkSize;
                        pdt->mergeStep=mergeStep;
                        pdt->bitonicSequenceSize=bitonicSequenceSize;
                        pthread_create(&p_thread[threadId],/*&attr*/
NULL,thread_func,(void*)pdt);
                    }
                    else{
                        my_write("malloc error for PTHREAD_DATA in bitonicSort\n");
                        free(p_thread);
                        if(paddedValues!=values)
                            free(paddedValues);
                        return;
                    }
                }

//                pthread_attr_destroy(&attr);

                // Wait for all threads to complete this stage
                for(unsigned int k=0;k<numThreads;k++) {
                    pthread_join(p_thread[k], NULL);
                }

                free(p_thread);
            }
```

```
            else{
                my_write("malloc error for paddedValues in bitonicSort\n");
                if(paddedValues!=values)
                    free(paddedValues);
                return;
            }
        }
    }

    // Step 5: Copy back the sorted values

    if(paddedValues!=values){
        memcpy(values, paddedValues, arrayLength * sizeof(int));
        free(paddedValues);
    }


    // Step 6: If descending order is required, reverse the array
    if (sortOrder == 0){
        reverse(values, values + arrayLength);
    }
}
```

myio.cpp

```
#include <pthread.h>
#include <string.h>
#include <ctype.h>
#include <unistd.h>

#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#define BUFSIZ 8192

ssize_t my_write(char* str){
    return write(STDOUT_FILENO, str, strlen(str));
}

ssize_t my_read(char* buf){
    return read(STDIN_FILENO, buf, BUFSIZ);
}

int file_open(char* filename){
    return open(filename, O_CREAT | O_TRUNC | O_RDWR , S_IRUSR | S_IWUSR | S_IRGRP |
S_IWGRP | S_IROTH | S_IWOTH);
}

int write_to_file(int descriptor, char* str){
```

```
    if(descriptor == -1){
        my_write("File error!\n");
        return 0;
    }
    write(descriptor, str, strlen(str));
    return 1;
}

int file_close(int descriptor){
    close(descriptor);
    return 0;
}
```

Протокол работы программы

Рассчитаем время исполнения, ускорение и эффективность при длине массива, равной 2^{16}. Число потоков равно степеням двойки из-за специфики распараллеленного алгоритма битонической сортировки

Число потоков	Время исполнения (мс)	Ускорение	Эффективность
1	**54764**	**1.0**	**1.0**
2	**50179**	**1.0913**	**0.54565**
4	**27031**	**2.0259**	**0.50648**
8	**27375**	**2.00051**	**0.25006**
16	**44306**	**1.23604**	**0.07725**
32	**68912**	**0.79470**	**0.02483**

Ускорение показывает, во сколько раз применение параллельного алгоритма уменьшает время решения задачи по сравнению с последовательным алгоритмом. Ускорение определяется величиной $S_N = T_1 / T_N$, где T_1 – время выполнения на одном потоке, T_N – время выполнения на N потоках.

Эффективность показывает, насколько хорошо используются ресурсы при параллельном выполнении алгоритма. Она определяется как отношение ускорения к количеству используемых потоков. Эффективность вычисляется по формуле $E = S_N / N$, где S_N – ускорение, а N – количество потоков.

Strace:

```
5450  execve("./main", ["./main", "4", "1", "10000"], 0x7ffd5ce0cc20 /* 49 vars */) = 0
5450  brk(NULL)                         = 0x55e989e12000
```

```
5450  arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc61ce9820) = -1 EINVAL (Недопустимый аргумент)
5450  mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f73f62fb000
5450  access("/etc/ld.so.preload", R_OK) = -1 ENOENT (Нет такого файла или каталога)
5450  openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
5450  newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=85143, ...}, AT_EMPTY_PATH) = 0
5450  mmap(NULL, 85143, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f73f62e6000
5450  close(3)                  = 0
5450  openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libstdc++.so.6", O_RDONLY|O_CLOEXEC) = 3
5450  read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
5450  newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2260296, ...}, AT_EMPTY_PATH) = 0
5450  mmap(NULL, 2275520, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f73f60ba000
5450  mprotect(0x7f73f6154000, 1576960, PROT_NONE) = 0
5450  mmap(0x7f73f6154000, 1118208, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x9a000) = 0x7f73f6154000
5450  mmap(0x7f73f6265000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1ab000) = 0x7f73f6265000
5450  mmap(0x7f73f62d5000, 57344, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x21a000) = 0x7f73f62d5000
5450  mmap(0x7f73f62e3000, 10432, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f73f62e3000
5450  close(3)                  = 0
5450  openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3
5450  read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
5450  newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=125488, ...}, AT_EMPTY_PATH) = 0
5450  mmap(NULL, 127720, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f73f609a000
5450  mmap(0x7f73f609d000, 94208, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f73f609d000
5450  mmap(0x7f73f60b4000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1a000) = 0x7f73f60b4000
5450  mmap(0x7f73f60b8000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1d000) = 0x7f73f60b8000
5450  close(3)                  = 0
5450  openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
5450  read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"..., 832) = 832
5450  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
5450  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
5450  pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0I\17\357\204\3$\f\221\2039x\324\224\323\236S"..., 68, 896) = 68
5450  newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2220400, ...}, AT_EMPTY_PATH) = 0
5450  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
5450  mmap(NULL, 2264656, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f73f5e71000
5450  mprotect(0x7f73f5e99000, 2023424, PROT_NONE) = 0
5450  mmap(0x7f73f5e99000, 1658880, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f73f5e99000
5450  mmap(0x7f73f602e000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1bd000) = 0x7f73f602e000
5450  mmap(0x7f73f6087000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x215000) = 0x7f73f6087000
5450  mmap(0x7f73f608d000, 52816, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f73f608d000
5450  close(3)                  = 0
5450  openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC) = 3
5450  read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
5450  newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0
5450  mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f73f5d8a000
5450  mmap(0x7f73f5d98000, 507904, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe000) = 0x7f73f5d98000
```

```
5450  mmap(0x7f73f5e14000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8a000) = 0x7f73f5e14000
5450  mmap(0x7f73f5e6f000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0xe4000) = 0x7f73f5e6f000
5450  close(3)                          = 0
5450  mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f73f5d88000
5450  arch_prctl(ARCH_SET_FS, 0x7f73f5d893c0) = 0
5450  set_tid_address(0x7f73f5d89690)   = 5450
5450  set_robust_list(0x7f73f5d896a0, 24) = 0
5450  rseq(0x7f73f5d89d60, 0x20, 0, 0x53053053) = 0
5450  mprotect(0x7f73f6087000, 16384, PROT_READ) = 0
5450  mprotect(0x7f73f5e6f000, 4096, PROT_READ) = 0
5450  mprotect(0x7f73f60b8000, 4096, PROT_READ) = 0
5450  mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f73f5d86000
5450  mprotect(0x7f73f62d5000, 45056, PROT_READ) = 0
5450  mprotect(0x55e987b56000, 4096, PROT_READ) = 0
5450  mprotect(0x7f73f6335000, 8192, PROT_READ) = 0
5450  prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
5450  munmap(0x7f73f62e6000, 85143)     = 0
5450  getrandom("\x8c\x7e\x7b\x7e\xfc\x15\xf5\xf8", 8, GRND_NONBLOCK) = 8
5450  brk(NULL)                         = 0x55e989e12000
5450  brk(0x55e989e33000)               = 0x55e989e33000
5450  write(1, "How many threads do you want to "..., 37) = 37
5450  openat(AT_FDCWD, "/sys/devices/system/cpu/online", O_RDONLY|O_CLOEXEC) = 3
5450  read(3, "0-5\n", 1024)            = 4
5450  close(3)                          = 0
5450  write(1, "Currently the amount of CPUs tha"..., 51) = 51
5450  write(1, "6", 1)                  = 1
5450  write(1, "\n", 1)                 = 1
5450  read(0, "2\n", 8192)              = 2
5450  write(1, "Time for recursion: ", 20) = 20
5450  write(1, "5.64100000", 10)        = 10
5450  write(1, "\n", 1)                 = 1
5450  rt_sigaction(SIGRT_1, {sa_handler=0x7f73f5f02870, sa_mask=[], sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f73f5eb3520}, NULL, 8) = 0
5450  rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
5450  mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7f73f5585000
5450  mprotect(0x7f73f5586000, 8388608, PROT_READ|PROT_WRITE) = 0
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450  clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910, parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} => {parent_tid=[5453]}, 88) = 5453
5453  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5453  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5453  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5450  mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>
5453  <... set_robust_list resumed>)    = 0
5450  <... mmap resumed>)               = 0x7f73f4d84000
```

```
5453  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450  mprotect(0x7f73f4d85000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>
5453  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  <... mprotect resumed>)           = 0
5453  mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_NORESERVE, -1, 0
<unfinished ...>
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5453  <... mmap resumed>)               = 0x7f73ecd84000
5450  <... rt_sigprocmask resumed>[], 8) = 0
5453  munmap(0x7f73ecd84000, 52936704 <unfinished ...>
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910,
parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} <unfinished
...>
5453  <... munmap resumed>)             = 0
5453  munmap(0x7f73f4000000, 14172160 <unfinished ...>
5450  <... clone3 resumed> => {parent_tid=[5454]}, 88) = 5454
5453  <... munmap resumed>)             = 0
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5453  mprotect(0x7f73f0000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5454  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5453  <... mprotect resumed>)           = 0
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5453, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5454  <... rseq resumed>)               = 0
5453  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5454  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5453  <... rt_sigprocmask resumed>NULL, 8) = 0
5454  <... set_robust_list resumed>)    = 0
5453  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED <unfinished ...>
5454  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5453  <... madvise resumed>)            = 0
5454  <... rt_sigprocmask resumed>NULL, 8) = 0
5453  exit(0 <unfinished ...>
5454  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5453  <... exit resumed>)               = ?
5454  <... rt_sigprocmask resumed>NULL, 8) = 0
5454  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED <unfinished ...>
5453  +++ exited with 0 +++
5450  <... futex resumed>)              = 0
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5454, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5454  <... madvise resumed>)            = 0
5454  exit(0)                           = ?
5450  <... futex resumed>)              = 0
5454  +++ exited with 0 +++
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910,
parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} =>
{parent_tid=[5455]}, 88) = 5455
5455  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
```

```
5455  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5455  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5455  <... set_robust_list resumed>)    = 0
5450  <... rt_sigprocmask resumed>[], 8) = 0
5455  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910,
parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} <unfinished ...>
5455  <... rt_sigprocmask resumed>NULL, 8) = 0
5455  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
5450  <... clone3 resumed> => {parent_tid=[5456]}, 88) = 5456
5455  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5456  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5455  <... madvise resumed>)            = 0
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5455, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5456  <... rseq resumed>)               = 0
5455  exit(0 <unfinished ...>
5456  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5455  <... exit resumed>)               = ?
5456  <... set_robust_list resumed>)    = 0
5455  +++ exited with 0 +++
5450  <... futex resumed>)              = 0
5456  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5456, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5456  <... rt_sigprocmask resumed>NULL, 8) = 0
5456  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
5456  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED) = 0
5456  exit(0)                           = ?
5450  <... futex resumed>)              = 0
5456  +++ exited with 0 +++
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910,
parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} =>
{parent_tid=[5457]}, 88) = 5457
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5457  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5457  <... rseq resumed>)               = 0
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5457  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5450  <... rt_sigprocmask resumed>[], 8) = 0
5457  <... set_robust_list resumed>)    = 0
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910,
parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} <unfinished ...>
5457  rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
```

```
5450  <... clone3 resumed> => {parent_tid=[5458]}, 88) = 5458
5458  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5457  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5458  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5457  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5457, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5458  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5457  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED <unfinished ...>
5458  <... set_robust_list resumed>)    = 0
5457  <... madvise resumed>)            = 0
5458  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5457  exit(0 <unfinished ...>
5458  <... rt_sigprocmask resumed>NULL, 8) = 0
5457  <... exit resumed>)               = ?
5458  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5450  <... futex resumed>)              = 0
5457  +++ exited with 0 +++
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5458, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5458  <... rt_sigprocmask resumed>NULL, 8) = 0
5458  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED) = 0
5458  exit(0)                           = ?
5450  <... futex resumed>)              = 0
5458  +++ exited with 0 +++
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450  clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910,
parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} =>
{parent_tid=[5459]}, 88) = 5459
5459  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5459  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5459  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5459  <... set_robust_list resumed>)    = 0
5450  <... rt_sigprocmask resumed>[], 8) = 0
5459  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450  clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910,
parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} <unfinished
...>
5459  <... rt_sigprocmask resumed>NULL, 8) = 0
5459  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5450  <... clone3 resumed> => {parent_tid=[5460]}, 88) = 5460
5459  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5460  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5459  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED <unfinished ...>
```

```
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5459, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5460  <... rseq resumed>)               = 0
5459  <... madvise resumed>)            = 0
5460  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5459  exit(0 <unfinished ...>
5460  <... set_robust_list resumed>)    = 0
5459  <... exit resumed>)               = ?
5460  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450  <... futex resumed>)              = 0
5459  +++ exited with 0 +++
5460  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5460, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5460  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
5460  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED) = 0
5460  exit(0)                           = ?
5450  <... futex resumed>)              = 0
5460  +++ exited with 0 +++
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910,
parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} =>
{parent_tid=[5461]}, 88) = 5461
5461  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5461  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5461  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5461  <... set_robust_list resumed>)    = 0
5450  <... rt_sigprocmask resumed>[], 8) = 0
5461  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910,
parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} <unfinished
...>
5461  <... rt_sigprocmask resumed>NULL, 8) = 0
5461  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
5450  <... clone3 resumed> => {parent_tid=[5462]}, 88) = 5462
5462  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5461  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5462  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5461  <... madvise resumed>)            = 0
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5461, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5462  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5461  exit(0 <unfinished ...>
5462  <... set_robust_list resumed>)    = 0
5461  <... exit resumed>)               = ?
5462  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5461  +++ exited with 0 +++
```

```
5450  <... futex resumed>)              = 0
5462  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5462, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5462  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
5462  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED) = 0
5462  exit(0)                           = ?
5450  <... futex resumed>)              = 0
5462  +++ exited with 0 +++
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910,
parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} =>
{parent_tid=[5463]}, 88) = 5463
5463  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5463  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5463  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5463  <... set_robust_list resumed>)    = 0
5450  <... rt_sigprocmask resumed>[], 8) = 0
5463  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910,
parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} <unfinished
...>
5463  <... rt_sigprocmask resumed>NULL, 8) = 0
5463  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5450  <... clone3 resumed> => {parent_tid=[5464]}, 88) = 5464
5463  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5464  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5463  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED <unfinished ...>
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5463, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5464  <... rseq resumed>)               = 0
5463  <... madvise resumed>)            = 0
5464  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5463  exit(0 <unfinished ...>
5464  <... set_robust_list resumed>)    = 0
5463  <... exit resumed>)               = ?
5464  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450  <... futex resumed>)              = 0
5463  +++ exited with 0 +++
5464  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5464, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5464  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
5464  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED) = 0
5464  exit(0)                           = ?
5450  <... futex resumed>)              = 0
5464  +++ exited with 0 +++
```

```
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910, parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} => {parent_tid=[5465]}, 88) = 5465
5465  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5465  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5465  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5465  <... set_robust_list resumed>)    = 0
5450  <... rt_sigprocmask resumed>[], 8) = 0
5465  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910, parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} <unfinished ...>
5465  <... rt_sigprocmask resumed>NULL, 8) = 0
5465  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5450  <... clone3 resumed> => {parent_tid=[5466]}, 88) = 5466
5466  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5465  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5466  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5465  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED <unfinished ...>
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5465, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
5466  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5465  <... madvise resumed>)            = 0
5466  <... set_robust_list resumed>)    = 0
5465  exit(0 <unfinished ...>
5466  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5465  <... exit resumed>)               = ?
5466  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  <... futex resumed>)              = 0
5466  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5465  +++ exited with 0 +++
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5466, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
5466  <... rt_sigprocmask resumed>NULL, 8) = 0
5466  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED) = 0
5466  exit(0)                           = ?
5450  <... futex resumed>)              = 0
5466  +++ exited with 0 +++
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910, parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} => {parent_tid=[5467]}, 88) = 5467
5467  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5467  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
```

```
5467  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5467  <... set_robust_list resumed>)    = 0
5450  <... rt_sigprocmask resumed>[], 8) = 0
5467  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910, parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} <unfinished ...>
5467  <... rt_sigprocmask resumed>NULL, 8) = 0
5467  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5450  <... clone3 resumed> => {parent_tid=[5468]}, 88) = 5468
5467  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5468  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5467  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED <unfinished ...>
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5467, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
5468  <... rseq resumed>)               = 0
5467  <... madvise resumed>)            = 0
5468  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5467  exit(0 <unfinished ...>
5468  <... set_robust_list resumed>)    = 0
5467  <... exit resumed>)               = ?
5468  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450  <... futex resumed>)              = 0
5467  +++ exited with 0 +++
5468  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5468, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>
5468  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
5468  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED) = 0
5468  exit(0)                           = ?
5450  <... futex resumed>)              = 0
5468  +++ exited with 0 +++
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910, parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} => {parent_tid=[5469]}, 88) = 5469
5469  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5469  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5469  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5469  <... set_robust_list resumed>)    = 0
5450  <... rt_sigprocmask resumed>[], 8) = 0
5469  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910, parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} <unfinished ...>
5469  <... rt_sigprocmask resumed>NULL, 8) = 0
```

```
5469  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5450  <... clone3 resumed> => {parent_tid=[5470]}, 88) = 5470
5470  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5469  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5470  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5469  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED <unfinished ...>
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5469, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5470  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5469  <... madvise resumed>)            = 0
5470  <... set_robust_list resumed>)    = 0
5469  exit(0 <unfinished ...>
5470  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5469  <... exit resumed>)               = ?
5470  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  <... futex resumed>)              = 0
5470  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5469  +++ exited with 0 +++
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5470, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5470  <... rt_sigprocmask resumed>NULL, 8) = 0
5470  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED) = 0
5470  exit(0)                           = ?
5450  <... futex resumed>)              = 0
5470  +++ exited with 0 +++
5450  rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5584910,
parent_tid=0x7f73f5584910, exit_signal=0, stack=0x7f73f4d84000, stack_size=0x7fff00, tls=0x7f73f5584640} =>
{parent_tid=[5471]}, 88) = 5471
5471  rseq(0x7f73f5584fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5471  <... rseq resumed>)               = 0
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5471  set_robust_list(0x7f73f5584920, 24 <unfinished ...>
5450  rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>
5471  <... set_robust_list resumed>)    = 0
5450  <... rt_sigprocmask resumed>[], 8) = 0
5471  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f73f5d85910,
parent_tid=0x7f73f5d85910, exit_signal=0, stack=0x7f73f5585000, stack_size=0x7fff00, tls=0x7f73f5d85640} <unfinished
...>
5471  <... rt_sigprocmask resumed>NULL, 8) = 0
5471  rt_sigprocmask(SIG_BLOCK, ~[RT_1],  <unfinished ...>
5450  <... clone3 resumed> => {parent_tid=[5472]}, 88) = 5472
5471  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5472  rseq(0x7f73f5d85fe0, 0x20, 0, 0x53053053 <unfinished ...>
5450  <... rt_sigprocmask resumed>NULL, 8) = 0
5471  madvise(0x7f73f4d84000, 8368128, MADV_DONTNEED <unfinished ...>
```

```
5450  futex(0x7f73f5584910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5471, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5472  <... rseq resumed>)              = 0
5471  <... madvise resumed>)           = 0
5472  set_robust_list(0x7f73f5d85920, 24 <unfinished ...>
5471  exit(0 <unfinished ...>
5472  <... set_robust_list resumed>)    = 0
5471  <... exit resumed>)              = ?
5472  rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
5450  <... futex resumed>)             = 0
5471  +++ exited with 0 +++
5472  <... rt_sigprocmask resumed>NULL, 8) = 0
5450  futex(0x7f73f5d85910, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 5472, NULL,
FUTEX_BITSET_MATCH_ANY <unfinished ...>
5472  rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0
5472  madvise(0x7f73f5585000, 8368128, MADV_DONTNEED) = 0
5472  exit(0)                          = ?
5450  <... futex resumed>)             = 0
5472  +++ exited with 0 +++
5450  write(1, "Time for multi-thread: ", 23) = 23
5450  write(1, "6913.13699999", 13)     = 13
5450  write(1, "\n", 1)                 = 1
5450  openat(AT_FDCWD, "recursion.txt", O_RDWR|O_CREAT|O_TRUNC, 0666) = 3
5450  write(3, "110", 3)                = 3
5450  write(3, "\n", 1)                 = 1
5450  write(3, "230", 3)                = 3
5450  write(3, " ", 1)                  = 1
5450  write(3, "652", 3)                = 3
5450  write(3, " ", 1)                  = 1
5450  write(3, "1215", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "1533", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "2624", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "2981", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "3329", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "4941", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "5275", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "5649", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "6707", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "7889", 4)               = 4
5450  write(3, " ", 1)                  = 1
5450  write(3, "8464", 4)               = 4
```

```
5450  write(3, " ", 1)               = 1
5450  write(3, "8821", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "9190", 4)            = 4
5450  write(3, "\n", 1)              = 1
5450  close(3)                       = 0
5450  openat(AT_FDCWD, "multiThread.txt", O_RDWR|O_CREAT|O_TRUNC, 0666) = 3
5450  write(3, "474", 3)             = 3
5450  write(3, "\n", 1)              = 1
5450  write(3, "819", 3)             = 3
5450  write(3, " ", 1)               = 1
5450  write(3, "965", 3)             = 3
5450  write(3, " ", 1)               = 1
5450  write(3, "2087", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "2639", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "3141", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "4403", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "5528", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "6052", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "7637", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "8660", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "8684", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "9048", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "9064", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "9856", 4)            = 4
5450  write(3, " ", 1)               = 1
5450  write(3, "9943", 4)            = 4
5450  write(3, "\n", 1)              = 1
5450  close(3)                       = 0
5450  write(1, "\nOK\n", 4)          = 4
5450  exit_group(0)                  = ?
5450  +++ exited with 0 +++
```

Вывод

Я научился писать многопоточные алгоритмы и приложения, разобрался с теорией по разделу и разобрал, какие проблемы могут возникнуть при разработке многопоточного приложения.