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-211Б-23

Студент: Фоменко А.С. Преподаватель: Бахарев В.Д. Оценка:

Дата: 24.12.24

Москва, 2024

# Постановка задачи

## Вариант 18.

Цель работы

Приобретение практических навыков в:

* Управление потоками в ОС
* Обеспечение синхронизации между потоками

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.

Найти образец в строке наивным алгоритмом.

# Общий метод и алгоритм решения

Использованные системные вызовы:

* + int pthread\_create(pthread\_t \*thread, const pthread\_attr\_t \*attr,
  + void \*(\*start)(void \*), void \*arg)– создание потока
  + int pthread\_join (pthread\_t THREAD\_ID, void \*\* DATA) – ожидание завершения потока

|  |  |  |  |
| --- | --- | --- | --- |
| Количество потоков | Время, мс | Ускорение | Эффективность |
| 1 | 1.4 | 1 | 1 |
| 2 | 1384 | 0.001 | 0.0005 |
| 3 | 2092 | 0.00067 | 0.00022 |
| 4 | 2832 | 0.00049 | 0.00012 |
| 6 | 3388 | 0.00041 | 0.000068 |
| 8 | 4021 | 0.00035 | 0.000044 |

Данные подсчитаны при строке длины 1000000000, слово находится на позиции 82346

# Код программы

## main.c

#include <stdint.h>

#include <stdbool.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <stdio.h>

#include <string.h>

#include <pthread.h>

#include <limits.h>

#include <time.h>

typedef struct thread\_data {

    char \*word;

    char \*string;

    int start;

    int end;

    int found;

} thread\_data;

void \*naive\_search(void \*arg) {

    thread\_data \*data = (thread\_data \*)arg;

    int m = strlen(data->word);

    for (int i = data->start; i < data->end; i++) {

        int flag = 0;

        for (int j = 0; j < m; j++) {

            if (data->string[i + j] == data->word[j]) {

                flag = 1;

            }

            else {

                flag = 0;

                break;

            }

        }

        if (flag) {

            data->found = i;

            break;

        }

    }

    return NULL;

}

char \*get\_string() {

    int len = 0;

    int capacity = 1;

    char \*s = (char\*) malloc(sizeof(char));

    char c = getchar();

    while (c != '\n') {

        s[len++] = c;

        char \* for\_realloc;

        if (len >= capacity) {

            capacity \*= 2;

            for\_realloc = (char\*) realloc(s, capacity \* sizeof(char));

            if(!for\_realloc) {

                free(s);

                return NULL;

            }

            s = for\_realloc;

        }

        c = getchar();

    }

    s[len] = '\0';

    return s;

}

int main(int argc, char \*argv[]) {

    int threads\_num = strtol(argv[1], NULL, 10);

    pthread\_t threads[threads\_num];

    thread\_data threads\_data[threads\_num];

    size\_t size = 1000000000;

    char \*string = (char \*)malloc(size \* sizeof(char));

    if (string == NULL) {

        perror("Failed to allocate memory");

        return 1;

    }

    memset(string, 'A', size);

    int place = 82346;

    string[place + 3] = 'l'; string[place + 2] = 'a'; string[place + 1] = 'o'; string[place] = 'g';

    printf("Enter search word: ");

    char \*word = get\_string();

    int part\_size = strlen(string) / threads\_num;

    int part\_remains = strlen(string) % threads\_num;

    clock\_t start\_time = clock();

    for (int i = 0; i < threads\_num; i++) {

        threads\_data[i].string = string;

        threads\_data[i].word = word;

        threads\_data[i].found = INT\_MAX;

        if(part\_remains) {

            threads\_data[i].start = i \* part\_size + 1;

            part\_remains--;

        }

        else threads\_data[i].start = i \* part\_size;

        threads\_data[i].end = threads\_data[i].start + part\_size + strlen(word);

        if(pthread\_create(&threads[i], NULL, naive\_search, &threads\_data[i])) {

            free(string); free(word);

            return 1;

        }

    }

    int min\_ind = INT\_MAX;

    for(int i = 0; i < threads\_num; i++) {

        pthread\_join(threads[i], NULL);

        if (threads\_data[i].found < min\_ind)

            min\_ind = threads\_data[i].found;

    }

    clock\_t end\_time = clock();

    double delta\_time = ((double)(end\_time - start\_time)) / CLOCKS\_PER\_SEC;

    if (min\_ind != INT\_MAX)

        printf("Goal number is: %d\n", min\_ind);

    else

        printf("Word hasn't been found\n");

    printf("Working time with %d thread(s) is %f s.\n", threads\_num, delta\_time);

    free(word);

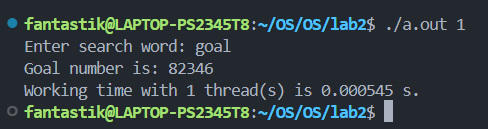
    free(string);

    return 0;

}

# Протокол работы программы

**Тест 1:**



## Strace:

strace ./a.out 1

execve("./a.out", ["./a.out", "1"], 0x7ffe30de1d18 /\* 37 vars \*/) = 0

brk(NULL) = 0x55a52ef77000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff4c63d0000

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

openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=19791, ...}) = 0

mmap(NULL, 19791, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7ff4c63cb000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"..., 832) = 832

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

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2125328, ...}) = 0

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

mmap(NULL, 2170256, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ff4c61b9000

mmap(0x7ff4c61e1000, 1605632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7ff4c61e1000

mmap(0x7ff4c6369000, 323584, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1b0000) = 0x7ff4c6369000

mmap(0x7ff4c63b8000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1fe000) = 0x7ff4c63b8000

mmap(0x7ff4c63be000, 52624, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ff4c63be000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff4c61b6000

arch\_prctl(ARCH\_SET\_FS, 0x7ff4c61b6740) = 0

set\_tid\_address(0x7ff4c61b6a10) = 323884

set\_robust\_list(0x7ff4c61b6a20, 24) = 0

rseq(0x7ff4c61b7060, 0x20, 0, 0x53053053) = 0

mprotect(0x7ff4c63b8000, 16384, PROT\_READ) = 0

mprotect(0x55a52e23e000, 4096, PROT\_READ) = 0

mprotect(0x7ff4c6408000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x7ff4c63cb000, 19791) = 0

getrandom("\xef\xc6\x74\x3b\xf1\xf0\x1f\x0c", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55a52ef77000

brk(0x55a52ef98000) = 0x55a52ef98000

mmap(NULL, 1000001536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ff48a809000

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x7), ...}) = 0

fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x7), ...}) = 0

write(1, "Enter search word: ", 19Enter search word: ) = 19

read(0, 0x55a52ef776d0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set)

--- SIGWINCH {si\_signo=SIGWINCH, si\_code=SI\_KERNEL} ---

read(0, goal

"goal\n", 1024) = 5

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=0, tv\_nsec=686568200}) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7ff4c6252520, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7ff4c61fe320}, NULL, 8) = 0

rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7ff48a008000

mprotect(0x7ff48a009000, 8388608, PROT\_READ|PROT\_WRITE) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7ff48a808990, parent\_tid=0x7ff48a808990, exit\_signal=0, stack=0x7ff48a008000, stack\_size=0x7fff80, tls=0x7ff48a8086c0} => {parent\_tid=[323957]}, 88) = 323957

rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

futex(0x7ff48a808990, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 323957, NULL, FUTEX\_BITSET\_MATCH\_ANY) = 0

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=0, tv\_nsec=687245400}) = 0

write(1, "Goal number is: 82346\n", 22Goal number is: 82346

) = 22

write(1, "Working time with 1 thread(s) is"..., 45Working time with 1 thread(s) is 0.000677 s.

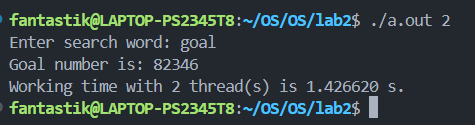
) = 45

munmap(0x7ff48a809000, 1000001536) = 0

exit\_group(0) = ?

+++ exited with 0 +++

**Тест 2:**



**Strace**:

strace ./a.out 2

execve("./a.out", ["./a.out", "2"], 0x7ffe193b07d8 /\* 37 vars \*/) = 0

brk(NULL) = 0x560bf3f3f000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ffa54350000

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

openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=19791, ...}) = 0

mmap(NULL, 19791, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7ffa5434b000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"..., 832) = 832

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

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2125328, ...}) = 0

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

mmap(NULL, 2170256, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7ffa54139000

mmap(0x7ffa54161000, 1605632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7ffa54161000

mmap(0x7ffa542e9000, 323584, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1b0000) = 0x7ffa542e9000

mmap(0x7ffa54338000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1fe000) = 0x7ffa54338000

mmap(0x7ffa5433e000, 52624, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7ffa5433e000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ffa54136000

arch\_prctl(ARCH\_SET\_FS, 0x7ffa54136740) = 0

set\_tid\_address(0x7ffa54136a10) = 324671

set\_robust\_list(0x7ffa54136a20, 24) = 0

rseq(0x7ffa54137060, 0x20, 0, 0x53053053) = 0

mprotect(0x7ffa54338000, 16384, PROT\_READ) = 0

mprotect(0x560bf39f5000, 4096, PROT\_READ) = 0

mprotect(0x7ffa54388000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x7ffa5434b000, 19791) = 0

getrandom("\xad\x78\xc5\xbd\x72\x80\x5d\x57", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x560bf3f3f000

brk(0x560bf3f60000) = 0x560bf3f60000

mmap(NULL, 1000001536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7ffa18789000

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x7), ...}) = 0

fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x7), ...}) = 0

write(1, "Enter search word: ", 19Enter search word: ) = 19

read(0, goal

"goal\n", 1024) = 5

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=0, tv\_nsec=486763800}) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7ffa541d2520, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7ffa5417e320}, NULL, 8) = 0

rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7ffa17f88000

mprotect(0x7ffa17f89000, 8388608, PROT\_READ|PROT\_WRITE) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7ffa18788990, parent\_tid=0x7ffa18788990, exit\_signal=0, stack=0x7ffa17f88000, stack\_size=0x7fff80, tls=0x7ffa187886c0} => {parent\_tid=[324720]}, 88) = 324720

rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7ffa17787000

mprotect(0x7ffa17788000, 8388608, PROT\_READ|PROT\_WRITE) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7ffa17f87990, parent\_tid=0x7ffa17f87990, exit\_signal=0, stack=0x7ffa17787000, stack\_size=0x7fff80, tls=0x7ffa17f876c0} => {parent\_tid=[324721]}, 88) = 324721

rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

futex(0x7ffa17f87990, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 324721, NULL, FUTEX\_BITSET\_MATCH\_ANY) = 0

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=1, tv\_nsec=883724200}) = 0

write(1, "Goal number is: 82346\n", 22Goal number is: 82346

) = 22

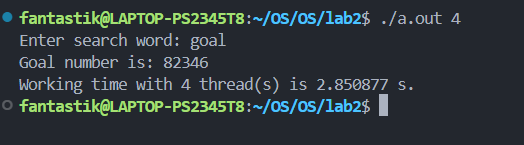
write(1, "Working time with 2 thread(s) is"..., 45Working time with 2 thread(s) is 1.396961 s.

) = 45

munmap(0x7ffa18789000, 1000001536) = 0

exit\_group(0) = ?

+++ exited with 0 +++

**Тест 3:  
  
  
  
Strace:**

strace ./a.out 4

execve("./a.out", ["./a.out", "4"], 0x7ffdce555c38 /\* 37 vars \*/) = 0

brk(NULL) = 0x55ceac51f000

mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fbd0b652000

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

openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3

fstat(3, {st\_mode=S\_IFREG|0644, st\_size=19791, ...}) = 0

mmap(NULL, 19791, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fbd0b64d000

close(3) = 0

openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0\0"..., 832) = 832

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

fstat(3, {st\_mode=S\_IFREG|0755, st\_size=2125328, ...}) = 0

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

mmap(NULL, 2170256, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fbd0b43b000

mmap(0x7fbd0b463000, 1605632, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x28000) = 0x7fbd0b463000

mmap(0x7fbd0b5eb000, 323584, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1b0000) = 0x7fbd0b5eb000

mmap(0x7fbd0b63a000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1fe000) = 0x7fbd0b63a000

mmap(0x7fbd0b640000, 52624, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fbd0b640000

close(3) = 0

mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fbd0b438000

arch\_prctl(ARCH\_SET\_FS, 0x7fbd0b438740) = 0

set\_tid\_address(0x7fbd0b438a10) = 326159

set\_robust\_list(0x7fbd0b438a20, 24) = 0

rseq(0x7fbd0b439060, 0x20, 0, 0x53053053) = 0

mprotect(0x7fbd0b63a000, 16384, PROT\_READ) = 0

mprotect(0x55ceab1d7000, 4096, PROT\_READ) = 0

mprotect(0x7fbd0b68a000, 8192, PROT\_READ) = 0

prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0

munmap(0x7fbd0b64d000, 19791) = 0

getrandom("\x25\x58\xd7\x5e\x56\xa0\xba\x4c", 8, GRND\_NONBLOCK) = 8

brk(NULL) = 0x55ceac51f000

brk(0x55ceac540000) = 0x55ceac540000

mmap(NULL, 1000001536, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fbccfa8b000

fstat(1, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x7), ...}) = 0

fstat(0, {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0x7), ...}) = 0

write(1, "Enter search word: ", 19Enter search word: ) = 19

read(0, goal

"goal\n", 1024) = 5

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=0, tv\_nsec=447976400}) = 0

rt\_sigaction(SIGRT\_1, {sa\_handler=0x7fbd0b4d4520, sa\_mask=[], sa\_flags=SA\_RESTORER|SA\_ONSTACK|SA\_RESTART|SA\_SIGINFO, sa\_restorer=0x7fbd0b480320}, NULL, 8) = 0

rt\_sigprocmask(SIG\_UNBLOCK, [RTMIN RT\_1], NULL, 8) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fbccf28a000

mprotect(0x7fbccf28b000, 8388608, PROT\_READ|PROT\_WRITE) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7fbccfa8a990, parent\_tid=0x7fbccfa8a990, exit\_signal=0, stack=0x7fbccf28a000, stack\_size=0x7fff80, tls=0x7fbccfa8a6c0} => {parent\_tid=[326202]}, 88) = 326202

rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fbccea89000

mprotect(0x7fbccea8a000, 8388608, PROT\_READ|PROT\_WRITE) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7fbccf289990, parent\_tid=0x7fbccf289990, exit\_signal=0, stack=0x7fbccea89000, stack\_size=0x7fff80, tls=0x7fbccf2896c0} => {parent\_tid=[326203]}, 88) = 326203

rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fbcce288000

mprotect(0x7fbcce289000, 8388608, PROT\_READ|PROT\_WRITE) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7fbccea88990, parent\_tid=0x7fbccea88990, exit\_signal=0, stack=0x7fbcce288000, stack\_size=0x7fff80, tls=0x7fbccea886c0} => {parent\_tid=[326204]}, 88) = 326204

rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

mmap(NULL, 8392704, PROT\_NONE, MAP\_PRIVATE|MAP\_ANONYMOUS|MAP\_STACK, -1, 0) = 0x7fbccda87000

mprotect(0x7fbccda88000, 8388608, PROT\_READ|PROT\_WRITE) = 0

rt\_sigprocmask(SIG\_BLOCK, ~[], [], 8) = 0

clone3({flags=CLONE\_VM|CLONE\_FS|CLONE\_FILES|CLONE\_SIGHAND|CLONE\_THREAD|CLONE\_SYSVSEM|CLONE\_SETTLS|CLONE\_PARENT\_SETTID|CLONE\_CHILD\_CLEARTID, child\_tid=0x7fbcce287990, parent\_tid=0x7fbcce287990, exit\_signal=0, stack=0x7fbccda87000, stack\_size=0x7fff80, tls=0x7fbcce2876c0} => {parent\_tid=[326205]}, 88) = 326205

rt\_sigprocmask(SIG\_SETMASK, [], NULL, 8) = 0

futex(0x7fbccf289990, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 326203, NULL, FUTEX\_BITSET\_MATCH\_ANY) = 0

futex(0x7fbccea88990, FUTEX\_WAIT\_BITSET|FUTEX\_CLOCK\_REALTIME, 326204, NULL, FUTEX\_BITSET\_MATCH\_ANY) = 0

clock\_gettime(CLOCK\_PROCESS\_CPUTIME\_ID, {tv\_sec=3, tv\_nsec=459417100}) = 0

write(1, "Goal number is: 82346\n", 22Goal number is: 82346

) = 22

write(1, "Working time with 4 thread(s) is"..., 45Working time with 4 thread(s) is 3.011441 s.

) = 45

munmap(0x7fbccfa8b000, 1000001536) = 0

exit\_group(0) = ?

+++ exited with 0 +++

# Вывод

В процессе выполнения лабораторной работы я научился управлять потоками в операционной системе. В рамках работы была создана и отлажена программа на языке Си, которая складывает массивы, используя потоки, работа с которыми производится через синхронизатор потоков pthread\_join() (барьер).