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: ЫсаевР. К.

Преподаватель: Бахарев В.Д. (ФИИТ)

Оценка: _____

Дата: 22.11.24

Москва, 2024

Постановка задачи

Вариант 15.

Есть колода из 52 карт, рассчитать экспериментально (метод Монте-Карло) вероятность того, что сверху лежат две одинаковых карты. Количество раундов задаётся ключом программы.

Общий метод и алгоритм решения

Использованные системные вызовы:

- sem_init(&thread_sem, 0, max_threads) — Инициализирует семафор с начальным значением max_threads, чтобы ограничить количество одновременно выполняющихся потоков.
- pthread_create(&threads[i], NULL, monte_carlo, &data) — Создаёт новый поток, который выполняет функцию monte_carlo, передавая в неё структуру данных data.
- pthread_mutex_lock(&success_mutex) — Блокирует мьютекс для синхронизации доступа к общей переменной successes между потоками.
- pthread_mutex_unlock(&success_mutex) — Разблокирует мьютекс после выполнения критической секции, чтобы другие потоки могли работать с общей переменной.
- write — Записывает данные в файловый дескриптор.
- sem_wait(&thread_sem) — Поток ждёт, пока не будет доступен разрешающий семафор для выполнения.
- sem_post(&thread_sem) — Освобождает семафор, позволяя следующему потоку продолжить выполнение.
- pthread_join(threads[i], NULL) — Ожидает завершения потока с индексом i, чтобы продолжить выполнение главного потока после завершения всех вычислений.
- sem_destroy(&thread_sem) — Удаляет семафор после завершения работы программы, освобождая ресурсы.

Создаем мьютекс для синхронизации доступа к переменной, которая отслеживает количество успешных раундов. Также инициализируем семафор для управления синхронизацией потоков. Далее запускаем несколько потоков, каждый из которых выполняет заданное количество раундов, переданное через аргументы командной строки. В каждом потоке мы создаём колоду карт, случайным образом ее перемешиваем и проверяем, совпадают ли верхние карты. Если карты совпадают, увеличиваем локальный счётчик успешных раундов.

Перед тем, как обновить общий счётчик успешных раундов, мы блокируем мьютекс. После обновления счётчика, мьютекс разблокируется, а затем увеличиваем семафор, сигнализируя, что раунд завершен. В конце программы вычисляется итоговая вероятность совпадения карт и выводится на экран.

В ниже приведенной таблице рассчитываем коэффициент ускорения, который равен S(p), где S(p) – это результат деления времени выполнения одного потока на время выполнения p потоков

Количество раундов	Количество потоков	Время выполнения	Коэффициент ускорения
100000	1	0,36с.	1
100000	4	0,10с.	3,6
100000	8	0,2с.	1,8

Код программы

task1.c

```
#include <pthread.h>
#include <unistd.h>
#include <semaphore.h>
#include <stdlib.h>

sem_t thread_sem;
#define DECK_SIZE 52
pthread_mutex_t success_mutex = PTHREAD_MUTEX_INITIALIZER;

typedef struct {
    int rounds;
    int successes;
} ThreadData;

void shuffle(int *deck) {
    for (int i = DECK_SIZE - 1; i > 0; i--) {
        int j = rand() & (i + 1);
        int tmp = deck[i];
        deck[i] = deck[j];
        deck[j] = tmp;
    }
```

```
}

void* monte_carlo(void* arg) {
    ThreadData *data = (ThreadData*)arg;
    int local_successes = 0;

    for (int i = 0; i < data->rounds; i++) {
        int deck[DECK_SIZE];
        for (int j = 0; j < DECK_SIZE; j++) {
            deck[j] = j / 4;
        }

        shuffle(deck);
        if (deck[0] == deck[1]) {
            local_successes++;
        }
    }

    if (pthread_mutex_lock(&success_mutex) != 0) {
        char msg[] = "Error lock mutex\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        pthread_exit((void*)1);
    }

    data->successes += local_successes;

    if (pthread_mutex_unlock(&success_mutex) != 0) {
        char msg[] = "Error unlock mutex\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
```

```
        pthread_exit((void *)1);
    }

    if (sem_post(&thread_sem) != 0) {
        char msg[] = "Error semaphore post\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
    }

    pthread_exit(NULL);
}

int my_atoi(const char *str, int *out) {
    int result = 0;
    int i = 0;
    if (str[i] == '\0' || str[i] == '-') {
        return -1;
    }

    for (; str[i] != '\0'; i++) {
        if (str[i] < '0' || str[i] > '9') {
            return -1;
        }
        result = result * 10 + (str[i] - '0');
    }
    if (result == 0) {
        return -1;
    }

    *out = result;
```

```
    return 0;
}

int probability_convert_str(char *str, double probability) {
    int int_part = (int)probability;
    int frac_part = (int)((probability - int_part) * 10000);

    int i = 0;
    str[i++] = '0' + int_part;
    str[i++] = '.';
    for (int j = 1000; j > 0; j /= 10) {
        str[i++] = '0' + (frac_part / j) % 10;
    }

    str[i++] = '\n';
    return i;
}

int main(int argc, char *argv[]) {
    if (argc != 3) {
        char msg[] = "Usage: ./main <rounds> <max_threads>\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return 1;
    }

    int rounds;
    int max_threads;
    if (my_atoi(argv[1], &rounds) != 0 || my_atoi(argv[2], &max_threads) != 0) {
        char msg[] = "Invalid input line arguments\n";
```

```
    write(STDERR_FILENO, msg, sizeof(msg) - 1);
    return 1;
}

if (sem_init(&thread_sem, 0, max_threads) != 0) {
    char msg[] = "Error init semaphore\n";
    write(STDERR_FILENO, msg, sizeof(msg) - 1);
    return 1;
}

pthread_t threads[max_threads];
ThreadData data = {rounds / max_threads, 0};

for (int i = 0; i < max_threads; i++) {
    if (sem_wait(&thread_sem) != 0) {
        char msg[] = "Error semaphore wait\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return 1;
    }

    if (pthread_create(&threads[i], NULL, monte_carlo, &data) != 0) {
        char msg[] = "Error thread creation\n";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return 1;
    }
}

for (int i = 0; i < max_threads; i++) {
    if (pthread_join(threads[i], NULL) != 0) {
```

```
            char msg[] = "Error waiting thread\n";
            write(STDERR_FILENO, msg, sizeof(msg) - 1);
            return 1;
        }
    }

    double probability = (double)data.successes / rounds;
    char buf[4096];
    int len = probability_convert_str(buf, probability);
    write(STDOUT_FILENO, buf, len);

    if (sem_destroy(&thread_sem) != 0) {
        char msg[] = "Error destroy semaphore";
        write(STDERR_FILENO, msg, sizeof(msg) - 1);
        return 1;
    }

    return 0;
}
```

Протокол работы программы

```
admin1@admin1-VMware-Virtual-Platform:~/OSY_LABS/Laba2$ strace -f ./task1 10 10
execve("./task1", ["./task1", "10", "10"], 0x7ffe36f6c668 /* 80 vars */) = 0
brk(NULL)                               = 0x5c50ce11e000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x74b1d8a19000
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=64511, ...}) = 0
mmap(NULL, 64511, PROT_READ, MAP_PRIVATE, 3, 0) = 0x74b1d8a09000
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) = 0x74b1d8600000

mmap(0x74b1d8628000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x74b1d8628000

mmap(0x74b1d87b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1b0000) = 0x74b1d87b0000

mmap(0x74b1d87ff000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x74b1d87ff000

mmap(0x74b1d8805000, 52624, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x74b1d8805000

close(3)   = 0

mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x74b1d8a06000

arch_prctl(ARCH_SET_FS, 0x74b1d8a06740) = 0

set_tid_address(0x74b1d8a06a10)   = 8933

set_robust_list(0x74b1d8a06a20, 24)   = 0

rseq(0x74b1d8a07060, 0x20, 0, 0x53053053) = 0

mprotect(0x74b1d87ff000, 16384, PROT_READ) = 0

mprotect(0x5c50cdf6b000, 4096, PROT_READ) = 0

mprotect(0x74b1d8a51000, 8192, PROT_READ) = 0

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

munmap(0x74b1d8a09000, 64511)   = 0

rt_sigaction(SIGRT_1, {sa_handler=0x74b1d8699520, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x74b1d8645320}, 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) =
0x74b1d7c00000

mprotect(0x74b1d7c01000, 8388608, PROT_READ|PROT_WRITE) = 0

getrandom("\x9d\x91\x30\xd4\x0a\x9a\x89\x21", 8, GRND_NONBLOCK) = 8

brk(NULL)   = 0x5c50ce11e000

brk(0x5c50ce13f000)   = 0x5c50ce13f000

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=0x74b1d8400990, parent_tid=0x74b1d8400990, exit_signal=0, stack=0x74b1d7c00000, stack_size=0x7fff80, tls=0x74b1d84006c0}strace: Process 8934 attached

 => {parent_tid=[8934]}, 88) = 8934

[pid 8934] rseq(0x74b1d8400fe0, 0x20, 0, 0x53053053) = 0

[pid 8934] set_robust_list(0x74b1d84009a0, 24 <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

[pid 8934] <... set_robust_list resumed>) = 0

[pid 8933] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>

[pid 8934] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8933] <... mmap resumed>)   = 0x74b1d7200000

[pid 8934] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] mprotect(0x74b1d7201000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 8934] openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC <unfinished ...>

[pid 8933] <... mprotect resumed>)   = 0

[pid 8934] <... openat resumed>)   = 3

[pid 8933] futex(0x74b1d8a53a58, FUTEX_WAIT_PRIVATE, 2, NULL <unfinished ...>

[pid 8934] fstat(3, {st_mode=S_IFREG|0644, st_size=64511, ...}) = 0

[pid 8934] mmap(NULL, 64511, PROT_READ, MAP_PRIVATE, 3, 0) = 0x74b1d8a09000

[pid 8934] close(3)   = 0

[pid 8934] mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x74b1cf200000

[pid 8934] munmap(0x74b1cf200000, 14680064) = 0

[pid 8934] munmap(0x74b1d4000000, 52428800) = 0

[pid 8934] mprotect(0x74b1d0000000, 135168, PROT_READ|PROT_WRITE) = 0

[pid 8934] openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3

[pid 8934] read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"..., 832) = 832

[pid 8934] fstat(3, {st_mode=S_IFREG|0644, st_size=183024, ...}) = 0

[pid 8934] mmap(NULL, 185256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x74b1d89d8000

[pid 8934] mmap(0x74b1d89dc000, 147456, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x4000) = 0x74b1d89dc000

[pid 8934] mmap(0x74b1d8a00000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x74b1d8a00000

[pid 8934] mmap(0x74b1d8a04000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2b000) = 0x74b1d8a04000

[pid 8934] close(3) = 0

[pid 8934] mprotect(0x74b1d8a04000, 4096, PROT_READ) = 0

[pid 8934] futex(0x74b1d8a53a58, FUTEX_WAKE_PRIVATE, 1) = 1

[pid 8933] <... futex resumed>) = 0

[pid 8934] munmap(0x74b1d8a09000, 64511 <unfinished ...>

[pid 8933] futex(0x74b1d8a53a58, FUTEX_WAKE_PRIVATE, 1 <unfinished ...>

[pid 8934] <... munmap resumed>) = 0

[pid 8933] <... futex resumed>) = 0

[pid 8934] futex(0x74b1d8a05230, FUTEX_WAKE_PRIVATE, 2147483647 <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>

[pid 8934] <... futex resumed>) = 0

[pid 8933] <... rt_sigprocmask resumed>[], 8) = 0

[pid 8934] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 8933] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x74b1d7a00990, parent_tid=0x74b1d7a00990, exit_signal=0, stack=0x74b1d7200000, stack_size=0x7fff80, tls=0x74b1d7a006c0} <unfinished ...>

[pid 8934] <... rt_sigprocmask resumed>NULL, 8) = 0

strace: Process 8935 attached

[pid 8933] <... clone3 resumed> => {parent_tid=[8935]}, 88) = 8935

[pid 8934] madvise(0x74b1d7c00000, 8368128, MADV_DONTNEED <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8935] rseq(0x74b1d7a00fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 8934] <... madvise resumed>) = 0

[pid 8933] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8935] <... rseq resumed>) = 0

[pid 8935] set_robust_list(0x74b1d7a009a0, 24 <unfinished ...>

[pid 8934] exit(0 <unfinished ...>

[pid 8933] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>

[pid 8935] <... set_robust_list resumed>) = 0

[pid 8934] <... exit resumed>)   = ?

[pid 8933] <... mmap resumed>)   = 0x74b1d6800000

[pid 8935] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8934] +++ exited with 0 +++

[pid 8933] mprotect(0x74b1d6801000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 8935] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] <... mprotect resumed>)   = 0

[pid 8935] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>

[pid 8935] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] <... rt_sigprocmask resumed>[], 8) = 0

[pid 8935] madvise(0x74b1d7200000, 8368128, MADV_DONTNEED <unfinished ...>

[pid 8933] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x74b1d7000990, parent_tid=0x74b1d7000990, exit_signal=0, stack=0x74b1d6800000, stack_size=0x7fff80, tls=0x74b1d70006c0} <unfinished ...>

[pid 8935] <... madvise resumed>)   = 0

strace: Process 8936 attached

[pid 8933] <... clone3 resumed> => {parent_tid=[8936]}, 88) = 8936

[pid 8935] exit(0)   = ?

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8936] rseq(0x74b1d7000fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 8935] +++ exited with 0 +++

[pid 8933] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8936] <... rseq resumed>)   = 0

[pid 8933] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>

[pid 8936] set_robust_list(0x74b1d70009a0, 24 <unfinished ...>

[pid 8933] <... mmap resumed>)   = 0x74b1d5e00000

[pid 8936] <... set_robust_list resumed>) = 0

[pid 8933] mprotect(0x74b1d5e01000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 8936] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8933] <... mprotect resumed>)   = 0

[pid 8936] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 8936] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 8933]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYS
VSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x74b1d6600990, parent_tid=0x74b1d6600990, exit_signal=0, stack=0x74b1d5e00000,
stack_size=0x7fff80, tls=0x74b1d66006c0} <unfinished ...>

[pid 8936] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8936] madvise(0x74b1d6800000, 8368128, MADV_DONTNEEDstrace: Process 8937 attached

 <unfinished ...>

[pid 8933] <... clone3 resumed> => {parent_tid=[8937]}, 88) = 8937

[pid 8936] <... madvise resumed>)   = 0

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

[pid 8937] rseq(0x74b1d6600fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 8936] exit(0 <unfinished ...>

[pid 8933] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0
<unfinished ...>

[pid 8937] <... rseq resumed>)   = 0

[pid 8936] <... exit resumed>)   = ?

[pid 8936] +++ exited with 0 +++

[pid 8933] <... mmap resumed>)   = 0x74b1d5400000

[pid 8937] set_robust_list(0x74b1d66009a0, 24 <unfinished ...>

[pid 8933] mprotect(0x74b1d5401000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 8937] <... set_robust_list resumed>) = 0

[pid 8933] <... mprotect resumed>)   = 0

[pid 8937] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>

[pid 8937] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] <... rt_sigprocmask resumed>[], 8) = 0

[pid 8937] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 8933] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x74b1d5c00990, parent_tid=0x74b1d5c00990, exit_signal=0, stack=0x74b1d5400000, stack_size=0x7fff80, tls=0x74b1d5c006c0} <unfinished ...>

[pid 8937] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8937] madvise(0x74b1d5e00000, 8368128, MADV_DONTNEED) = 0

[pid 8933] <... clone3 resumed> => {parent_tid=[8938]}, 88) = 8938

strace: Process 8938 attached

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8937] exit(0 <unfinished ...>

[pid 8933] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8938] rseq(0x74b1d5c00fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 8937] <... exit resumed>)   = ?

[pid 8933] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>

[pid 8938] <... rseq resumed>)   = 0

[pid 8937] +++ exited with 0 +++

[pid 8933] <... mmap resumed>)   = 0x74b1d4a00000

[pid 8938] set_robust_list(0x74b1d5c009a0, 24 <unfinished ...>

[pid 8933] mprotect(0x74b1d4a01000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 8938] <... set_robust_list resumed>) = 0

[pid 8933] <... mprotect resumed>)   = 0

[pid 8938] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 8938] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x74b1d5200990, parent_tid=0x74b1d5200990, exit_signal=0, stack=0x74b1d4a00000, stack_size=0x7fff80, tls=0x74b1d52006c0} <unfinished ...>

[pid 8938] rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

strace: Process 8939 attached

[pid 8933] <... clone3 resumed> => {parent_tid=[8939]}, 88) = 8939

[pid 8939] rseq(0x74b1d5200fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 8938] madvise(0x74b1d5400000, 8368128, MADV_DONTNEED <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8939] <... rseq resumed>)   = 0

[pid 8933] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8939] set_robust_list(0x74b1d52009a0, 24 <unfinished ...>

[pid 8938] <... madvise resumed>)   = 0

[pid 8933] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0)
= 0x74b1cf600000

[pid 8939] <... set_robust_list resumed>) = 0

[pid 8938] exit(0 <unfinished ...>

[pid 8933] mprotect(0x74b1cf601000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 8939] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8938] <... exit resumed>)   = ?

[pid 8933] <... mprotect resumed>)   = 0

[pid 8939] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8938] +++ exited with 0 +++

[pid 8933] rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

[pid 8939] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 8933]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYS
VSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x74b1cfe00990, parent_tid=0x74b1cfe00990, exit_signal=0, stack=0x74b1cf600000,
stack_size=0x7fff80, tls=0x74b1cfe006c0} <unfinished ...>

[pid 8939] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8939] madvise(0x74b1d4a00000, 8368128, MADV_DONTNEEDstrace: Process 8940 attached

 <unfinished ...>

[pid 8933] <... clone3 resumed> => {parent_tid=[8940]}, 88) = 8940

[pid 8940] rseq(0x74b1cfe00fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8940] <... rseq resumed>)   = 0

[pid 8939] <... madvise resumed>)   = 0

[pid 8933] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>

[pid 8939] exit(0 <unfinished ...>

[pid 8940] set_robust_list(0x74b1cfe009a0, 24 <unfinished ...>

[pid 8933] <... mmap resumed>)   = 0x74b1cec00000

[pid 8939] <... exit resumed>)   = ?

[pid 8933] mprotect(0x74b1cec01000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 8940] <... set_robust_list resumed>) = 0

[pid 8939] +++ exited with 0 +++

[pid 8933] <... mprotect resumed>)   = 0

[pid 8940] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>

[pid 8940] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] <... rt_sigprocmask resumed>[], 8) = 0

[pid 8933] clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x74b1cf400990, parent_tid=0x74b1cf400990, exit_signal=0, stack=0x74b1cec00000, stack_size=0x7fff80, tls=0x74b1cf4006c0} <unfinished ...>

[pid 8940] rt_sigprocmask(SIG_BLOCK, ~[RT_1], strace: Process 8941 attached

 <unfinished ...>

[pid 8933] <... clone3 resumed> => {parent_tid=[8941]}, 88) = 8941

[pid 8940] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8941] rseq(0x74b1cf400fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8941] <... rseq resumed>)   = 0

[pid 8933] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8941] set_robust_list(0x74b1cf4009a0, 24 <unfinished ...>

[pid 8940] madvise(0x74b1cf600000, 8368128, MADV_DONTNEED <unfinished ...>

[pid 8933] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>

[pid 8941] <... set_robust_list resumed>) = 0

[pid 8940] <... madvise resumed>)   = 0

[pid 8933] <... mmap resumed>)   = 0x74b1ce200000

[pid 8941] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8933] mprotect(0x74b1ce201000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 8940] exit(0 <unfinished ...>

[pid 8933] <... mprotect resumed>)   = 0

[pid 8941] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8940] <... exit resumed>)   = ?

[pid 8933] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>

[pid 8941] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 8940] +++ exited with 0 +++

[pid 8933] <... rt_sigprocmask resumed>[], 8) = 0

[pid 8941] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYS
VSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x74b1cea00990, parent_tid=0x74b1cea00990, exit_signal=0, stack=0x74b1ce200000,
stack_size=0x7fff80, tls=0x74b1cea006c0} <unfinished ...>

[pid 8941] madvise(0x74b1cec00000, 8368128, MADV_DONTNEEDstrace: Process 8942 attached

) = 0

[pid 8933] <... clone3 resumed> => {parent_tid=[8942]}, 88) = 8942

[pid 8942] rseq(0x74b1cea00fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 8941] exit(0 <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8941] <... exit resumed>)   = ?

[pid 8942] <... rseq resumed>)   = 0

[pid 8933] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8942] set_robust_list(0x74b1cea009a0, 24 <unfinished ...>

[pid 8933] mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0)
= 0x74b1cd800000

[pid 8942] <... set_robust_list resumed>) = 0

[pid 8941] +++ exited with 0 +++

[pid 8933] mprotect(0x74b1cd801000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

[pid 8942] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8933] <... mprotect resumed>)   = 0

[pid 8942] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] rt_sigprocmask(SIG_BLOCK, ~[], <unfinished ...>

[pid 8942] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 8933] <... rt_sigprocmask resumed>[], 8) = 0

[pid 8942] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933]
clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYS
VSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x74b1ce000990, parent_tid=0x74b1ce000990, exit_signal=0, stack=0x74b1cd800000,
stack_size=0x7fff80, tls=0x74b1ce0006c0} => {parent_tid=[8943]}, 88) = 8943

[pid 8942] madvise(0x74b1ce200000, 8368128, MADV_DONTNEED <unfinished ...>

[pid 8933] rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

[pid 8942] <... madvise resumed>)   = 0

[pid 8933] munmap(0x74b1d7c00000, 8392704) = 0

[pid 8942] exit(0strace: Process 8943 attached

 <unfinished ...>

[pid 8933] munmap(0x74b1d7200000, 8392704 <unfinished ...>

[pid 8942] <... exit resumed>)   = ?

[pid 8943] rseq(0x74b1ce000fe0, 0x20, 0, 0x53053053 <unfinished ...>

[pid 8942] +++ exited with 0 +++

[pid 8933] <... munmap resumed>)   = 0

[pid 8943] <... rseq resumed>)   = 0

[pid 8933] munmap(0x74b1d6800000, 8392704 <unfinished ...>

[pid 8943] set_robust_list(0x74b1ce0009a0, 24 <unfinished ...>

[pid 8933] <... munmap resumed>)   = 0

[pid 8943] <... set_robust_list resumed>) = 0

[pid 8933] munmap(0x74b1d5e00000, 8392704 <unfinished ...>

[pid 8943] rt_sigprocmask(SIG_SETMASK, [], <unfinished ...>

[pid 8933] <... munmap resumed>)   = 0

[pid 8943] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] munmap(0x74b1d5400000, 8392704 <unfinished ...>

[pid 8943] rt_sigprocmask(SIG_BLOCK, ~[RT_1], <unfinished ...>

[pid 8933] <... munmap resumed>)   = 0

[pid 8943] <... rt_sigprocmask resumed>NULL, 8) = 0

[pid 8933] futex(0x74b1ce000990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 8943, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

[pid 8943] madvise(0x74b1cd800000, 8368128, MADV_DONTNEED) = 0

[pid 8943] exit(0)   = ?

[pid 8933] <... futex resumed>)   = 0

[pid 8943] +++ exited with 0 +++

munmap(0x74b1d4a00000, 8392704)   = 0

write(1, "0.2000\n", 70.2000

)   = 7

exit_group(0)   = ?

+++ exited with 0 +++

Вывод

В ходе лабораторной работы были изучены возможности библиотек языка C в создании многопоточных приложений, также были изучены инструменты для работы с потоками.