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: Тропн М.А.

Преподаватель: Бахарев В.Д.

Оценка: ____________________

Дата: 31.10.24

Москва, 2024

Постановка задачи

Вариант 15.

Цель работы

Приобретение практических навыков в:

- Управление потоками в ОС
- Обеспечение синхронизации между потоками

Задание

Составить программу на языке Си(C++), 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.

15 вариант) 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:

- int pthread_create(pthread_t *__restrict__ __newthread, const pthread_attr_t *__restrict__ __attr, void *(*__start_routine)(void *), void *__restrict__ __arg) — создаёт поток с рутиной (стартовой функцией) и заданными аргументами
- int pthread_join(pthread_t __th, void **__thread_return) — вызов потока
- ssize_t read(int __fd, void* __buf, size_t __nbytes) — чтение из fd в буфер
- ssize_t write(int __fd, const void* __buf, size_t __n) — запись байтов в буфер
- void exit(int status) - завершения выполнения процесса и возвращение статуса

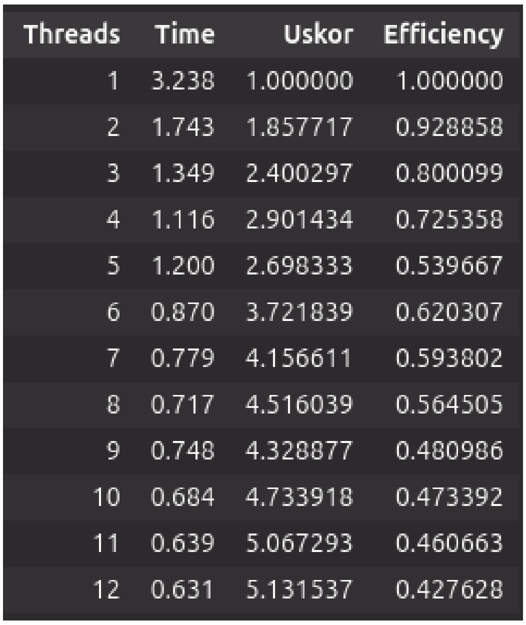
Также используется для синхронизации данных:
std::atomic — атомарный тип данных вместо std::mutex
pthread_mutex_t — тип данных mutex

- pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER; - для инициализации мьютекса (либо можно использовать:
  int pthread_mutex_init(pthread_mutex_t *__mutex, const pthread_mutexattr_t *__mutexattr) noexcept(true) — создание мьютекса)
- int pthread_mutex_lock(pthread_mutex_t *__mutex) noexcept(true) — блокировка мьютекса
- int pthread_mutex_unlock(pthread_mutex_t *__mutex) noexcept(true) — разблокирование мьютекса
- int pthread_mutex_destroy(pthread_mutex_t *__mutex) noexcept(true) — удаление мьютекса

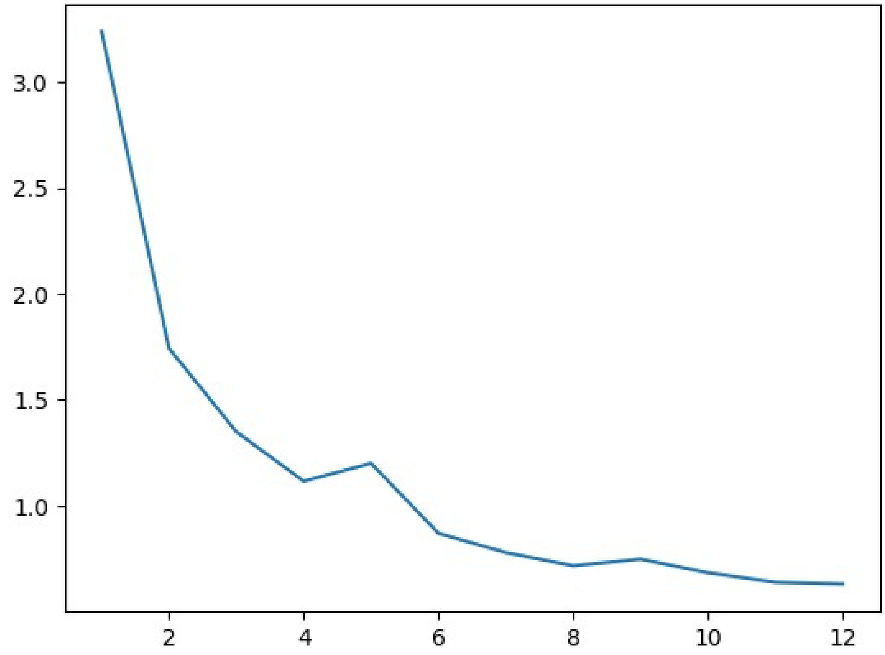
Создаём потоки (кол-во указывается ключом программы). Аргументом функции pthread_create будет являться функция just_do с аргументом rounds (рассчитывается как общее кол-во раундов (является второым ключом программы), разделенное на кол-во потоков). Таким образом, каждый поток будет рассчитывать приблизительно одинаковое количество раундов. Также введем атомарный счетчик ac (атомарность обязательна, в противном случае между потоками происходит т. н. «data race», из-за чего кол-во счетчика не совпадает с действительностью (потоки борятся за получение доступа к счетчика. Использование мьютекса в задаче излишне, ведь увеличаться затраты на ображение к Kernel).

Сам метод Монте-Карло заключается в оперировании случайностями. По заданию необходимо найти вероятность того, что первые две карты в колоде (52 карты) будут одинаковой масти можно расчитать проведя эксперимент — взять колоду, перемешать её и проверить что верхние карты одинаковы. Колоду будем перемешивать стандартным алгоритмом C++ std::shuffle. В таком удачном случае счетчик ac будем атомарно увеличивать. В итоге вероятность будем рассчитывать как результат деления количества удачный исходов на общее количество раундов.

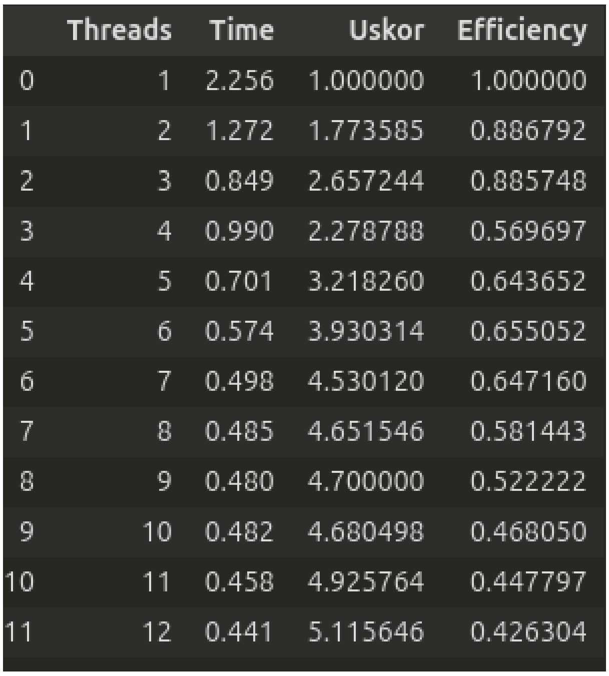
При N=1000000. Time — в секундах. Использование std::atomic_int



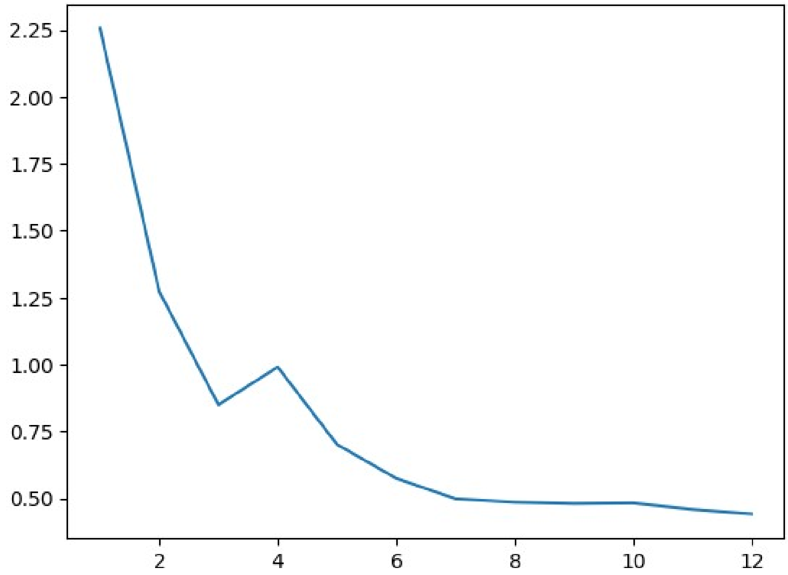
Threads	Time	Uskor	Efficiency
1	3.238	1.000000	1.000000
2	1.743	1.857717	0.928858
3	1.349	2.400297	0.800099
4	1.116	2.901434	0.725358
5	1.200	2.698333	0.539667
6	0.870	3.721839	0.620307
7	0.779	4.156611	0.593802
8	0.717	4.516039	0.564505
9	0.748	4.328877	0.480986
10	0.684	4.733918	0.473392
11	0.639	5.067293	0.460663
12	0.631	5.131537	0.427628



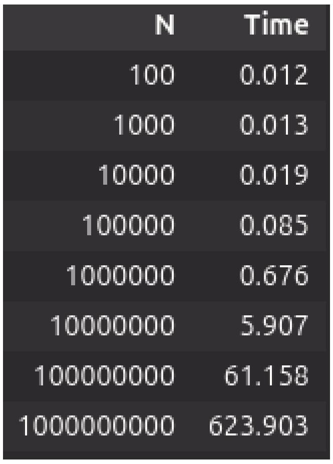
При N=1000000. Time — в секундах. Использование pthread_mutex_t



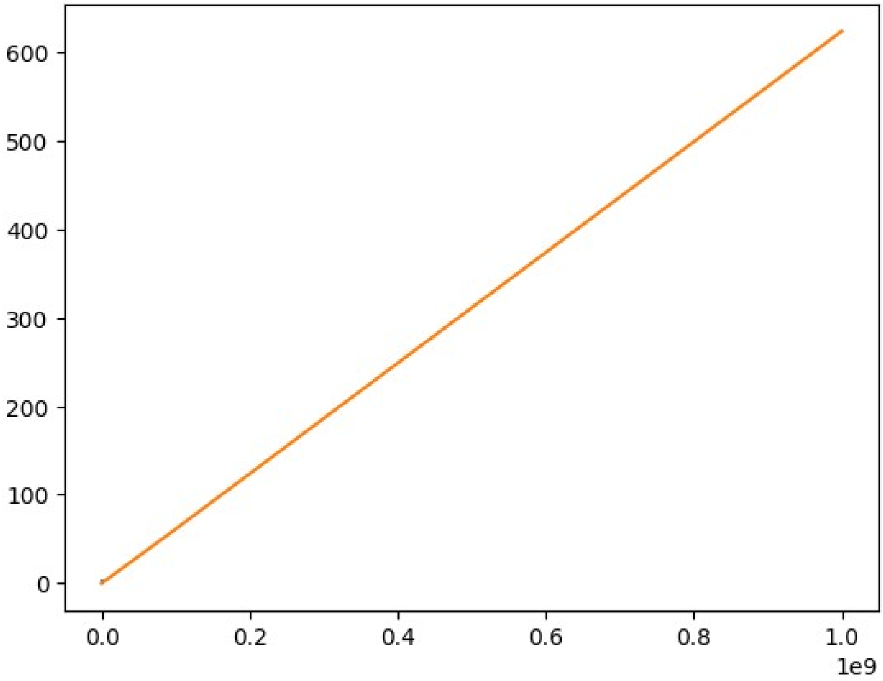
	Threads	Time	Uskor	Efficiency
0	1	2.256	1.000000	1.000000
1	2	1.272	1.773585	0.886792
2	3	0.849	2.657244	0.885748
3	4	0.990	2.278788	0.569697
4	5	0.701	3.218260	0.643652
5	6	0.574	3.930314	0.655052
6	7	0.498	4.530120	0.647160
7	8	0.485	4.651546	0.581443
8	9	0.480	4.700000	0.522222
9	10	0.482	4.680498	0.468050
10	11	0.458	4.925764	0.447797
11	12	0.441	5.115646	0.426304



При threads=12. Time — в секундах



N	Time
100	0.012
1000	0.013
10000	0.019
100000	0.085
1000000	0.676
10000000	5.907
100000000	61.158
1000000000	623.903



Код программы

main.cpp

```
#include <algorithm>

#include <atomic>

#include <pthread.h>
```

```
#include <random>

#include <stdarg.h>

#include <string.h>

#include <unistd.h>

#include <vector>


#define all(x) x.begin(), x.end()

#define DECK_NUM 52


std::atomic_int ac{0};


void *just_do(void *args) {

  size_t round = *(size_t *)args;

  std::random_device rd;

  std::mt19937 random_generator(rd());

  int local = 0;

  std::vector<int> deck(DECK_NUM);

  for (size_t j = 0; j < DECK_NUM; j++) {

    deck[j] = j;

  }

  for (size_t i = 0; i < round; i++) {

    std::shuffle(all(deck), random_generator);

    if (deck[0] % 4 == deck[1] % 4) {

      ++local;

    }

  }

  ac += local;


  return NULL;

}


void _print(char *fmt, ...) {

  va_list args;
  va_start(args, fmt);
  char BUF[BUFSIZ];
  vsprintf(BUF, fmt, args);
  if (-1 == write(STDOUT_FILENO, BUF, strlen(BUF))) {
    exit(EXIT_FAILURE);
  }
  va_end(args);
}

int main(int argc, char **argv) {
  if (argc != 3) {
    _print("Input error. Use: %s <threads> <rounds>\n", argv[0]);
    exit(EXIT_FAILURE);
  }
  if (argv[1] == NULL || argv[2] == NULL) {
    _print("Input error. Use: %s <threads> <rounds>\n", argv[0]);
    exit(EXIT_FAILURE);
  }
  // ограничение потоков и количество раундов
  size_t threads_num = atol(argv[1]);
  size_t rounds = atol(argv[2]);
  size_t rounds_for_thread = rounds / threads_num;
  if (rounds % threads_num != 0) {
    threads_num++;
  }
  rounds_for_thread = rounds / threads_num;
  std::vector<pthread_t> threads(threads_num);
  for (size_t i = 0; i < threads_num; i++) {
    if (-1 == pthread_create(&(threads[i]), NULL, just_do,
                             (void *)&rounds_for_thread)) {
      _print("Error. Thread %d nor created\n", i);
      exit(EXIT_FAILURE);
    };
  }

  for (auto &thread : threads) {
    if (-1 == pthread_join(thread, NULL)) {
      _print("Error. Thread not joined\n");
      exit(EXIT_FAILURE);
    };
  }

  // выводилка удачный попаданий
  {
    int b = ac.load();
    _print("== %d ==\n", b);
  }

  {
    // расчеты
    int total_rounds = rounds; // threads_num * rounds;
    int f = ac.load();
    double probability = (double)f / (double)total_rounds;
    char BUF[BUFSIZ];
    sprintf(BUF, "== %f ==\n", probability);
    _print("%s\n", BUF);
  }

  return EXIT_SUCCESS;
}
```

main_mutex.cpp (реализация на mutex):

```
#include <algorithm>
#include <array>
#include <iostream>
#include <mutex>
```

```
#include <pthread.h>
#include <random>
#include <stdarg.h>
#include <string.h>
#include <unistd.h>

#define all(x) x.begin(), x.end()
#define DECK_NUM 52

int ac = 0;
pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER;

void *just_do(void *args) {
  size_t round = *(size_t *)args;
  std::random_device rd;
  std::mt19937 random_generator(rd());
  int local = 0;
  std::array<int, DECK_NUM> deck;
  for (size_t j = 0; j < DECK_NUM; j++) {
    deck[j] = j;
  }
  for (size_t i = 0; i < round; i++) {
    std::shuffle(all(deck), random_generator);
    if (deck[0] % 4 == deck[1] % 4) {
      ++local;
    }
  }
  pthread_mutex_lock(&m);
  ac += local;
  pthread_mutex_unlock(&m);

  return NULL;
}
```

```
void _print(char *fmt, ...) {
  va_list args;
  va_start(args, fmt);
  char BUF[BUFSIZ];
  vsprintf(BUF, fmt, args);
  if (-1 == write(STDOUT_FILENO, BUF, strlen(BUF))) {
    exit(EXIT_FAILURE);
  }
  va_end(args);
}

int main(int argc, char **argv) {
  if (argc != 3) {
    _print("Input error. Use: %s <threads> <rounds>\n", argv[0]);
    exit(EXIT_FAILURE);
  }
  if (argv[1] == NULL || argv[2] == NULL) {
    _print("Input error. Use: %s <threads> <rounds>\n", argv[0]);
    exit(EXIT_FAILURE);
  }
  // ограничение потоков и количество раундов
  size_t threads_num = atol(argv[1]);
  size_t rounds = atol(argv[2]);
  size_t rounds_for_thread = rounds / threads_num;
  if (rounds % threads_num != 0) {
    threads_num++;
  }
  rounds_for_thread = rounds / threads_num;
  std::vector<pthread_t> threads(threads_num);
  for (size_t i = 0; i < threads_num; i++) {
    if (-1 == pthread_create(&(threads[i]), NULL, just_do,
                             (void *)&rounds_for_thread)) {
```

```
        _print("Error. Thread %d nor created\n", i);
        exit(EXIT_FAILURE);
      };
    }

    for (auto &thread : threads) {
      if (-1 == pthread_join(thread, NULL)) {
        _print("Error. Thread not joined\n");
        exit(EXIT_FAILURE);
      };
    }

    {
      // расчеты
      int total_rounds = rounds; // threads_num * rounds;
      _print("== %d ==\n", ac);
      double probability = (double)ac / (double)total_rounds;
      char BUF[BUFSIZ];
      sprintf(BUF, "== %f ==\n", probability);
      _print("%s\n", BUF);
    }
    pthread_mutex_destroy(&m);
    return EXIT_SUCCESS;
  }
```

Протокол работы программы

Тестирование:

```
lemito@lemito:~/Desktop/OSi/lab2/out/build/gcc$
 /home/lemito/Desktop/OSi/lab2/out/build/gcc/lab2 12 1000000
== 234392 ==
== 0.234392 ==
lemito@lemito:~/Desktop/OSi/lab2/out/build/gcc$ time
/home/lemito/Desktop/OSi/lab2/out/build/gcc/lab2 12 1000000
```

```
     == 235216 ==

     == 0.235216 ==



     real    0m0.631s

     user    0m6.150s

     sys     0m0.010s

     lemito@lemito:~/Desktop/OSi/lab2/out/build/gcc$ time
/home/lemito/Desktop/OSi/lab2/out/build/gcc/lab2 6 1000000

     == 235308 ==

     == 0.235308 ==



     real    0m0.870s

     user    0m4.887s

     sys     0m0.006s

     lemito@lemito:~/Desktop/OSi/lab2/out/build/gcc$ time
/home/lemito/Desktop/OSi/lab2/out/build/gcc/lab2 3 1000000

     == 235482 ==

     == 0.235482 ==



     real    0m1.349s

     user    0m3.976s

     sys     0m0.003s

     lemito@lemito:~/Desktop/OSi/lab2/out/build/gcc$ time
/home/lemito/Desktop/OSi/lab2/out/build/gcc/lab2 1 1000000

     == 235372 ==

     == 0.235372 ==



     real    0m3.238s

     user    0m3.230s

     sys     0m0.007s
     lemito@lemito:~/Desktop/OSi/lab2/out/build/gcc$ time
/home/lemito/Desktop/OSi/lab2/out/build/gcc/lab2 12 1000

     == 254 ==

     == 0.254000 ==



     real    0m0.012s

     user    0m0.010s

     sys     0m0.007s
```

Strace:

```
     175226 execve("/home/lemito/Desktop/OSi/lab2/out/build/gcc/lab2",
["/home/lemito/Desktop/OSi/lab2/ou"..., "12", "1000000"], 0x7fffb4e1e668 /* 118 vars
*/) = 0

     175226 brk(NULL)                       = 0x6236481d5000

     175226 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x7ec2ec769000

     175226 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or
directory)

     175226 openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3

     175226 fstat(3, {st_mode=S_IFREG|0644, st_size=98135, ...}) = 0

     175226 mmap(NULL, 98135, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7ec2ec751000

     175226 close(3)                        = 0

     175226 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libstdc++.so.6", O_RDONLY|
O_CLOEXEC) = 3

     175226 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\0"..., 832) = 832

     175226 fstat(3, {st_mode=S_IFREG|0644, st_size=2592224, ...}) = 0

     175226 mmap(NULL, 2609472, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7ec2ec400000

     175226 mmap(0x7ec2ec49d000, 1343488, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x9d000) = 0x7ec2ec49d000

     175226 mmap(0x7ec2ec5e5000, 552960, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1e5000) = 0x7ec2ec5e5000

     175226 mmap(0x7ec2ec66c000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x26b000) = 0x7ec2ec66c000

     175226 mmap(0x7ec2ec67a000, 12608, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7ec2ec67a000
```

```
     175226 close(3)                         = 0

     175226 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", O_RDONLY|
O_CLOEXEC) = 3

     175226 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

     175226 fstat(3, {st_mode=S_IFREG|0644, st_size=183024, ...}) = 0

     175226 mmap(NULL, 185256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7ec2ec723000

     175226 mmap(0x7ec2ec727000, 147456, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x4000) = 0x7ec2ec727000

     175226 mmap(0x7ec2ec74b000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x28000) = 0x7ec2ec74b000

     175226 mmap(0x7ec2ec74f000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2b000) = 0x7ec2ec74f000

     175226 close(3)                         = 0

     175226 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC)
= 3

     175226 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

     175226 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

     175226 fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0

     175226 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

     175226 mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7ec2ec000000

     175226 mmap(0x7ec2ec028000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|
MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7ec2ec028000

     175226 mmap(0x7ec2ec1b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1b0000) = 0x7ec2ec1b0000

     175226 mmap(0x7ec2ec1ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x1fe000) = 0x7ec2ec1ff000

     175226 mmap(0x7ec2ec205000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7ec2ec205000

     175226 close(3)                         = 0

     175226 openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.so.6", O_RDONLY|O_CLOEXEC)
= 3

     175226 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\0"..., 832) = 832

     175226 fstat(3, {st_mode=S_IFREG|0644, st_size=952616, ...}) = 0
     175226 mmap(NULL, 950296, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7ec2ec317000

     175226 mmap(0x7ec2ec327000, 520192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x10000) = 0x7ec2ec327000

     175226 mmap(0x7ec2ec3a6000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x8f000) = 0x7ec2ec3a6000

     175226 mmap(0x7ec2ec3fe000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0xe7000) = 0x7ec2ec3fe000

     175226 close(3)                         = 0

     175226 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
= 0x7ec2ec721000

     175226 mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7ec2ec71e000

     175226 arch_prctl(ARCH_SET_FS, 0x7ec2ec71e740) = 0

     175226 set_tid_address(0x7ec2ec71ea10)  = 175226

     175226 set_robust_list(0x7ec2ec71ea20, 24) = 0

     175226 rseq(0x7ec2ec71f060, 0x20, 0, 0x53053053) = 0

     175226 mprotect(0x7ec2ec1ff000, 16384, PROT_READ) = 0

     175226 mprotect(0x7ec2ec3fe000, 4096, PROT_READ) = 0

     175226 mprotect(0x7ec2ec74f000, 4096, PROT_READ) = 0

     175226 mprotect(0x7ec2ec66c000, 45056, PROT_READ) = 0

     175226 mprotect(0x62364770b000, 4096, PROT_READ) = 0

     175226 mprotect(0x7ec2ec7a1000, 8192, PROT_READ) = 0

     175226 prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0

     175226 munmap(0x7ec2ec751000, 98135)    = 0

     175226 futex(0x7ec2ec67a7bc, FUTEX_WAKE_PRIVATE, 2147483647) = 0

     175226 getrandom("\xe8\x14\x89\x47\x67\xf4\x84\xb8", 8, GRND_NONBLOCK) = 8

     175226 brk(NULL)                        = 0x6236481d5000

     175226 brk(0x6236481f6000)              = 0x6236481f6000

     175226 rt_sigaction(SIGRT_1, {sa_handler=0x7ec2ec099520, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7ec2ec045320},
NULL, 8) = 0

     175226 rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0) = 0x7ec2eb600000

     175226 mprotect(0x7ec2eb601000, 8388608, PROT_READ|PROT_WRITE) = 0
     175226 rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2ebe00990, parent_tid=0x7ec2ebe00990, exit_signal=0,
stack=0x7ec2eb600000, stack_size=0x7fff80, tls=0x7ec2ebe006c0} =>
{parent_tid=[175227]}, 88) = 175227

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175227 rseq(0x7ec2ebe00fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175227 <... rseq resumed>)              = 0

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0 <unfinished ...>

     175227 set_robust_list(0x7ec2ebe009a0, 24 <unfinished ...>

     175226 <... mmap resumed>)              = 0x7ec2eac00000

     175227 <... set_robust_list resumed>)   = 0

     175226 mprotect(0x7ec2eac01000, 8388608, PROT_READ|PROT_WRITE) = 0

     175226 rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     175227 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175226 <... rt_sigprocmask resumed>[], 8) = 0

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2eb400990, parent_tid=0x7ec2eb400990, exit_signal=0,
stack=0x7ec2eac00000, stack_size=0x7fff80, tls=0x7ec2eb4006c0} <unfinished ...>

     175227 <... rt_sigprocmask resumed>NULL, 8) = 0

     175226 <... clone3 resumed> => {parent_tid=[175228]}, 88) = 175228

     175228 rseq(0x7ec2eb400fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175228 <... rseq resumed>)              = 0

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175228 set_robust_list(0x7ec2eb4009a0, 24 <unfinished ...>

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0 <unfinished ...>

     175228 <... set_robust_list resumed>)   = 0

     175226 <... mmap resumed>)              = 0x7ec2ea200000

     175228 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175227 mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>
```

```
     175226 mprotect(0x7ec2ea201000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     175228 <... rt_sigprocmask resumed>NULL, 8) = 0

     175226 <... mprotect resumed>)          = 0

     175226 rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     175228 mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>

     175227 <... mmap resumed>)              = 0x7ec2e2200000

     175226 <... rt_sigprocmask resumed>[], 8) = 0

     175228 <... mmap resumed>)              = 0x7ec2da200000

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2eaa00990, parent_tid=0x7ec2eaa00990, exit_signal=0,
stack=0x7ec2ea200000, stack_size=0x7fff80, tls=0x7ec2eaa006c0} <unfinished ...>

     175228 munmap(0x7ec2da200000, 31457280 <unfinished ...>

     175227 munmap(0x7ec2e2200000, 31457280 <unfinished ...>

     175228 <... munmap resumed>)            = 0

     175227 <... munmap resumed>)            = 0

     175226 <... clone3 resumed> => {parent_tid=[175229]}, 88) = 175229

     175229 rseq(0x7ec2eaa00fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175228 munmap(0x7ec2e0000000, 35651584 <unfinished ...>

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175229 <... rseq resumed>)              = 0

     175227 munmap(0x7ec2e8000000, 35651584 <unfinished ...>

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175229 set_robust_list(0x7ec2eaa009a0, 24 <unfinished ...>

     175228 <... munmap resumed>)            = 0

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0 <unfinished ...>

     175229 <... set_robust_list resumed>)   = 0

     175227 <... munmap resumed>)            = 0

     175226 <... mmap resumed>)              = 0x7ec2e9800000

     175229 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175228 mprotect(0x7ec2dc000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

     175226 mprotect(0x7ec2e9801000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     175229 <... rt_sigprocmask resumed>NULL, 8) = 0
     175228 <... mprotect resumed>)          = 0

     175227 mprotect(0x7ec2e4000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

     175226 <... mprotect resumed>)          = 0

     175229 mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>

     175227 <... mprotect resumed>)          = 0

     175229 <... mmap resumed>)              = 0x7ec2d4000000

     175229 munmap(0x7ec2d8000000, 67108864 <unfinished ...>

     175226 rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     175229 <... munmap resumed>)            = 0

     175226 <... rt_sigprocmask resumed>[], 8) = 0

     175229 mprotect(0x7ec2d4000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2ea000990, parent_tid=0x7ec2ea000990, exit_signal=0,
stack=0x7ec2e9800000, stack_size=0x7fff80, tls=0x7ec2ea0006c0} <unfinished ...>

     175229 <... mprotect resumed>)          = 0

     175226 <... clone3 resumed> => {parent_tid=[175230]}, 88) = 175230

     175230 rseq(0x7ec2ea000fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175230 <... rseq resumed>)              = 0

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175230 set_robust_list(0x7ec2ea0009a0, 24 <unfinished ...>

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0 <unfinished ...>

     175230 <... set_robust_list resumed>)   = 0

     175226 <... mmap resumed>)              = 0x7ec2e8e00000

     175230 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175226 mprotect(0x7ec2e8e01000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     175230 <... rt_sigprocmask resumed>NULL, 8) = 0

     175226 <... mprotect resumed>)          = 0

     175230 mmap(0x7ec2d8000000, 67108864, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0 <unfinished ...>

     175226 rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     175230 <... mmap resumed>)              = 0x7ec2d0000000

     175226 <... rt_sigprocmask resumed>[], 8) = 0
     175230 mprotect(0x7ec2d0000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2e9600990, parent_tid=0x7ec2e9600990, exit_signal=0,
stack=0x7ec2e8e00000, stack_size=0x7fff80, tls=0x7ec2e96006c0} <unfinished ...>

     175230 <... mprotect resumed>)          = 0

     175226 <... clone3 resumed> => {parent_tid=[175231]}, 88) = 175231

     175231 rseq(0x7ec2e9600fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175231 <... rseq resumed>)              = 0

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175231 set_robust_list(0x7ec2e96009a0, 24 <unfinished ...>

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0 <unfinished ...>

     175231 <... set_robust_list resumed>)   = 0

     175226 <... mmap resumed>)              = 0x7ec2e8400000

     175231 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175226 mprotect(0x7ec2e8401000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     175231 <... rt_sigprocmask resumed>NULL, 8) = 0

     175226 <... mprotect resumed>)          = 0

     175226 rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     175231 mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>

     175226 <... rt_sigprocmask resumed>[], 8) = 0

     175231 <... mmap resumed>)              = 0x7ec2c8000000

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2e8c00990, parent_tid=0x7ec2e8c00990, exit_signal=0,
stack=0x7ec2e8400000, stack_size=0x7fff80, tls=0x7ec2e8c006c0} <unfinished ...>

     175231 munmap(0x7ec2cc000000, 67108864) = 0

     175226 <... clone3 resumed> => {parent_tid=[175232]}, 88) = 175232

     175231 mprotect(0x7ec2c8000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175232 rseq(0x7ec2e8c00fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175231 <... mprotect resumed>)          = 0
     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0 <unfinished ...>

     175232 <... rseq resumed>)              = 0

     175226 <... mmap resumed>)              = 0x7ec2e3600000

     175232 set_robust_list(0x7ec2e8c009a0, 24 <unfinished ...>

     175226 mprotect(0x7ec2e3601000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     175232 <... set_robust_list resumed>)   = 0

     175226 <... mprotect resumed>)          = 0

     175232 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175226 rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     175232 <... rt_sigprocmask resumed>NULL, 8) = 0

     175226 <... rt_sigprocmask resumed>[], 8) = 0

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2e3e00990, parent_tid=0x7ec2e3e00990, exit_signal=0,
stack=0x7ec2e3600000, stack_size=0x7fff80, tls=0x7ec2e3e006c0} <unfinished ...>

     175232 mmap(0x7ec2cc000000, 67108864, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0) = 0x7ec2c4000000

     175226 <... clone3 resumed> => {parent_tid=[175233]}, 88) = 175233

     175233 rseq(0x7ec2e3e00fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175232 mprotect(0x7ec2c4000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175233 <... rseq resumed>)              = 0

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175232 <... mprotect resumed>)          = 0

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0 <unfinished ...>

     175233 set_robust_list(0x7ec2e3e009a0, 24 <unfinished ...>

     175226 <... mmap resumed>)              = 0x7ec2e2c00000

     175233 <... set_robust_list resumed>)   = 0

     175226 mprotect(0x7ec2e2c01000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     175233 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175226 <... mprotect resumed>)          = 0

     175233 <... rt_sigprocmask resumed>NULL, 8) = 0

     175226 rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0
     175233 mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2e3400990, parent_tid=0x7ec2e3400990, exit_signal=0,
stack=0x7ec2e2c00000, stack_size=0x7fff80, tls=0x7ec2e34006c0} <unfinished ...>

     175233 <... mmap resumed>)              = 0x7ec2bc000000

     175233 munmap(0x7ec2c0000000, 67108864 <unfinished ...>

     175226 <... clone3 resumed> => {parent_tid=[175234]}, 88) = 175234

     175234 rseq(0x7ec2e3400fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175233 <... munmap resumed>)            = 0

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175234 <... rseq resumed>)              = 0

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175233 mprotect(0x7ec2bc000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0 <unfinished ...>

     175234 set_robust_list(0x7ec2e34009a0, 24 <unfinished ...>

     175233 <... mprotect resumed>)          = 0

     175226 <... mmap resumed>)              = 0x7ec2e2200000

     175234 <... set_robust_list resumed>)   = 0

     175226 mprotect(0x7ec2e2201000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     175234 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175226 <... mprotect resumed>)          = 0

     175234 <... rt_sigprocmask resumed>NULL, 8) = 0

     175226 rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

     175234 mmap(0x7ec2c0000000, 67108864, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0 <unfinished ...>

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2e2a00990, parent_tid=0x7ec2e2a00990, exit_signal=0,
stack=0x7ec2e2200000, stack_size=0x7fff80, tls=0x7ec2e2a006c0} <unfinished ...>

     175234 <... mmap resumed>)              = 0x7ec2b8000000

     175234 mprotect(0x7ec2b8000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

     175226 <... clone3 resumed> => {parent_tid=[175235]}, 88) = 175235

     175235 rseq(0x7ec2e2a00fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>
     175234 <... mprotect resumed>)          = 0

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175235 <... rseq resumed>)              = 0

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) = 0x7ec2e1800000

     175226 mprotect(0x7ec2e1801000, 8388608, PROT_READ|PROT_WRITE) = 0

     175226 rt_sigprocmask(SIG_BLOCK, ~[], [], 8) = 0

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7ec2e2000990, parent_tid=0x7ec2e2000990, exit_signal=0, stack=0x7ec2e1800000, stack_size=0x7fff80, tls=0x7ec2e20006c0} => {parent_tid=[175236]}, 88) = 175236

     175236 rseq(0x7ec2e2000fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175236 <... rseq resumed>)              = 0

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175236 set_robust_list(0x7ec2e20009a0, 24 <unfinished ...>

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0 <unfinished ...>

     175236 <... set_robust_list resumed>)   = 0

     175226 <... mmap resumed>)              = 0x7ec2e0e00000

     175236 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175226 mprotect(0x7ec2e0e01000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     175236 <... rt_sigprocmask resumed>NULL, 8) = 0

     175226 <... mprotect resumed>)          = 0

     175226 rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     175236 openat(AT_FDCWD, "/sys/devices/system/cpu/online", O_RDONLY|O_CLOEXEC <unfinished ...>

     175226 <... rt_sigprocmask resumed>[], 8) = 0

     175236 <... openat resumed>)            = 3

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7ec2e1600990, parent_tid=0x7ec2e1600990, exit_signal=0, stack=0x7ec2e0e00000, stack_size=0x7fff80, tls=0x7ec2e16006c0} <unfinished ...>

     175236 read(3, "0-11\n", 1024)          = 5

     175226 <... clone3 resumed> => {parent_tid=[175237]}, 88) = 175237

     175237 rseq(0x7ec2e1600fe0, 0x20, 0, 0x53053053 <unfinished ...>
     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175237 <... rseq resumed>)              = 0

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0

     175237 set_robust_list(0x7ec2e16009a0, 24 <unfinished ...>

     175226 mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1,
0) = 0x7ec2e0400000

     175237 <... set_robust_list resumed>)   = 0

     175236 close(3 <unfinished ...>

     175226 mprotect(0x7ec2e0401000, 8388608, PROT_READ|PROT_WRITE <unfinished ...>

     175237 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175226 <... mprotect resumed>)          = 0

     175237 <... rt_sigprocmask resumed>NULL, 8) = 0

     175236 <... close resumed>)             = 0

     175226 rt_sigprocmask(SIG_BLOCK, ~[],  <unfinished ...>

     175237 openat(AT_FDCWD, "/sys/devices/system/cpu/online", O_RDONLY|O_CLOEXEC
<unfinished ...>

     175226 <... rt_sigprocmask resumed>[], 8) = 0

     175236 mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0
<unfinished ...>

     175237 <... openat resumed>)            = 3

     175226 clone3({flags=CLONE_VM|CLONE_FS|CLONE_FILES|CLONE_SIGHAND|CLONE_THREAD|
CLONE_SYSVSEM|CLONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID,
child_tid=0x7ec2e0c00990, parent_tid=0x7ec2e0c00990, exit_signal=0,
stack=0x7ec2e0400000, stack_size=0x7fff80, tls=0x7ec2e0c006c0} <unfinished ...>

     175236 <... mmap resumed>)              = 0x7ec2b0000000

     175237 read(3, "0-11\n", 1024)          = 5

     175236 munmap(0x7ec2b4000000, 67108864 <unfinished ...>

     175226 <... clone3 resumed> => {parent_tid=[175238]}, 88) = 175238

     175237 close(3 <unfinished ...>

     175236 <... munmap resumed>)            = 0

     175238 rseq(0x7ec2e0c00fe0, 0x20, 0, 0x53053053 <unfinished ...>

     175236 mprotect(0x7ec2b0000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

     175237 <... close resumed>)             = 0

     175226 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

     175238 <... rseq resumed>)              = 0

     175226 <... rt_sigprocmask resumed>NULL, 8) = 0
```

```
175237 mmap(0x7ec2b4000000, 67108864, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0 <unfinished ...>

175226 futex(0x7ec2ebe00990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 175227, NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

175238 set_robust_list(0x7ec2e0c009a0, 24 <unfinished ...>

175237 <... mmap resumed>)              = 0x7ec2ac000000

175238 <... set_robust_list resumed>)   = 0

175237 mprotect(0x7ec2ac000000, 135168, PROT_READ|PROT_WRITE <unfinished ...>

175238 rt_sigprocmask(SIG_SETMASK, [],  <unfinished ...>

175237 <... mprotect resumed>)          = 0

175236 <... mprotect resumed>)          = 0

175238 <... rt_sigprocmask resumed>NULL, 8) = 0

175235 set_robust_list(0x7ec2e2a009a0, 24) = 0

175235 rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0

175235 mmap(NULL, 134217728, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ec2a4000000

175235 munmap(0x7ec2a8000000, 67108864) = 0

175235 mprotect(0x7ec2a4000000, 135168, PROT_READ|PROT_WRITE) = 0

175238 mmap(0x7ec2a8000000, 67108864, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ec2a0000000

175238 mprotect(0x7ec2a0000000, 135168, PROT_READ|PROT_WRITE) = 0

175231 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

175231 madvise(0x7ec2e8e00000, 8368128, MADV_DONTNEED) = 0

175231 exit(0)                          = ?

175231 +++ exited with 0 +++

175234 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

175234 madvise(0x7ec2e2c00000, 8368128, MADV_DONTNEED) = 0

175234 exit(0)                          = ?

175234 +++ exited with 0 +++

175228 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

175228 madvise(0x7ec2eac00000, 8368128, MADV_DONTNEED) = 0

175228 exit(0)                          = ?

175228 +++ exited with 0 +++

175227 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

175227 madvise(0x7ec2eb600000, 8368128, MADV_DONTNEED) = 0
```

```
     175227 exit(0)                          = ?

     175227 +++ exited with 0 +++

     175226 <... futex resumed>)             = 0

     175226 futex(0x7ec2eaa00990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 175229,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

     175235 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

     175235 madvise(0x7ec2e2200000, 8368128, MADV_DONTNEED) = 0

     175235 exit(0)                          = ?

     175235 +++ exited with 0 +++

     175230 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

     175230 madvise(0x7ec2e9800000, 8368128, MADV_DONTNEED) = 0

     175230 exit(0)                          = ?

     175230 +++ exited with 0 +++

     175232 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

     175232 madvise(0x7ec2e8400000, 8368128, MADV_DONTNEED) = 0

     175232 exit(0)                          = ?

     175232 +++ exited with 0 +++

     175233 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

     175233 madvise(0x7ec2e3600000, 8368128, MADV_DONTNEED) = 0

     175233 exit(0)                          = ?

     175233 +++ exited with 0 +++

     175236 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

     175236 madvise(0x7ec2e1800000, 8368128, MADV_DONTNEED) = 0

     175236 exit(0)                          = ?

     175236 +++ exited with 0 +++

     175229 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

     175229 madvise(0x7ec2ea200000, 8368128, MADV_DONTNEED) = 0

     175229 exit(0)                          = ?

     175226 <... futex resumed>)             = 0

     175229 +++ exited with 0 +++

     175226 munmap(0x7ec2eb600000, 8392704)  = 0

     175226 munmap(0x7ec2eac00000, 8392704)  = 0

     175226 munmap(0x7ec2ea200000, 8392704)  = 0
```

```
    175226 munmap(0x7ec2e9800000, 8392704)  = 0

    175226 munmap(0x7ec2e8e00000, 8392704)  = 0

    175226 munmap(0x7ec2e8400000, 8392704)  = 0

    175226 futex(0x7ec2e1600990, FUTEX_WAIT_BITSET|FUTEX_CLOCK_REALTIME, 175237,
NULL, FUTEX_BITSET_MATCH_ANY <unfinished ...>

    175238 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

    175238 madvise(0x7ec2e0400000, 8368128, MADV_DONTNEED) = 0

    175238 exit(0)                          = ?

    175238 +++ exited with 0 +++

    175237 rt_sigprocmask(SIG_BLOCK, ~[RT_1], NULL, 8) = 0

    175237 madvise(0x7ec2e0e00000, 8368128, MADV_DONTNEED) = 0

    175237 exit(0)                          = ?

    175237 +++ exited with 0 +++

    175226 <... futex resumed>)             = 0

    175226 munmap(0x7ec2e3600000, 8392704)  = 0

    175226 munmap(0x7ec2e2c00000, 8392704)  = 0

    175226 write(1, "== 235280 ==\n", 13)   = 13

    175226 write(1, "== 0.235280 ==\n\n", 16) = 16

    175226 exit_group(0)                    = ?

    175226 +++ exited with 0 +++
```

Вывод

В ходе лабораторной работы я приобрел практические навыки в управлении потоков и обеспечении синхронизации между потоками. Составил и отладить программу на языке C++, осуществляющую работу с процессами и взаимодействие между ними в одной из двух операционных систем. Как видно по графикам. Не всегда использование atomic ускоряет программу. Реализация на mutex работает быстрее, чем реализация на atomic, хоть разница и не критична. Поэтому выбор между ними зависит от стоящей задачи.