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: М80-206Б-22

Студентка: Шипилова Т.П.

Преподаватель: Миронов Е.С.

Оценка: ____________________

Дата: 01.12.2023 г.

Москва, 2023

Постановка задачи

Вариант 6.

Умножение матриц, содержащих комплексные числа.

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.

Общий метод и алгоритм решения

Использованные системные вызовы:

1. thread(matrix_multiple, ref(matr1), ref(matr2), ref(answer), start, end, m, n, k); – создаёт новый поток;
2. threads[i].join(); – ожидает завершения потока.

Программа условно делит матрицы на части, передает их потокам, где в одном потоке обрабатывается только определенное количество строк первой матрицы, то есть формируем ответ также построчно.

Сами матрицы храним в линеаризованном виде, используя вектор пары. Такой способ хранения данных выбран из-за особенностей данных. Они представлены комплексными числами, то есть содержат действительную и мнимую части. Умножение таких чисел имеет вид:
$(a + bi) * (c + di) = ac + adi + bci - bd.$

Код программы

```
#include<iostream>
#include<chrono>
#include<vector>
#include<thread>
using namespace std;
using namespace std::chrono;

void matrix_multiple(std::vector<std::pair <double, double> >& a,
std::vector<std::pair <double, double> >& b,
                   std::vector<std::pair <double, double> >& ans, size_t start,
size_t end, int m, int n, int k){
   for (size_t i = start; i < end; i++){
       for (size_t j = 0; j < n; j++){
           double d_ans = 0.0; double m_ans = 0.0;
           for (size_t l = 0; l < k; l++){
               std::pair pair1 = a[m*i+l];
               std::pair pair2 = b[n*l+j];
               d_ans += pair1.first * pair2.first - pair1.second * pair2.second;
               m_ans += pair1.first * pair2.second + pair1.second * pair2.first;
           }
           ans.push_back(make_pair(d_ans,m_ans));
       }
   }
}

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

   if (argc != 2){
       cout << "Incorrect usage. Arguments are not instantiated" << endl;
       cout << "Usage: ./lr2 number_of_treads" << endl;
       exit(1);
   }

   size_t num_threads = atoi(argv[1]);

   //Матрица хранится в векторы пары. В первой ячейке действительная часть, во
второй мнимая
   //Пример a + b*i, arr.first = a, arr.second = b
   //Умножение таких чисел: (a+bi)*(c+di) = ac+adi+bci-bd, тогда результат:
res.first=ac-bd, res.second=ad+bc

   cout << "Enter the dimension of the matrices to be multiplied to fill them
with random numbers" << endl;
   cout << "Matrix format: m*n, n*k, enter 3 natural numbers" << endl;
   int m,n,k; cin >> m >> n >> k; cout << endl;
   if (m < num_threads){
       cout << "No need for treads. Parameter m is less then number of threads"
<< endl;
       exit(1);
   }
   if (m < 0 || n < 0 || k < 0){
       cout << "Incorrect values of matrix dimensions. Values should be natural"
<< endl;
       exit(1);
   }

   vector <pair <double, double> > matr1;
   for (int i = 0; i < m*n; i++){
       double a,b;
       a = rand() % 100;
       b = rand() % 100;
       matr1.push_back(make_pair(a,b));
```

```
    }

    vector <pair <double, double> > matr2;
    for (int i = 0; i < k*n; i++){
        double a,b;
        a = rand() % 100;
        b = rand() % 100;
        matr2.push_back(make_pair(a,b));
    }
    vector <pair <double, double> > answer(n*k);

    vector<thread> threads(num_threads);

    size_t block_size = m / num_threads;
    size_t remainder = m % num_threads;

    auto begining = std::chrono::high_resolution_clock::now();

    size_t start = 0;
    for (int i = 0; i < num_threads; i++) {
        size_t end = start + block_size;
        if (i < remainder) {
            end++;
        }
        threads[i] = thread( matrix_multiple, ref(matr1), ref(matr2),
ref(answer), start, end, m, n, k);
        start = end;
    }

    for (int i = 0; i < num_threads; i++) {
        threads[i].join();
    }

    auto ending = std::chrono::high_resolution_clock::now();

    duration<double> sec = ending - begining;
    cout << "Result: ";
    cout << sec.count() << " s" << std::endl;
    answer.clear(); matr1.clear(); matr2.clear();

    return 0;
}
```

Протокол работы программы

Тестирование:

```
./lr2 1
Enter the dimension of the matrices to be multiplied to fill them with random numbers
Matrix format: m*n, n*k, enter 3 natural numbers
6 6 6
Result: 0.000411662 s
./lr2 2
Enter the dimension of the matrices to be multiplied to fill them with random numbers
Matrix format: m*n, n*k, enter 3 natural numbers
6 6 6
Result: 0.000893935 s
./lr2 3
Enter the dimension of the matrices to be multiplied to fill them with random numbers
Matrix format: m*n, n*k, enter 3 natural numbers
6 6 6
Result: 0.00234427 s
./lr2 4
Enter the dimension of the matrices to be multiplied to fill them with random numbers
Matrix format: m*n, n*k, enter 3 natural numbers
6 6 6
Result: 0.00123548 s
./lr2 5
Enter the dimension of the matrices to be multiplied to fill them with random numbers
Matrix format: m*n, n*k, enter 3 natural numbers
6 6 6
Result: 0.00164288 s
```

Количество потоков	Время, с	Ускорение	Эффективность
1	0.000411662	1	1
2	0.000893935	0,460505518	0,230252759
3	0.00234427	0,175603493	0,058534498
4	0.00123548	0,333200052	0,083300013
5	0.00164288	0,250573383	0,050114677

При анализе таблицы становится понятно, что выделение потоков значительно превышает по времени математические операции. При 4 потоках можем увидеть небольшое повышение эффективности. Можно объяснить это полным задействованием ресурсов машины, то есть 4 потоков на 2 ядрах.

Strace:

```
strace ./lr2 1

execve("./lr2", ["./lr2", "1"], 0x7ffc05820da8 /* 74 vars */) = 0

brk(NULL)                               = 0x555a34778000

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffeffd89a60) = -1 EINVAL (Недопустимый аргумент)

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f905d825000

access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Нет такого файла или каталога)

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

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=68035, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 68035, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f905d814000

close(3)                                = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libstdc++.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\0\0\0\0\0\0\0\0"..., 832) = 832

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2260296, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 2275520, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f905d400000

mprotect(0x7f905d49a000, 1576960, PROT_NONE) = 0

mmap(0x7f905d49a000, 1118208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x9a000) = 0x7f905d49a000

mmap(0x7f905d5ab000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1ab000) = 0x7f905d5ab000

mmap(0x7f905d61b000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x21a000) = 0x7f905d61b000

mmap(0x7f905d629000, 10432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f905d629000

close(3)                                = 0

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

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

newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=125488, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 127720, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f905d7f4000

mmap(0x7f905d7f7000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x3000) = 0x7f905d7f7000

mmap(0x7f905d80e000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000)
= 0x7f905d80e000

mmap(0x7f905d812000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1d000) = 0x7f905d812000

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\0P\237\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

     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

     pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\244;\374\204(\337f#\315I\214\234\f\256\271\32"..., 68, 896)
= 68

     newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2216304, ...}, AT_EMPTY_PATH) = 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, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f905d000000

     mmap(0x7f905d028000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x28000) = 0x7f905d028000

     mmap(0x7f905d1bd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1bd000) = 0x7f905d1bd000

     mmap(0x7f905d215000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x214000) = 0x7f905d215000

     mmap(0x7f905d21b000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f905d21b000

     close(3)                                = 0

     openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.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\0\0\0\0\0\0\0\0"..., 832) = 832

     newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0

     mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f905d70d000

     mmap(0x7f905d71b000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe000) = 0x7f905d71b000

     mmap(0x7f905d797000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x8a000) = 0x7f905d797000

     mmap(0x7f905d7f2000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe4000) = 0x7f905d7f2000

     close(3)                                = 0

     mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f905d70b000

     arch_prctl(ARCH_SET_FS, 0x7f905d70c3c0) = 0

     set_tid_address(0x7f905d70c690)         = 11015

     set_robust_list(0x7f905d70c6a0, 24)     = 0

     rseq(0x7f905d70cd60, 0x20, 0, 0x53053053) = 0

     mprotect(0x7f905d215000, 16384, PROT_READ) = 0

     mprotect(0x7f905d7f2000, 4096, PROT_READ) = 0

     mprotect(0x7f905d812000, 4096, PROT_READ) = 0

     mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f905d709000
```

```
    mprotect(0x7f905d61b000, 45056, PROT_READ) = 0

    mprotect(0x555a32980000, 4096, PROT_READ) = 0

    mprotect(0x7f905d85f000, 8192, PROT_READ) = 0

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

    munmap(0x7f905d814000, 68035)          = 0

    getrandom("\x81\xd8\xf1\x17\x6c\x8a\xcb\x62", 8, GRND_NONBLOCK) = 8

    brk(NULL)                              = 0x555a34778000

    brk(0x555a34799000)                    = 0x555a34799000

    futex(0x7f905d62977c, FUTEX_WAKE_PRIVATE, 2147483647) = 0

    newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...},
AT_EMPTY_PATH) = 0

    write(1, "Enter the dimension of the matri"..., 86Enter the dimension of the matrices
to be multiplied to fill them with random numbers

    ) = 86

    write(1, "Matrix format: m*n, n*k, enter 3"..., 49Matrix format: m*n, n*k, enter 3
natural numbers

    ) = 49

    newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...},
AT_EMPTY_PATH) = 0

    read(0, 6

    "6\n", 1024)                    = 2

    read(0, 6

    "6\n", 1024)                    = 2

    read(0, 6

    "6\n", 1024)                    = 2

    write(1, "\n", 1

    )                         = 1

    rt_sigaction(SIGRT_1, {sa_handler=0x7f905d091870, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f905d042520}, 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) =
0x7f905c7ff000

    mprotect(0x7f905c800000, 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|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f905cfff910,
parent_tid=0x7f905cfff910, exit_signal=0, stack=0x7f905c7ff000, stack_size=0x7fff00,
tls=0x7f905cfff640} => {parent_tid=[11088]}, 88) = 11088

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

```
write(1, "Result: 0.00215949 s\n", 21Result: 0.00215949 s
)  = 21
lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Недопустимая операция смещения)
exit_group(0)                           = ?
+++ exited with 0 +++
strace ./lr2 4
execve("./lr2", ["./lr2", "4"], 0x7fffc9daf648 /* 74 vars */) = 0
brk(NULL)                               = 0x561a4f476000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffef7fdd850) = -1 EINVAL (Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f902e8e4000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (Нет такого файла или каталога)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=68035, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 68035, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f902e8d3000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libstdc++.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\0\0\0\0\0\0\0\0"..., 832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2260296, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 2275520, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f902e600000
mprotect(0x7f902e69a000, 1576960, PROT_NONE) = 0
mmap(0x7f902e69a000, 1118208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x9a000) = 0x7f902e69a000
mmap(0x7f902e7ab000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1ab000) = 0x7f902e7ab000
mmap(0x7f902e81b000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x21a000) = 0x7f902e81b000
mmap(0x7f902e829000, 10432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f902e829000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libgcc_s.so.1", O_RDONLY|O_CLOEXEC) = 3
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
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=125488, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 127720, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f902e8b3000
mmap(0x7f902e8b6000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x3000) = 0x7f902e8b6000
mmap(0x7f902e8cd000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000)
= 0x7f902e8cd000
```

```
     mmap(0x7f902e8d1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1d000) = 0x7f902e8d1000

     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\0P\237\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

     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

     pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\244;\374\204(\337f#\315I\214\234\f\256\271\32"..., 68, 896)
= 68

     newfstatat(3, "", {st_mode=S_IFREG|0755, st_size=2216304, ...}, AT_EMPTY_PATH) = 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, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f902e200000

     mmap(0x7f902e228000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x28000) = 0x7f902e228000

     mmap(0x7f902e3bd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1bd000) = 0x7f902e3bd000

     mmap(0x7f902e415000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x214000) = 0x7f902e415000

     mmap(0x7f902e41b000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7f902e41b000

     close(3)                               = 0

     openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libm.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\0\0\0\0\0\0\0\0"..., 832) = 832

     newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=940560, ...}, AT_EMPTY_PATH) = 0

     mmap(NULL, 942344, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f902e519000

     mmap(0x7f902e527000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe000) = 0x7f902e527000

     mmap(0x7f902e5a3000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x8a000) = 0x7f902e5a3000

     mmap(0x7f902e5fe000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe4000) = 0x7f902e5fe000

     close(3)                               = 0

     mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f902e8b1000

     arch_prctl(ARCH_SET_FS, 0x7f902e8b23c0) = 0

     set_tid_address(0x7f902e8b2690)        = 11189

     set_robust_list(0x7f902e8b26a0, 24)    = 0
     rseq(0x7f902e8b2d60, 0x20, 0, 0x53053053) = 0

     mprotect(0x7f902e415000, 16384, PROT_READ) = 0

     mprotect(0x7f902e5fe000, 4096, PROT_READ) = 0

     mprotect(0x7f902e8d1000, 4096, PROT_READ) = 0

     mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f902e8af000

     mprotect(0x7f902e81b000, 45056, PROT_READ) = 0

     mprotect(0x561a4ddc9000, 4096, PROT_READ) = 0

     mprotect(0x7f902e91e000, 8192, PROT_READ) = 0

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

     munmap(0x7f902e8d3000, 68035)          = 0

     getrandom("\x85\x5d\x1b\x1f\x74\xd6\x70\xd1", 8, GRND_NONBLOCK) = 8

     brk(NULL)                              = 0x561a4f476000

     brk(0x561a4f497000)                    = 0x561a4f497000

     futex(0x7f902e82977c, FUTEX_WAKE_PRIVATE, 2147483647) = 0

     newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...},
AT_EMPTY_PATH) = 0

     write(1, "Enter the dimension of the matri"..., 86Enter the dimension of the matrices
to be multiplied to fill them with random numbers

     ) = 86

     write(1, "Matrix format: m*n, n*k, enter 3"..., 49Matrix format: m*n, n*k, enter 3
natural numbers

     ) = 49

     newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...},
AT_EMPTY_PATH) = 0

     read(0, 6

     "6\n", 1024)                     = 2

     read(0, 6

     "6\n", 1024)                     = 2

     read(0, 6

     "6\n", 1024)                     = 2

     write(1, "\n", 1

     )                          = 1

     rt_sigaction(SIGRT_1, {sa_handler=0x7f902e291870, sa_mask=[],
sa_flags=SA_RESTORER|SA_ONSTACK|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f902e242520}, 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) =
0x7f902d9ff000
```

```
    mprotect(0x7f902da00000, 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|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f902e1ff910,
parent_tid=0x7f902e1ff910, exit_signal=0, stack=0x7f902d9ff000, stack_size=0x7fff00,
tls=0x7f902e1ff640} => {parent_tid=[11209]}, 88) = 11209

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

    mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f902d1fe000

    mprotect(0x7f902d1ff000, 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|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f902d9fe910,
parent_tid=0x7f902d9fe910, exit_signal=0, stack=0x7f902d1fe000, stack_size=0x7fff00,
tls=0x7f902d9fe640} => {parent_tid=[11210]}, 88) = 11210

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

    mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f902c9fd000

    mprotect(0x7f902c9fe000, 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|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f902d1fd910,
parent_tid=0x7f902d1fd910, exit_signal=0, stack=0x7f902c9fd000, stack_size=0x7fff00,
tls=0x7f902d1fd640} => {parent_tid=[11211]}, 88) = 11211

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

    mmap(NULL, 8392704, PROT_NONE, MAP_PRIVATE|MAP_ANONYMOUS|MAP_STACK, -1, 0) =
0x7f902c1fc000

    mprotect(0x7f902c1fd000, 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|CL
ONE_SETTLS|CLONE_PARENT_SETTID|CLONE_CHILD_CLEARTID, child_tid=0x7f902c9fc910,
parent_tid=0x7f902c9fc910, exit_signal=0, stack=0x7f902c1fc000, stack_size=0x7fff00,
tls=0x7f902c9fc640} => {parent_tid=[11212]}, 88) = 11212

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

    write(1, "Result: 0.00341146 s\n", 21Result: 0.00341146 s

    )  = 21

    lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Недопустимая операция смещения)

    exit_group(0)                           = ?

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

Вывод

В ходе данной лабораторной работы я научилась создавать в своей программе потоки и использовать их для экономии времени вычислений. Я разделяла матрицу на части и передавала ее потокам. К сожалению, это, по сути, не могло ускорить решения, так как сложность все равно осталась O(n*m*k), где n,m,k - размерности умножаемых матриц. Но при выделении определенного числа потоков можно было увидеть небольшое повышение эффективности, что свидетельствует о том, что распараллеливание вычислений способствует ускорению работы программы.