

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №4
по курсу «Операционные системы»

Выполнил: Д. А. Алгиничев
Группа: М8О-208БВ-24
Преподаватель: Е. С. Миронов
Дата: 25.12.2025

Москва, 2025

Условие

Составить и отладить программу, осуществляющую работу с динамическими библиотеками и их загрузку во время выполнения. Программа должна поддерживать возможность смены реализаций алгоритмов без перекомпиляции.

Цель работы

Приобретение навыков в создании динамических библиотек и программ, которые используют функции динамических библиотек.

Задание

Требуется создать динамические библиотеки, которые реализуют заданных вариантом функционал. Далее использовать данные библиотеки 2-мя способами: 1. Во время компиляции (на этапе "линковки"). 2. Во время выполнения программы. Библиотеки загружаются в память с помощью интерфейса ОС для работы с динамическими библиотеками.

Задание варианта

Реализовать программу, которая может загружать различные реализации алгоритмов вычисления производной и сортировки массивов через динамические библиотеки. Программа должна предоставлять интерфейс для смены библиотек во время выполнения.

Вариант

15

Архитектура проекта

Архитектура проекта реализует клиент-серверную модель с использованием разделяемой памяти (shared memory) для межпроцессного взаимодействия. Система состоит из трёх ключевых компонентов, определённых в файлах проекта: Сервер (Server.hpp, Server.cpp) — центральный координатор, который управляет всеми игровыми сессиями, обрабатывает команды клиентов и поддерживает целостность игрового состояния. Сервер работает в бесконечном цикле, ожидая сообщений в общей очереди. Клиенты (Client.hpp, Client.cpp) — независимые процессы, предоставляющие консольный интерфейс игрокам. Каждый клиент подключается к существующей разделяемой памяти, регистрируется с уникальным логином и взаимодействует с системой через меню. Разделяемая память (SharedTypes.hpp, SharedMemory.hpp, SharedMemory.cpp) — общая область памяти, содержащая структуру SharedMemoryRoot с очередью сообщений, слотами клиентов и массивами игр. Этот компонент обеспечивает высокоскоростное взаимодействие между процессами. Все компоненты взаимодействуют через единую структуру данных, определённую в SharedTypes.hpp, которая включает типы сообщений (MsgType), состояния ячеек (CellState), кораблей (Ship) и игр (GameData). Синхронизация осуществляется с помощью POSIX мьютексов и условных переменных, также определённых в структуре разделяемой памяти.

Метод решения

Метод решения основан на использовании POSIX Shared Memory для реализации меж-процессного взаимодействия между сервером и клиентами. В файле SharedTypes.hpp определена центральная структура SharedMemoryRoot, содержащая: Очередь сообщений (Message queue[QUEUE_SIZE])
`MsgType : (MSG_REGISTER), (MSG_CREATE), (MSG_SHOT). (Client.cpp) : MSG_REGISTER(, ,)..Game(Game.hpp, Game.cpp), (, ,), .().().mutexSharedMemoryRoot., (raceconditions).(server_condcondClientSlot). : (0 - 9), (,), (,).`

Описание программы

Сервер запускается первым и выполняет следующие функции: Создаёт и инициализирует разделяемую память (если не существует). Устанавливает структуры данных: очередь сообщений, слоты клиентов, игровые сессии. Инициализирует мьютексы и условные переменные с атрибутами PTHREAD_PROCESS_SHARED., . : , , , , .

Клиент (client)

Клиентское приложение предоставляет пользователю консольный интерфейс с контекстными меню: Главное меню (отображается когда игрок не в игре): Список игроков и игр (команда list → MSG_LIST)(MSG_CREATE)ID(MSG_JOIN)(MSG_INVITE)(MSG_QUIT)() :
`: place, x, y, MSG_PLACE_SHIP : auto(Client.cpp) : boardMSG_GET_BOARD : readyMSG_GETUP inviteMSG_INVITE_TO_GAME : menuMSG_LEAVE_GAME`

Игровое меню (после начала боя)

Сделать выстрел: ввод координат → MSG_SHOTMSG_GET_BOARDMSG_GET_OPPONENT_BOARD

Игровой процесс

Игра следует классическим правилам "Морского боя": Поле 10×10 клеток. Флот из 10 кораблей: 1×4, 2×3, 3×2, 4×1 клетки. Расстановка с расстоянием минимум 1 клетка между кораблями. Поочерёдные ходы с правом дополнительного хода при попадании. Отображение полей: своё поле показывает корабли, поле противника скрывает неподбитые корабли. Победа при уничтожении всех кораблей противника.

Особенности реализации

Автоматическая расстановка использует алгоритм случайного размещения с проверкой корректности позиции. Система приглашений поддерживает два сценария: создание новой приватной игры и приглашение в существующую игру. Визуализация полей с помощью символов ASCII: . — пусто, S — корабль (на своём поле), X — попадание, O — промах, — потопленный корабль. Обработка отключений: сервер обнаруживает неактивных клиентов и корректно завершает игры. Масштабирование: ограничение на 32 одновременных клиента и 16 активных игр (константы в SharedTypes.hpp).

Результаты тестирования

Пример работы программы

```
Введите, что хотите сделать
-1 Конец программы
0 Смена библиотеки
1 Производная
2 Сортировка
0
Введите название библиотеки
libcontract1.so
Введите, что хотите сделать
1 Производная
Введите точку
1.57
Введите отклонение
0.01
-0.99995
Введите, что хотите сделать
2 Сортировка
Введите размер массива
5
Введите элементы массива
5 3 1 4 2
1 2 3 4 5
```

Сравнение алгоритмов

Сортировка пузырьком vs Быстрая сортировка:

- **BubbleSort:** Время $O(n^2)$
- **QuickSort:** Время $O(n \log n)$ (В СРЕДНЕМ!)

Производная методом 1 vs методом 2:

- **Derivative1:** $(f(x+h) - f(x)) / h$ - точность $O(h)$
- **Derivative2:** $(f(x+h) - f(x-h)) / (2h)$ - точность $O(h^2)$

Выводы

В процессе выполнения лабораторной работы я получил практический опыт работы с динамическими библиотеками и двумя типами их использования.

Статическое связывание - библиотеки линкуются на этапе компиляции. Преимущества: высокая производительность, простота развертывания. Недостатки: большой размер исполняемого файла, невозможность обновления без перекомпиляции.

Динамическая загрузка - библиотеки загружаются во время выполнения. Преимущества: гибкость, возможность обновления, разделение кода. Недостатки: дополнительная сложность, overhead на загрузку.

Приобретенные навыки

- **Работа с dlopen/dlsym** - освоил механизм динамической загрузки в Linux
- **Архитектура плагинов** - научился проектировать системы с подключаемыми модулями
- **Создание shared libraries** - изучил процесс сборки .so файлов

Заключение

Динамическая загрузка библиотек открывает большие возможности для создания гибких и расширяемых приложений. Несмотря на дополнительную сложность, этот подход незаменим в случаях, когда требуется:

- Поддержка плагинов и расширений
- Обновление компонентов без перекомпиляции

Полученные знания позволяют создавать более архитектурно сложные и поддерживаемые приложения.

Исходная программа

```
1  #include <iostream>
2  #include "../include/AbsDerivative.hpp"
3  #include "../include/AbsSort.hpp"
4
5  extern "C"
6  {
7      AbsDerivative * CreateDerivative();
8      AbsSort * CreateSort();
9      void DeleteDerivative(AbsDerivative*);
10     void DeleteSort(AbsSort*);
11 }
12
13 int main(int argc, char * argv[])
14 {
15     AbsDerivative * der = CreateDerivative();
16     AbsSort * sort = CreateSort();
17
18     while (true)
19     {
20         int cmd = 0;
21         std::cout << "Введите, чтохотитесделать" << std::endl;
22         std::cout << "0 Конецпрограммы" << std::endl;
23         std::cout << "1 Производная" << std::endl;
24         std::cout << "2 Сортировка" << std::endl;
25         std::cin >> cmd;
26
27         if (cmd == 1)
```

```

28     {
29         float a;
30         float dx;
31         std::cout << "Введите точку" << std::endl;
32         std::cin >> a;
33         std::cout << "Введите отклонение" << std::endl;
34         std::cin >> dx;
35         std::cout << der->Derivative(a, dx) << std::endl;
36     }
37
38     else if (cmd == 2)
39     {
40         int n;
41         std::cout << "Введите размермассива" << std::endl;
42         std::cin >> n;
43         int * array = new int[n];
44
45         size_t i = 0;
46         std::cout << "Введите элементымассива" << std::endl;
47         for (i = 0; i < n; ++i) std::cin >> array[i];
48
49         sort->Sort(array, n);
50
51         for (i = 0; i < n; ++i) std::cout << array[i] << " ";
52         std::cout << std::endl;
53
54         delete[] array;
55     }
56
57     else if (cmd == 0) break;
58 }
59 DeleteDerivative(der);
60 DeleteSort(sort);
61
62 return 0;
63 }

```

Листинг 1: main.cpp (program1) - пример статического использования библиотеки с контрактом 1

```

1  #include <iostream>
2  #include "DynamicLoader.hpp"
3
4  int main() {
5      DynamicLoader loader;
6      AbsDerivative* der = nullptr;
7      AbsSort* sorter = nullptr;
8
9      if (!loader.Load("./libcontract1.so")) {
10         std::cout << "Initial load error: " << loader.Error() << std::endl;

```

```

11     return 1;
12 }
13
14 der = loader.CreateDerivative();
15 sorter = loader.CreateSort();
16
17 // Проверка создания объектов
18 if (!der || !sorter) {
19     std::cout << "Failed to create objects from initial library" << std::
        endl;
20     if (!der) std::cout << "Derivative object is null" << std::endl;
21     if (!sorter) std::cout << "Sort object is null" << std::endl;
22     return 1;
23 }
24
25 while (true) {
26     std::cout << "Введите, что хотите сделать" << std::endl;
27     std::cout << "-1 Конец программы" << std::endl;
28     std::cout << "0 Смена библиотеки" << std::endl;
29     std::cout << "1 Производная" << std::endl;
30     std::cout << "2 Сортировка" << std::endl;
31     int cmd;
32     std::cin >> cmd;
33
34     if (cmd == 0) {
35         std::cout << "Введите название библиотеки" << std::endl;
36         std::string lib;
37         std::cin >> lib;
38
39         loader.DeleteDerivative(der);
40         loader.DeleteSort(sorter);
41
42         if (!loader.Load(lib)) {
43             std::cout << "Load error: " << loader.Error() << std::endl;
44             continue;
45         }
46
47         der = loader.CreateDerivative();
48         sorter = loader.CreateSort();
49     }
50     else if (cmd == 1) {
51         float a;
52         float dx;
53         std::cout << "Введите точку" << std::endl;
54         std::cin >> a;
55         std::cout << "Введите отклонение" << std::endl;
56         std::cin >> dx;
57         std::cout << der->Derivative(a, dx) << std::endl;
58     }
59     else if (cmd == 2) {
60         int n;

```

```

61         std::cout << "Введите размермассива" << std::endl;
62         std::cin >> n;
63         int * array = new int[n];
64
65         size_t i = 0;
66         std::cout << "Введите элементымассива" << std::endl;
67         for (i = 0; i < n; ++i) std::cin >> array[i];
68
69         sorter->Sort(array, n);
70
71         for (int i = 0; i < n; i++) std::cout << array[i] << " ";
72         std::cout << std::endl;
73
74         delete[] array;
75     }
76     else if (cmd == -1) break;
77 }
78
79 loader.DeleteDerivative(der);
80 loader.DeleteSort(sorter);
81 }

```

Листинг 2: main.cpp (program2) - пример динамического использования библиотек с переключением контрактов

```

1  #include "DynamicLoader.hpp"
2
3  #include <dlfcn.h>
4
5  DynamicLoader::DynamicLoader()
6      : handle_(nullptr), createDer(nullptr), createSort(nullptr),
7        deleteDer(nullptr), deleteSort(nullptr) {}
8
9  DynamicLoader::~DynamicLoader()
10 {
11     if (handle_) dlclose(handle_);
12 }
13
14 bool DynamicLoader::Load(const std::string& path)
15 {
16     if (handle_) dlclose(handle_);
17     handle_ = dlopen(path.c_str(), RTLD_LAZY);
18     if (!handle_)
19     {
20         lastErr = dlerror();
21         return false;
22     }
23
24     createDer = (CreateDerivativeFunc)dlsym(handle_, "CreateDerivative");
25     createSort = (CreateSortFunc)dlsym(handle_, "CreateSort");

```



```

26     deleteDer = (DeleteDerivativeFunc)dlsym(handle_, "DeleteDerivative");
27     deleteSort = (DeleteSortFunc)dlsym(handle_, "DeleteSort");
28
29     if (!createDer || !createSort || !deleteDer || !deleteSort) {
30         lastErr = dlerror();
31         dlclose(handle_);
32         handle_ = nullptr;
33         return false;
34     }
35
36     return true;
37 }
38
39 std::string DynamicLoader::Error() const {
40     return lastErr;
41 }
42
43 AbsDerivative* DynamicLoader::CreateDerivative() {
44     return createDer ? createDer() : nullptr;
45 }
46
47 AbsSort* DynamicLoader::CreateSort() {
48     return createSort ? createSort() : nullptr;
49 }
50
51 void DynamicLoader::DeleteDerivative(AbsDerivative* p) {
52     if (deleteDer) deleteDer(p);
53 }
54
55 void DynamicLoader::DeleteSort(AbsSort* p) {
56     if (deleteSort) deleteSort(p);
57 }

```

Листинг 3: DynamicLoader.cpp - класс, выгружающий библиотеки в память во время выполнения программы

```

1  #include "Derivative_1.hpp"
2  #include "BubbleSort.hpp"
3  #include "AbsDerivative.hpp"
4  #include "AbsSort.hpp"
5
6  extern "C"
7  {
8      __attribute__((visibility("default"))) AbsDerivative * CreateDerivative()
9      {
10         return new Derivative1();
11     }
12
13     __attribute__((visibility("default"))) AbsSort * CreateSort()
14     {

```

```

15     return new BubbleSort();
16 }
17
18 __attribute__((visibility("default"))) void DeleteDerivative(AbsDerivative *
19     p)
20 {
21     delete p;
22 }
23
24 __attribute__((visibility("default"))) void DeleteSort(AbsSort * p)
25 {
26     delete p;
27 }

```

Листинг 4: fact1.cpp - сборщик динамической библиотеки, реализующей контракт 1

```

1  #include "Derivative_2.hpp"
2  #include "QuickSort.hpp"
3  #include "AbsDerivative.hpp"
4  #include "AbsSort.hpp"
5
6  extern "C"
7  {
8      __attribute__((visibility("default"))) AbsDerivative * CreateDerivative()
9      {
10         return new Derivative2();
11     }
12
13     __attribute__((visibility("default"))) AbsSort * CreateSort()
14     {
15         return new QuickSort();
16     }
17
18     __attribute__((visibility("default"))) void DeleteDerivative(AbsDerivative *
19         p)
20     {
21         delete p;
22     }
23
24     __attribute__((visibility("default"))) void DeleteSort(AbsSort * p)
25     {
26         delete p;
27     }
28 }

```

Листинг 5: fact2.cpp - сборщик динамической библиотеки, реализующей контракт 2

```

1  #include "Derivative_1.hpp"
2
3  #include <cmath>
4
5  float Derivative1::Derivative(float A, float deltaX)
6  {
7      return (std::cos(A + deltaX) - std::cos(A)) / deltaX;
8  }

```

Листинг 6: Derivative1.cpp - класс, реализующий 1-й способ нахождения производной $\cos(x)$

```

1  #include "Derivative_2.hpp"
2
3  #include <cmath>
4
5  float Derivative2::Derivative(float A, float deltaX)
6  {
7      return (std::cos(A + deltaX) - std::cos(A - deltaX)) / (2.0f * deltaX);
8  }

```

Листинг 7: Derivative2.cpp - класс, реализующий 2-й способ нахождения производной $\cos(x)$

```

1  #include "BubbleSort.hpp"
2
3  #include <algorithm>
4
5  int * BubbleSort::Sort(int * array, int size)
6  {
7      int i = 0;
8      int j = 0;
9
10     for (i = 0; i < size - 1; ++i)
11     {
12         for (j = 0; j < size - i - 1; ++j)
13         {
14             if (array[j] > array[j + 1])
15             {
16                 std::swap(array[j], array[j + 1]);
17             }
18         }
19     }
20
21     return array;
22 }

```

Листинг 8: BubbleSort.cpp - класс, алгоритм пузырьковой сортировки для целочисленных массивов

```

1  #include "QuickSort.hpp"
2
3  #include <algorithm>
4
5  void QuickSort::quicksort(int * array, int low, int high)
6  {
7      int i = low;
8      int j = high;
9      int pivot = array[high];
10
11     while (i <= j)
12     {
13         while (array[i] < pivot) ++i;
14         while (array[j] < pivot) ++j;
15
16         if (i <= j)
17         {
18             std::swap(array[i], array[j]);
19             ++i;
20             --j;
21         }
22     }
23
24     if (low < j) quicksort(array, low, j);
25     if (high > i) quicksort(array, high, i);
26 }
27
28 int * QuickSort::Sort(int * array, int size)
29 {
30     if (size > 1)
31     {
32         quicksort(array, 0, size - 1);
33     }
34
35     return array;
36 }

```

Листинг 9: QuickSort.cpp - класс, алгоритм быстрой сортировки для целочисленных массивов

```

1  #pragma once
2
3  class AbsDerivative
4  {
5      public:
6          virtual float Derivative(float A, float deltaX) = 0;
7          virtual ~AbsDerivative() = default;
8  };

```

Листинг 10: AbsDerivative.hpp - абстрактный класс для реализации производной косинуса

```
1 #pragma once
2
3 class AbsSort
4 {
5     public:
6         virtual int * Sort(int * array, int size) = 0;
7         virtual ~AbsSort() = default;
8 };
```

Листинг 11: AbsSort.hpp - абстрактный класс для реализации сортировки целочисленных массивов

Системные вызовы статической библиотеки

```
execve("./program1",["./program1"],0x7ffd9a10e080 /* 36 vars */) = 0
brk(NULL) = 0x55a72a072000
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7be415b94000
access("/etc/ld.so.preload",R_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/daalginichev/OS/Lab_4/build/glibc-hwcaps/x86-64-v3/libcontract
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/daalginichev/OS/Lab_4/build/glibc-hwcaps/x86-64-v3/",0x7ff
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/daalginichev/OS/Lab_4/build/glibc-hwcaps/x86-64-v2/libcontract
= -1 ENOENT (No such file or directory)
newfstatat(AT_FDCWD,"/home/daalginichev/OS/Lab_4/build/glibc-hwcaps/x86-64-v2/",0x7ff
= -1 ENOENT (No such file or directory)
openat(AT_FDCWD,"/home/daalginichev/OS/Lab_4/build/libcontract1.so",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\0"... ,832)
= 832
fstat(3,{st_mode=S_IFREG|0755,st_size=22160,...}) = 0
mmap(NULL,20632,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7be415b8e000
mmap(0x7be415b90000,4096,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x
= 0x7be415b90000
mmap(0x7be415b91000,4096,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x3000)
= 0x7be415b91000
mmap(0x7be415b92000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0
= 0x7be415b92000
close(3) = 0
openat(AT_FDCWD,"/home/daalginichev/OS/Lab_4/build/libstdc++.so.6",O_RDONLY|O_CLOEXEC
= -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=34331,...}) = 0
mmap(NULL,34331,PROT_READ,MAP_PRIVATE,3,0) = 0x7be415b85000
```

```

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"..., 832)
= 832
fstat(3, {st_mode=S_IFREG|0644, st_size=2592224, ...}) = 0
mmap(NULL, 2609472, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7be415800000
mmap(0x7be41589d000, 1343488, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0)
= 0x7be41589d000
mmap(0x7be4159e5000, 552960, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e5000)
= 0x7be4159e5000
mmap(0x7be415a6c000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0)
= 0x7be415a6c000
mmap(0x7be415a7a000, 12608, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0)
= 0x7be415a7a000
close(3) = 0
openat(AT_FDCWD, "/home/daalginichev/OS/Lab_4/build/libc.so.6", O_RDONLY|O_CLOEXEC)
= -1 ENOENT (No such file or directory)
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) = 0x7be415400000
mmap(0x7be415428000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0)
= 0x7be415428000
mmap(0x7be4155b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000)
= 0x7be4155b0000
mmap(0x7be4155ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0)
= 0x7be4155ff000
mmap(0x7be415605000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0)
= 0x7be415605000
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"..., 832)
= 832
fstat(3, {st_mode=S_IFREG|0644, st_size=952616, ...}) = 0
mmap(NULL, 950296, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7be415a9c000
mmap(0x7be415aac000, 520192, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0)
= 0x7be415aac000
mmap(0x7be415b2b000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8f000)
= 0x7be415b2b000
mmap(0x7be415b83000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0)
= 0x7be415b83000
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"...,832)
= 832
fstat(3,{st_mode=S_IFREG|0644,st_size=183024,...}) = 0
mmap(NULL,185256,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x7be4157d2000
mmap(0x7be4157d6000,147456,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7be4157d6000
mmap(0x7be4157fa000,16384,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x28000)
= 0x7be4157fa000
mmap(0x7be4157fe000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x7be4157fe000
close(3) = 0
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7be415a9a000
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x7be415a97000
arch_prctl(ARCH_SET_FS,0x7be415a97740) = 0
set_tid_address(0x7be415a97a10) = 40263
set_robust_list(0x7be415a97a20,24) = 0
rseq(0x7be415a98060,0x20,0,0x53053053) = 0
mprotect(0x7be4155ff000,16384,PROT_READ) = 0
mprotect(0x7be4157fe000,4096,PROT_READ) = 0
mprotect(0x7be415b83000,4096,PROT_READ) = 0
mprotect(0x7be415a6c000,45056,PROT_READ) = 0
mprotect(0x7be415b92000,4096,PROT_READ) = 0
mprotect(0x55a72456e000,4096,PROT_READ) = 0
mprotect(0x7be415bcc000,8192,PROT_READ) = 0
prlimit64(0,RLIMIT_STACK,NULL,{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY})
= 0
munmap(0x7be415b85000,34331) = 0
futex(0x7be415a7a7bc,FUTEX_WAKE_PRIVATE,2147483647) = 0
getrandom("\xf4\x58\xc1\xde\x95\xd0\x9b\xe",8,GRND_NONBLOCK) = 8
brk(NULL) = 0x55a72a072000
brk(0x55a72a093000) = 0x55a72a093000
fstat(1,{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0x5),...}) = 0
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265,\321\207\321\202\320\265\320\276\321\202\320\270\321"...,51) = 51
write(1,"0 \320\232\320\276\320\275\320\265\321\206 \320\277\321\200\320\276\320\263\321"...,32) = 32
write(1,"1 \320\237\321\200\320\276\320\270\320\267\320\262\320\276\320\264\320\275\321"...,25) = 25
write(1,"2 \320\241\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\272\321"...,23) = 23
fstat(0,{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0x5),...}) = 0
read(0,"1\n",1024) = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \321\202\320\276\321"...,26) = 26
read(0,"0\n",1024) = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\276\321\202\320\265"...,26) = 26

```

```

= 36
read(0,"0.00000001\n",1024)          = 11
write(1,"0\n",2)                      = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265,\321\207\321\202\320\265\321\205\320\276\321\202\320\270\321"... ,51) = 51
write(1,"0 \320\232\320\276\320\275\320\265\321\206 \320\277\321\200\320\276\320\263\321"... ,32) = 32
write(1,"1 \320\237\321\200\320\276\320\270\320\267\320\262\320\276\320\264\320\275\321"... ,25) = 25
write(1,"2 \320\241\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\272\321"... ,23) = 23
read(0,"2\n",1024)                    = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \321\200\320\260\320\274\320\260"... ,43) = 43
read(0,"5\n",1024)                    = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \321\215\320\273\321\202\274\321\202\275\321"... ,47) = 47
read(0,"5 4 3 2 1\n",1024)            = 10
write(1,"1 2 3 4 5 \n",11)            = 11
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265,\321\207\321\202\320\265\321\205\320\276\321\202\320\270\321"... ,51) = 51
write(1,"0 \320\232\320\276\320\275\320\265\321\206 \320\277\321\200\320\276\320\263\321"... ,32) = 32
write(1,"1 \320\237\321\200\320\276\320\270\320\267\320\262\320\276\320\264\320\275\321"... ,25) = 25
write(1,"2 \320\241\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\272\321"... ,23) = 23
read(0,"0\n",1024)                    = 2
lseek(0,-1,SEEK_CUR)                   = -1 ESPIPE (Illegal seek)
exit_group(0)                          = ?
+++ exited with 0 +++

```

Системные вызовы динамических библиотек

```

execve("./program2",["./program2"],0x7ffffaf0efe10 /* 36 vars */) = 0
brk(NULL)                      = 0x60bbb5bf5000
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x725ea6607000
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=34331,...}) = 0
mmap(NULL,34331,PROT_READ,MAP_PRIVATE,3,0) = 0x725ea65fe000
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\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"... ,832) = 832
fstat(3,{st_mode=S_IFREG|0644,st_size=2592224,...}) = 0
mmap(NULL,2609472,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x725ea6200000
mmap(0x725ea629d000,1343488,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0) = 0x725ea629d000

```



```

= 0x725ea629d000
mmap(0x725ea63e5000,552960,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x1e5000)
= 0x725ea63e5000
mmap(0x725ea646c000,57344,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,
= 0x725ea646c000
mmap(0x725ea647a000,12608,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1
= 0x725ea647a000
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"... ,832)
= 832
fstat(3,{st_mode=S_IFREG|0644,st_size=183024,...}) = 0
mmap(NULL,185256,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x725ea65d0000
mmap(0x725ea65d4000,147456,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,
= 0x725ea65d4000
mmap(0x725ea65f8000,16384,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x28000)
= 0x725ea65f8000
mmap(0x725ea65fc000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0
= 0x725ea65fc000
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\0\3\0>\0\1\0\0\0\0\0\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\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\0\0"... ,784,64)
= 784
mmap(NULL,2170256,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x725ea5e00000
mmap(0x725ea5e28000,1605632,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,
= 0x725ea5e28000
mmap(0x725ea5fb0000,323584,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x1b0000)
= 0x725ea5fb0000
mmap(0x725ea5fff000,24576,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,
= 0x725ea5fff000
mmap(0x725ea6005000,52624,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,-1
= 0x725ea6005000
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\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"... ,832)
= 832
fstat(3,{st_mode=S_IFREG|0644,st_size=952616,...}) = 0
mmap(NULL,950296,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x725ea64e7000
mmap(0x725ea64f7000,520192,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,
= 0x725ea64f7000
mmap(0x725ea6576000,360448,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x8f000)
= 0x725ea6576000

```

```

mmap(0x725ea65ce000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x725ea65ce000
close(3) = 0
mmap(NULL,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x725ea64e5000
mmap(NULL,12288,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0) = 0x725ea64e2000
arch_prctl(ARCH_SET_FS,0x725ea64e2740) = 0
set_tid_address(0x725ea64e2a10) = 38775
set_robust_list(0x725ea64e2a20,24) = 0
rseq(0x725ea64e3060,0x20,0,0x53053053) = 0
mprotect(0x725ea5fff000,16384,PROT_READ) = 0
mprotect(0x725ea65ce000,4096,PROT_READ) = 0
mprotect(0x725ea65fc000,4096,PROT_READ) = 0
mprotect(0x725ea646c000,45056,PROT_READ) = 0
mprotect(0x60bba64b7000,4096,PROT_READ) = 0
mprotect(0x725ea663f000,8192,PROT_READ) = 0
prlimit64(0,RLIMIT_STACK,NULL,{rlim_cur=8192*1024,rlim_max=RLIM64_INFINITY})
= 0
munmap(0x725ea65fe000,34331) = 0
futex(0x725ea647a7bc,FUTEX_WAKE_PRIVATE,2147483647) = 0
getrandom("\x55\x7d\xae\x92\x05\x21\x91\x46",8,GRND_NONBLOCK) = 8
brk(NULL) = 0x60bbb5bf5000
brk(0x60bbb5c16000) = 0x60bbb5c16000
openat(AT_FDCWD,"./libcontract1.so",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
fstat(3,{st_mode=S_IFREG|0755,st_size=22160,...}) = 0
getcwd("/home/daalginichev/OS/Lab_4/build",128) = 34
mmap(NULL,20632,PROT_READ,MAP_PRIVATE|MAP_DENYWRITE,3,0) = 0x725ea6601000
mmap(0x725ea6603000,4096,PROT_READ|PROT_EXEC,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x725ea6603000
mmap(0x725ea6604000,4096,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x3000)
= 0x725ea6604000
mmap(0x725ea6605000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0)
= 0x725ea6605000
close(3) = 0
mprotect(0x725ea6605000,4096,PROT_READ) = 0
fstat(1,{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0x5),...}) = 0
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265,\321\207\321\202\320\265\320\276\321\202\320\270\321"... ,51) = 51
write(1,"-1 \320\232\320\276\320\275\320\265\321\206 \320\277\321\200\320\276\320\263\321\207\321\202\320\265\320\276\321\202\320\270\321"... ,51) = 51
write(1,"0 \320\241\320\274\320\265\320\275\320\260 \320\261\320\270\320\261\320\273\321\207\321\202\320\265\320\276\321\202\320\270\321"... ,51) = 51
write(1,"1 \320\237\321\200\320\276\320\270\320\267\320\262\320\276\320\264\320\275\321\207\321\202\320\265\320\276\321\202\320\270\321"... ,51) = 51
write(1,"2 \320\241\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\272\321\207\321\202\320\265\320\276\321\202\320\270\321"... ,51) = 51
fstat(0,{st_mode=S_IFCHR|0620,st_rdev=makedev(0x88,0x5),...}) = 0

```

[illegible]

```

= 0x725ea6603000
mmap(0x725ea6604000,4096,PROT_READ,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0x3000)
= 0x725ea6604000
mmap(0x725ea6605000,8192,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,3,0
= 0x725ea6605000
close(3) = 0
mprotect(0x725ea6605000,4096,PROT_READ) = 0
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265,\321\207\321\202\32
\321\205\320\276\321\202\320\270\321"... ,51) = 51
write(1,"-1 \320\232\320\276\320\275\320\265\321\206 \320\277\321\200\320\276\320\263
= 33
write(1,"0 \320\241\320\274\320\265\320\275\320\260 \320\261\320\270\320\261\320\273\
= 34
write(1,"1 \320\237\321\200\320\276\320\270\320\267\320\262\320\276\320\264\320\275\3
= 25
write(1,"2 \320\241\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\272\3
= 23
read(0,"1\n",1024) = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \321\202\320\276\32
= 26
read(0,"0\n",1024) = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\276\321\202\32
= 36
read(0,"0.0000001\n",1024) = 10
write(1,"0\n",2) = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265,\321\207\321\202\32
\321\205\320\276\321\202\320\270\321"... ,51) = 51
write(1,"-1 \320\232\320\276\320\275\320\265\321\206 \320\277\321\200\320\276\320\263
= 33
write(1,"0 \320\241\320\274\320\265\320\275\320\260 \320\261\320\270\320\261\320\273\
= 34
write(1,"1 \320\237\321\200\320\276\320\270\320\267\320\262\320\276\320\264\320\275\3
= 25
write(1,"2 \320\241\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\272\3
= 23
read(0,"2\n",1024) = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \321\200\320\260\32
\320\274\320\260"... ,43) = 43
read(0,"5\n",1024) = 2
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \321\215\320\273\32
"... ,47) = 47
read(0,"5 4 3 2 1\n",1024) = 10
write(1,"1 2 3 4 5 \n",11) = 11
write(1,"\320\222\320\262\320\265\320\264\320\270\321\202\320\265,\321\207\321\202\32
\321\205\320\276\321\202\320\270\321"... ,51) = 51
write(1,"-1 \320\232\320\276\320\275\320\265\321\206 \320\277\321\200\320\276\320\263
= 33
write(1,"0 \320\241\320\274\320\265\320\275\320\260 \320\261\320\270\320\261\320\273\

```

[illegible]