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»

Группа: М80-206Б-20

Студент: Шипилова Т.П.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 16.12.23

Москва, 2023

Постановка задачи

Вариант 26.

Требуется создать динамические библиотеки, которые реализуют определенный функционал. Далее использовать данные библиотеки 2-мя способами:

- Во время компиляции (на этапе «линковки»/linking)
- Во время исполнения программы. Библиотеки загружаются в память с помощью интерфейса ОС для работы с динамическими библиотеками В конечном итоге, в лабораторной работе необходимо получить следующие части: Динамические библиотеки, реализующие контракты, которые заданы вариантом;

1. Тестовая программа (программа №1), которая используют одну из библиотек, используя знания полученные на этапе компиляции;
2. Тестовая программа (программа №2), которая загружает библиотеки, используя только их местоположение и контракты. Провести анализ двух типов использования библиотек.

Пользовательский ввод для обоих программ должен быть организован следующим образом:

1. Если пользователь вводит команду «0», то программа переключает одну реализацию контрактов на другую (необходимо только для программы №2).
2. «1 arg1 arg2 … argN», где после «1» идут аргументы для первой функции, предусмотренной контрактами. После ввода команды происходит вызов первой функции, и на экране появляется результат её выполнения;
3. «2 arg1 arg2 … argM», где после «2» идут аргументы для второй функции, предусмотренной контрактами. После ввода команды происходит вызов второй функции, и на экране появляется результат её выполнения.

№	Описание	Сигнатура	Реализация 1	Реализация 2
4	Подсчёт наибольшего общего делителя для двух натуральных чисел	Int GCF(int A, int B)	Алгоритм Евклида	Наивный алгоритм. Пытаться разделить числа на все числа, что меньше A и B.
9	Отсортировать целочисленный массив	Int * Sort(int * array, int n)	Пузырьковая сортировка	Сортировка Хоара

Общий метод и алгоритм решения

Использованные системные вызовы:

- void* gcf_lib = dlopen(gcf_lib_path.c_str(), RTLD_LAZY); - открывает файл динамической библиотеки для использования в коде.
- (GCFType)dlsym(gcf_lib, "GCF1"); (SortType)dlsym(sort_lib, "bubble_sort"); - возвращают указатели на библиотечные функции по имени.
- dlclose(gcf_lib); - закрывает файл динамической библиотеки. То, что в скобках - это дескриптор открытой библиотеки.

Код программы

main1.cpp

```
#include <iostream>

#include "GCF/GCF.hpp"

#include "Sort/sort.hpp"


enum Command {

   CHANGE_FUNC = 0,

   GCF = 1,

   SORT = 2,

   EXIT = 3

};


std::istream& operator>>(std::istream& is, Command& cmd){

   int tmp;

   while (((is >> tmp) && (tmp < 0 || tmp > 3))){

       std::cout << "Invalid command" << std::endl;

   }

   cmd = static_cast<Command>(tmp);

   return is;

}


enum Function {
```

```
    FIRST = 1,
    SECOND = 2
};


int main(){


    Command cmd = EXIT;
    Function func = FIRST;


    std::cout<< "Choose command: \n" <<
                    "\t0 - switch func in lib,\n" <<
                    "\t1 - find greatest common fraction,\n" <<
                    "\t2 - sort an array,\n" <<
                    "\t3 - exit." << std::endl;
    std::cout << std::endl;


    while (1){
        if (!(std::cin >> cmd)){
            std::cout << "Invalid input" << std::endl;
            break;
        }
        switch (cmd)
        {
        case CHANGE_FUNC:
            func = (func == FIRST) ? SECOND : FIRST;
            std::cout << "Function switched to " << ((func == FIRST) ?
"first" : "second")
                    << std::endl;
            break;
        case GCF:
```

```
            int a, b;
            std::cin >> a >> b;
            switch (func)
            {
            case FIRST:
                std::cout << GCF1(a, b) << std::endl;
                break;
            case SECOND:
                std::cout << GCF2(a, b) << std::endl;
                break;
            }
            break;
        case SORT:
            int n;
            std::cin >> n;
            int * arr;
            int * sorted;
            std::cout << "\nEnter an array :";
            for (int i = 0; i < n; ++i)
                std::cin >> arr[i];
            switch (func)
            {
            case FIRST:
                sorted = bubble_sort(arr, n);
                break;
            case SECOND:
                sorted = bubble_sort(arr, n);
                break;
            }
```

```
            std::cout << std::endl;
            for (size_t i = 0; i < n; ++i){
                std::cout << sorted[i] << " ";
            }
            std::cout << std::endl;
            break;
        case EXIT:
            exit(0);
        }
    }
    return 0;
}
```

main2.cpp

```
#include <dlfcn.h>
#include <iostream>

enum Command {
    CHANGE_FUNC = 0,
    GCF = 1,
    SORT = 2,
    EXIT = 3
};

std::istream& operator>>(std::istream& is, Command& cmd){
    int tmp;
    while (((is >> tmp) && (tmp < 0 || tmp > 3))){
        std::cout << "Invalid command" << std::endl;
    }
    cmd = static_cast<Command>(tmp);
```

```
    return is;
}

enum Algo {
    FIRST = 1,
    SECOND = 2
};

std::string gcf_lib_path = "./libGCF.so";
std::string sort_lib_path = "./libSort.so";

int main(){

    void* gcf_lib = dlopen(gcf_lib_path.c_str(), RTLD_LAZY);
    void* sort_lib = dlopen(sort_lib_path.c_str(), RTLD_LAZY);

    if (!sort_lib || !gcf_lib) {
        std::cout << "Cannot load library: " << dlerror() << '\n';
        return 1;
    }

    typedef int (*GCFType)(int, int);
    typedef int* (*SortType)(int *, int);

    if (!dlsym(gcf_lib, "GCF1") || !dlsym(gcf_lib, "GCF2") ||
        !dlsym(sort_lib, "bubble_sort") || !dlsym(sort_lib, "hoar_sort")) {
        std::cout << "Cannot load function: " << dlerror() << '\n';
        dlclose(gcf_lib);
        dlclose(sort_lib);
```

```
        return 1;
    }


    Command cmd = EXIT;
    Algo cur_algo = FIRST;


    GCFType GCF_alg = (GCFType)dlsym(gcf_lib, "GCF1");
    SortType Sort_alg = (SortType)dlsym(sort_lib, "bubble_sort");


    std::cout<< "Choose command: \n" <<
                    "\t0 - switch algo in lib,\n" <<
                    "\t1 - calculate sin integral,\n" <<
                    "\t2 - calculate cos derivative,\n" <<
                    "\t3 - exit." << std::endl;
    std::cout << std::endl;


    while (1){
        if (!(std::cin >> cmd)){
            std::cout << "Invalid input" << std::endl;
            break;
        }
        switch (cmd)
        {
        case CHANGE_FUNC:
            cur_algo = (cur_algo == FIRST) ? SECOND : FIRST;
            GCF_alg = (cur_algo == FIRST) ? (GCFType)dlsym(gcf_lib, "GCF1") :
(GCFType)dlsym(gcf_lib, "GCF2");
            Sort_alg = (cur_algo == FIRST) ? (SortType)dlsym(sort_lib,
"bubble_sort") : (SortType)dlsym(sort_lib, "hoar_sort");
            std::cout << "Algo switched to " << ((cur_algo == FIRST) ?
"first" : "second") << std::endl;
```

```
            break;
        case GCF:
            int a, b;
            std::cin >> a >> b;
            std::cout << "GCF is: " << GCF_alg(a, b) << std::endl;
            break;
        case SORT:
            int n;
            std::cin >> n;
            int * arr;
            int * sorted;
            std::cout << "\nEnter an array :";
            for (int i = 0; i < n; ++i)
                std::cin >> arr[i];
            sorted = Sort_alg(arr,n);
            std::cout << std::endl << "Sorted: ";
            for (size_t i = 0; i < n; ++i){
                std::cout << sorted[i] << " ";
            }
            std::cout << std::endl;
            break;
        case EXIT:
            exit(0);
        }
    }
    dlclose(gcf_lib);
    dlclose(sort_lib);
    return 0;
}
```

GCF.cpp

```cpp
#include "GCF.hpp"

#include <cmath>


int GCF1(int A, int B){

   if(A != B){

       if(A > B) {

           A -= B;

           GCF1(A,B);

       }

       else

       if(B > A) {

           B -= A;

           GCF1(A,B);

       }

   }

   else

   return A;

}


int GCF2(int A, int B){

   int nod = 1;

   int min = A > B ? A : B;

   for (int i = min; i > 1; --i)

       if(A % i == 0 && B % i == 0){

           nod = i;

           break;

       }

   return nod;

}
```

sort.cpp

```
#include "sort.hpp"
#include <cmath>


int * bubble_sort(int * arr, int n){
   for (int i = 0; i < n-1; ++i){
       for (int j = i+1; j < n; ++j){
           if (arr[i] > arr[j]){
               int tmp = arr[i];
               arr[i] = arr[j];
               arr[j] = tmp;
           }
       }
   }
   return arr;
}


int * hoar_sort_rec(int * a,int first, int last){

   int i = first, j = last;
   int tmp, x = a[(first + last) / 2];
   do {
       while (a[i] < x)
           i++;
       while (a[j] > x)
           j--;
   if (i <= j)
   {
       if (i < j)
```

```
        {
            tmp=a[i];
            a[i]=a[j];
            a[j]=tmp;
        }
        i++;
        j--;
    }
    } while (i <= j);
    if (i < last)
        hoar_sort_rec(a, i, last);
    if (first < j)
        hoar_sort_rec(a, first,j);
    return a;
}


int * hoar_sort(int * arr, int n){
    int begin = 0;
    return hoar_sort_rec(arr,begin,n-1);
}
```

Протокол работы программы

Тестирование:

```
tanya@tanya:~/Рабочий стол/4$ ./main1
Choose command:
      0 - switch func in lib,
      1 - find greatest common fraction,
      2 - sort an array,
      3 - exit.
========================================
1
5 8
1
1
36 8
4
0
Function switched to second
1
28 32
4
0
Function switched to first
2
4

Enter an array :18 11 2 78

2 11 18 78
0
Function switched to second
2
5

Enter an array :29 11 0 -5 17

-5 0 11 17 29
3
tanya@tanya:~/Рабочий стол/4$ ./main2
Choose command:
      0 - switch algo in lib,
      1 - calculate sin integral,
      2 - calculate cos derivative,
      3 - exit.
========================================
0
Algo switched to second
1
```

```
78 39
GCF is: 39
0
Algo switched to first
1
78 39
GCF is: 39
2
4

Enter an array :5 1 9 6

Sorted: 1 5 6 9
0
Algo switched to second
2
5

Enter an array :9 6 7 3 8

Sorted: 3 6 7 8 9
3
```

Strace:

```
tanya@tanya:~/Рабочий стол/4$ strace ./main2
execve("./main2", ["./main2"], 0x7ffc6dd2a3e0 /* 74 vars */) = 0
brk(NULL)                               = 0x555d8a341000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffcf364e3c0) = -1 EINVAL (Недопустимый аргумент)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fe26dbe4000
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) = 0x7fe26dbd3000
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\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) = 0x7fe26d800000
mprotect(0x7fe26d89a000, 1576960, PROT_NONE) = 0
mmap(0x7fe26d89a000, 1118208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x9a000) = 0x7fe26d89a000
```

```
     mmap(0x7fe26d9ab000, 454656, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1ab000) = 0x7fe26d9ab000

     mmap(0x7fe26da1b000, 57344, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x21a000) = 0x7fe26da1b000

     mmap(0x7fe26da29000, 10432, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7fe26da29000

     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) = 0x7fe26dbb3000

     mmap(0x7fe26dbb6000, 94208, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x3000) = 0x7fe26dbb6000

     mmap(0x7fe26dbcd000, 16384, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a000)
= 0x7fe26dbcd000

     mmap(0x7fe26dbd1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x1d000) = 0x7fe26dbd1000

     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"..., 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) = 0x7fe26d400000

     mmap(0x7fe26d428000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x28000) = 0x7fe26d428000

     mmap(0x7fe26d5bd000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1bd000) = 0x7fe26d5bd000

     mmap(0x7fe26d615000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x214000) = 0x7fe26d615000

     mmap(0x7fe26d61b000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS,
-1, 0) = 0x7fe26d61b000

     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) = 0x7fe26dacc000

     mmap(0x7fe26dada000, 507904, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe000) = 0x7fe26dada000

     mmap(0x7fe26db56000, 372736, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x8a000) = 0x7fe26db56000

     mmap(0x7fe26dbb1000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0xe4000) = 0x7fe26dbb1000

     close(3)                                = 0

     mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fe26daca000

     arch_prctl(ARCH_SET_FS, 0x7fe26dacb3c0) = 0

     set_tid_address(0x7fe26dacb690)         = 6456

     set_robust_list(0x7fe26dacb6a0, 24)     = 0

     rseq(0x7fe26dacbd60, 0x20, 0, 0x53053053) = 0

     mprotect(0x7fe26d615000, 16384, PROT_READ) = 0

     mprotect(0x7fe26dbb1000, 4096, PROT_READ) = 0

     mprotect(0x7fe26dbd1000, 4096, PROT_READ) = 0

     mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fe26dac8000

     mprotect(0x7fe26da1b000, 45056, PROT_READ) = 0

     mprotect(0x555d89e3c000, 4096, PROT_READ) = 0

     mprotect(0x7fe26dc1e000, 8192, PROT_READ) = 0

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

     munmap(0x7fe26dbd3000, 68035)           = 0

     getrandom("\xd8\x7e\xe0\xf9\xb1\xa2\xc9\xaa", 8, GRND_NONBLOCK) = 8

     brk(NULL)                               = 0x555d8a341000

     brk(0x555d8a362000)                     = 0x555d8a362000

     futex(0x7fe26da2977c, FUTEX_WAKE_PRIVATE, 2147483647) = 0

     openat(AT_FDCWD, "./libGCF.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

     newfstatat(3, "", {st_mode=S_IFREG|0775, st_size=15200, ...}, AT_EMPTY_PATH) = 0

     getcwd("/home/tanya/\320\240\320\260\320\261\320\276\321\207\320\270\320\271
\321\201\321\202\320\276\320\273/4", 128) = 38

     mmap(NULL, 16424, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fe26dbdf000

     mmap(0x7fe26dbe0000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1000) = 0x7fe26dbe0000

     mmap(0x7fe26dbe1000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7fe26dbe1000
```

```
     mmap(0x7fe26dbe2000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x2000) = 0x7fe26dbe2000

     close(3)                                = 0

     mprotect(0x7fe26dbe2000, 4096, PROT_READ) = 0

     openat(AT_FDCWD, "./libSort.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

     newfstatat(3, "", {st_mode=S_IFREG|0775, st_size=15432, ...}, AT_EMPTY_PATH) = 0

     getcwd("/home/tanya/\320\240\320\260\320\261\320\276\321\207\320\270\320\271
\321\201\321\202\320\276\320\273/4", 128) = 38

     mmap(NULL, 16432, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fe26dbda000

     mmap(0x7fe26dbdb000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1000) = 0x7fe26dbdb000

     mmap(0x7fe26dbdc000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) =
0x7fe26dbdc000

     mmap(0x7fe26dbdd000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x2000) = 0x7fe26dbdd000

     close(3)                                = 0

     mprotect(0x7fe26dbdd000, 4096, PROT_READ) = 0

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

     write(1, "Choose command: \n", 17Choose command:

     )     = 17

     write(1, "\t0 - switch algo in lib,\n", 25    0 - switch algo in lib,

     ) = 25

     write(1, "\t1 - calculate sin integral,\n", 29  1 - calculate sin integral,

     ) = 29

     write(1, "\t2 - calculate cos derivative,\n", 31    2 - calculate cos derivative,

     ) = 31

     write(1, "\t3 - exit.\n", 11     3 - exit.

     )            = 11

     write(1, "================================"...,
44============================================

     ) = 44

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

     read(0, 1

     "1\n", 1024)                    = 2

     read(0, 16

     "16\n", 1024)                   = 3
```

```
read(0, 28
"28\n", 1024)                   = 3
write(1, "GCF is: 4\n", 10GCF is: 4
)            = 10
read(0, 0
"0\n", 1024)                    = 2
write(1, "Algo switched to second\n", 24Algo switched to second
) = 24
read(0, 16
"16\n", 1024)                   = 3
write(1, "Invalid command\n", 16Invalid command
)     = 16
read(0, 28
"28\n", 1024)                   = 3
write(1, "Invalid command\n", 16Invalid command
)     = 16
read(0, 1
"1\n", 1024)                    = 2
read(0, 16
"16 \n", 1024)                  = 4
read(0, 28
"28\n", 1024)                   = 3
write(1, "GCF is: 4\n", 10GCF is: 4
)            = 10
read(0, 0
"0\n", 1024)                    = 2
write(1, "Algo switched to first\n", 23Algo switched to first
) = 23
read(0, 2
"2\n", 1024)                    = 2
read(0, 4
"4\n", 1024)                    = 2
write(1, "\n", 1
)                         = 1
write(1, "Enter an array :", 16Enter an array :)    = 16
```

```
read(0, 1
"1\n", 1024)                    = 2
read(0, 8
"8\n", 1024)                    = 2
read(0, -8
"-8\n", 1024)                   = 3
read(0, 9
"9\n", 1024)                    = 2
write(1, "\n", 1
)                           = 1
write(1, "Sorted: -8 1 8 9 \n", 18Sorted: -8 1 8 9
)     = 18
read(0, 3
"3\n", 1024)                    = 2
--- SIGSEGV {si_signo=SIGSEGV, si_code=SEGV_MAPERR, si_addr=0x1fffffff0} ---
+++ killed by SIGSEGV (core dumped) +++
```

Вывод

В ходе выполнения данной работы я узнала про динамические библиотеки и их преимущество перед статическими, научилась использовать их в яп C++. При использовании динамических библиотек мы будем использовать 1 экземпляр библиотеки для многих программ, что значительно экономит память, а также позволяет упростить поддержку кода, легко динамически обновлять. Использование таких библиотек делает программирование модульным и более гибким.