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: М8О-209Б-23

Студент: Борисов Д.С.

Преподаватель: Миронов Е.С.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата: 18.12.24

Москва, 2024

**Постановка задачи**

**Вариант 8.**

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

1. Во время компиляции (на этапе «линковки»/linking)
2. 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;
* Тестовая программа (*программа №1*), которая используют одну из библиотек, используя информацию полученные на этапе компиляции;
* Тестовая программа (*программа №2*), которая загружает библиотеки, используя только их относительные пути и контракты.

Провести анализ двух типов использования библиотек.

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Сигнатура | Реализация 1 | Реализация 2 |
| Рассчет интеграла функции sin(x) на отрезке [A, B] с шагом e | Float SinIntegral(float A, float B, float e) | Подсчет интеграла методом прямоугольников. | Подсчет интеграла методом трапеций. |
| Отсортировать целочисленный массив | Int \* Sort(int \* array) | Пузырьковая сортировка | Сортировка Хоара |

**Общий метод и алгоритм решения**

Использованные системные вызовы:

● void\* dlopen(const char\* filename, int flag); – загружает динамическую библиотеку в память.  
● void\* dlsym(void\* handle, const char\* symbol); – возвращает указатель на функцию или переменную из загруженной библиотеки.  
● int dlclose(void\* handle); – освобождает ресурсы, связанные с загруженной библиотекой.

В рамках лабораторной работы были выполнены следующие основные этапы:

1. **Создание динамических библиотек:**
   * Реализованы две функции:
     + **SinIntegral:** Рассчитывает интеграл функции sin(x) на заданном отрезке [A, B] с шагом e методом прямоугольников и методом трапеций.
     + **Sort:** Сортирует целочисленный массив с использованием пузырьковой сортировки и быстрой сортировки (алгоритм Хоара).
   * Для каждой реализации функции была создана отдельная динамическая библиотека:
     + libSinIntegral\_Rect.so и libSinIntegral\_Trapezoid.so для функции SinIntegral.
     + libSort\_Bubble.so и libSort\_Quick.so для функции Sort.
2. **Разработка тестовых программ:**
   * **TestProgram1:** Использует динамические библиотеки на этапе компиляции (линковки).
   * **TestProgram2:** Загружает динамические библиотеки во время исполнения программы с помощью функций dlopen, dlsym и dlclose.
3. **Автоматизация сборки с помощью Makefile:**
   * Создан Makefile для упрощения процесса компиляции и сборки библиотек и тестовых программ.
   * Включены настройки RPATH для автоматического поиска динамических библиотек без необходимости ручной установки переменной окружения LD\_LIBRARY\_PATH.

**Makefile**

CXX = g++

CXXFLAGS = -std=c++11 -Wall -Wextra -fPIC -shared -O2

APPFLAGS = -std=c++11 -Wall -Wextra -O2

LDFLAGS = -ldl

# RPATH настройки

RPATH = -Wl,-rpath,\$$ORIGIN

# Библиотеки

LIBS = libSinIntegral\_Rect.so libSinIntegral\_Trapezoid.so libSort\_Bubble.so libSort\_Quick.so

all: $(LIBS) TestProgram1 TestProgram2

libSinIntegral\_Rect.so: SinIntegral\_Rect.cpp

    $(CXX) $(CXXFLAGS) -o $@ $<

libSinIntegral\_Trapezoid.so: SinIntegral\_Trapezoid.cpp

    $(CXX) $(CXXFLAGS) -o $@ $<

libSort\_Bubble.so: Sort\_Bubble.cpp

    $(CXX) $(CXXFLAGS) -o $@ $<

libSort\_Quick.so: Sort\_Quick.cpp

    $(CXX) $(CXXFLAGS) -o $@ Sort\_Quick.cpp

TestProgram1: TestProgram1.cpp

    $(CXX) $(APPFLAGS) -o $@ TestProgram1.cpp -L. -lSinIntegral\_Rect -lSort\_Bubble $(RPATH)

TestProgram2: TestProgram2.cpp

    $(CXX) $(APPFLAGS) -o $@ TestProgram2.cpp $(LDFLAGS) $(RPATH)

clean:

    rm -f $(LIBS) TestProgram1 TestProgram2

**Описание используемых флагов**

В Makefile используются следующие флаги компиляции и линковки:

* **Компилятор и стандарты:**
  + CXX = g++: Используется компилятор g++ для C++.
  + -std=c++11: Устанавливает стандарт C++11 для компиляции, обеспечивая доступ к современным возможностям языка.
* **Предупреждения и оптимизации:**
  + -Wall -Wextra: Включают все стандартные предупреждения и дополнительные предупреждения для выявления потенциальных ошибок в коде.
  + -O2: Включает уровень оптимизации O2 для повышения производительности создаваемых бинарных файлов.
* **Создание динамических библиотек:**
  + -fPIC: Генерирует позиционно-независимый код, необходимый для создания динамических библиотек.
  + -shared: Указывает компилятору создать динамическую библиотеку.
* **Настройка пути поиска библиотек:**
  + RPATH = -Wl,-rpath,\$$ORIGIN: Встраивает путь поиска библиотек непосредственно в исполняемый файл. $ORIGIN указывает на директорию, содержащую исполняемый файл, что позволяет динамическому загрузчику находить библиотеки без необходимости установки LD\_LIBRARY\_PATH.
* **Линковка с системными библиотеками:**
  + -ldl: Линкует программу с библиотекой dl, необходимой для динамической загрузки библиотек во время исполнения (используется в TestProgram2).

**Код программы**

**SinIntegral\_Rect.cpp**

|  |
| --- |
| #include <cmath> #include <stdexcept>  extern "C" float SinIntegral(float A, float B, float e) {  if (e <= 0.0f) {  throw std::invalid\_argument("Шаг интегрирования должен быть положительным");  }  float sum = 0.0f;  for (float x = A; x < B; x += e) {  sum += std::sin(x) \* e;  }  return sum; } |

**SinIntegral\_Trapezoid.cpp**

|  |
| --- |
| #include <cmath> #include <stdexcept>  extern "C" float SinIntegral(float A, float B, float e) {  if (e <= 0.0f) {  throw std::invalid\_argument("Шаг интегрирования должен быть положительным");  }  float sum = 0.0f;  for (float x = A; x < B; x += e) {  float x\_next = x + e;  if (x\_next > B) x\_next = B;  sum += (std::sin(x) + std::sin(x\_next)) \* (x\_next - x) / 2.0f;  }  return sum; } |

**Sort\_Bubble.cpp**

|  |
| --- |
| #include <algorithm> #include <stdexcept>  extern "C" int\* Sort(int\* array, int size) {  if (!array) {  throw std::invalid\_argument("Массив не может быть null");  }  if (size <= 0) {  throw std::invalid\_argument("Размер массива должен быть положительным");  }  for(int i = 0; i < size-1; ++i) {  for(int j = 0; j < size-i-1; ++j) {  if(array[j] > array[j+1]) {  std::swap(array[j], array[j+1]);  }  }  }  return array; } |

**Sort\_Quick.cpp**

|  |
| --- |
| #include <algorithm> #include <stdexcept>  extern "C" int\* Sort(int\* array, int size) {  if (!array) {  throw std::invalid\_argument("Массив не может быть null");  }  if (size <= 0) {  throw std::invalid\_argument("Размер массива должен быть положительным");  }  std::sort(array, array + size);  return array; } |

**TestProgram1.cpp**

|  |
| --- |
| #include <iostream> #include <stdexcept>  extern "C" float SinIntegral(float A, float B, float e); extern "C" int\* Sort(int\* array, int size);  int main() {  std::string command;  while (true) {  std::cout << "Введите команду (0 - переключение не поддерживается, 1 - интеграл, 2 - сортировка): ";  if (!(std::cin >> command)) break;  try {  if (command == "0") {  std::cout << "Переключение реализации не поддерживается в программе №1.\n";  }  else if (command == "1") {  float A, B, e;  std::cout << "Введите A, B, e: ";  std::cin >> A >> B >> e;  float result = SinIntegral(A, B, e);  std::cout << "Интеграл: " << result << "\n";  }  else if (command == "2") {  int size;  std::cout << "Введите размер массива: ";  std::cin >> size;  if (size <= 0) {  std::cerr << "Размер массива должен быть положительным.\n";  continue;  }  int\* array = new int[size];  std::cout << "Введите элементы массива: ";  for(int i = 0; i < size; ++i) {  std::cin >> array[i];  }  int\* sorted = Sort(array, size);  std::cout << "Отсортированный массив: ";  for(int i = 0; i < size; ++i) {  std::cout << sorted[i] << " ";  }  std::cout << "\n";  delete[] array;  }  else {  std::cout << "Неверная команда.\n";  }  } catch (const std::exception& ex) {  std::cerr << "Ошибка: " << ex.what() << "\n";  }  }  return 0; } |

**TestProgram2.cpp**

|  |
| --- |
| #include <iostream> #include <dlfcn.h> #include <stdexcept>  typedef float (\*SinIntegralFunc)(float, float, float); typedef int\* (\*SortFunc)(int\*, int);  int main() {  const char\* libSinPaths[2] = {"./libSinIntegral\_Rect.so", "./libSinIntegral\_Trapezoid.so"};  const char\* libSortPaths[2] = {"./libSort\_Bubble.so", "./libSort\_Quick.so"};  int currentLibIndex = 0;   void\* handleSin = dlopen(libSinPaths[currentLibIndex], RTLD\_LAZY);  if (!handleSin) {  std::cerr << "Не удалось загрузить библиотеку SinIntegral: " << dlerror() << "\n";  return 1;  }   void\* handleSort = dlopen(libSortPaths[currentLibIndex], RTLD\_LAZY);  if (!handleSort) {  std::cerr << "Не удалось загрузить библиотеку Sort: " << dlerror() << "\n";  dlclose(handleSin);  return 1;  }   // Получение адресов функций  SinIntegralFunc SinIntegral = (SinIntegralFunc)dlsym(handleSin, "SinIntegral");  SortFunc Sort = (SortFunc)dlsym(handleSort, "Sort");   if (!SinIntegral || !Sort) {  std::cerr << "Не удалось найти функции в библиотеках.\n";  dlclose(handleSin);  dlclose(handleSort);  return 1;  }   std::string command;  while (true) {  std::cout << "Введите команду(0 - переключение, 1 - интеграл, 2 - сортировка): ";  if (!(std::cin >> command)) break;  try {  if (command == "0") {  // Переключение реализации библиотек  dlclose(handleSin);  dlclose(handleSort);  currentLibIndex = 1 - currentLibIndex; // Переключение между 0 и 1  handleSin = dlopen(libSinPaths[currentLibIndex], RTLD\_LAZY);  handleSort = dlopen(libSortPaths[currentLibIndex], RTLD\_LAZY);  if (!handleSin || !handleSort) {  std::cerr << "Не удалось переключить библиотеки: " << dlerror() << "\n";  return 1;  }  SinIntegral = (SinIntegralFunc)dlsym(handleSin, "SinIntegral");  Sort = (SortFunc)dlsym(handleSort, "Sort");  if (!SinIntegral || !Sort) {  std::cerr << "Не удалось найти функции после переключения.\n";  dlclose(handleSin);  dlclose(handleSort);  return 1;  }  std::cout << "Реализации переключены.\n";  }  else if (command == "1") {  float A, B, e;  std::cout << "Введите A, B, e: ";  std::cin >> A >> B >> e;  float result = SinIntegral(A, B, e);  std::cout << "Интеграл: " << result << "\n";  }  else if (command == "2") {  int size;  std::cout << "Введите размер массива: ";  std::cin >> size;  if (size <= 0) {  std::cerr << "Размер массива должен быть положительным.\n";  continue;  }  int\* array = new int[size];  std::cout << "Введите элементы массива: ";  for(int i = 0; i < size; ++i) {  std::cin >> array[i];  }  int\* sorted = Sort(array, size);  std::cout << "Отсортированный массив: ";  for(int i = 0; i < size; ++i) {  std::cout << sorted[i] << " ";  }  std::cout << "\n";  delete[] array;  }  else {  std::cout << "Неверная команда.\n";  }  } catch (const std::exception& ex) {  std::cerr << "Ошибка: " << ex.what() << "\n";  }  }   dlclose(handleSin);  dlclose(handleSort);  return 0; } |

**Протокол работы программы**

|  |
| --- |
| ➜ src git:(main) ✗ ./TestProgram2  Введите команду(0 - переключение, 1 - интеграл, 2 - сортировка): 0  Реализации переключены.  Введите команду(0 - переключение, 1 - интеграл, 2 - сортировка): 1  Введите A, B, e: 0 3.1416 0.001  Интеграл: 2  Введите команду(0 - переключение, 1 - интеграл, 2 - сортировка): 2  Введите размер массива: 4  Введите элементы массива: 4 3 2 1  Отсортированный массив: 1 2 3 4 |

|  |
| --- |
| **1507 execve("./TestProgram1", ["./TestProgram1"], 0x7ffd16befdc8 /\* 26 vars \*/) = 0 1507 brk(NULL) = 0x2c4f5000 1507 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe146add000 1507 readlink("/proc/self/exe", "/workspaces/os\_base/lab4/src/Tes"..., 4096) = 41 1507 access("/etc/ld.so.preload", R\_OK) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/glibc-hwcaps/x86-64-v4/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/glibc-hwcaps/x86-64-v4", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/glibc-hwcaps/x86-64-v3/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/glibc-hwcaps/x86-64-v3", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/glibc-hwcaps/x86-64-v2/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/glibc-hwcaps/x86-64-v2", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/tls/x86\_64/x86\_64/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/tls/x86\_64/x86\_64", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/tls/x86\_64/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/tls/x86\_64", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/tls/x86\_64/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/tls/x86\_64", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/tls/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/tls", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/x86\_64/x86\_64/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/x86\_64/x86\_64", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/x86\_64/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/x86\_64", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/x86\_64/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 newfstatat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/x86\_64", 0x7fff9bc615b0, 0) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/libSinIntegral\_Rect.so", O\_RDONLY|O\_CLOEXEC) = 3 1507 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 1507 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=16048, ...}, AT\_EMPTY\_PATH) = 0 1507 mmap(NULL, 16456, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe146ad8000 1507 mmap(0x7fe146ad9000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7fe146ad9000 1507 mmap(0x7fe146ada000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fe146ada000 1507 mmap(0x7fe146adb000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fe146adb000 1507 close(3) = 0 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/libSort\_Bubble.so", O\_RDONLY|O\_CLOEXEC) = 3 1507 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 1507 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=15984, ...}, AT\_EMPTY\_PATH) = 0 1507 mmap(NULL, 16448, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe146ad3000 1507 mmap(0x7fe146ad4000, 4096, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1000) = 0x7fe146ad4000 1507 mmap(0x7fe146ad5000, 4096, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fe146ad5000 1507 mmap(0x7fe146ad6000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2000) = 0x7fe146ad6000 1507 close(3) = 0 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/etc/ld.so.cache", O\_RDONLY|O\_CLOEXEC) = 3 1507 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=25258, ...}, AT\_EMPTY\_PATH) = 0 1507 mmap(NULL, 25258, PROT\_READ, MAP\_PRIVATE, 3, 0) = 0x7fe146acc000 1507 close(3) = 0 1507 openat(AT\_FDCWD, "/usr/local/lib64/libstdc++.so.6", O\_RDONLY|O\_CLOEXEC) = 3 1507 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 1507 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=2530008, ...}, AT\_EMPTY\_PATH) = 0 1507 mmap(NULL, 2543808, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe14685e000 1507 mmap(0x7fe146903000, 1216512, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xa5000) = 0x7fe146903000 1507 mmap(0x7fe146a2c000, 581632, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ce000) = 0x7fe146a2c000 1507 mmap(0x7fe146aba000, 57344, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x25c000) = 0x7fe146aba000 1507 mmap(0x7fe146ac8000, 12480, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe146ac8000 1507 close(3) = 0 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/libm.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libm.so.6", O\_RDONLY|O\_CLOEXEC) = 3 1507 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 1507 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=907784, ...}, AT\_EMPTY\_PATH) = 0 1507 mmap(NULL, 909560, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe14677f000 1507 mmap(0x7fe14678f000, 471040, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x10000) = 0x7fe14678f000 1507 mmap(0x7fe146802000, 368640, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x83000) = 0x7fe146802000 1507 mmap(0x7fe14685c000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0xdc000) = 0x7fe14685c000 1507 close(3) = 0 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/usr/local/lib64/libgcc\_s.so.1", O\_RDONLY|O\_CLOEXEC) = 3 1507 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 1507 newfstatat(3, "", {st\_mode=S\_IFREG|0644, st\_size=906528, ...}, AT\_EMPTY\_PATH) = 0 1507 mmap(NULL, 181160, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe146752000 1507 mmap(0x7fe146756000, 143360, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x4000) = 0x7fe146756000 1507 mmap(0x7fe146779000, 16384, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x27000) = 0x7fe146779000 1507 mmap(0x7fe14677d000, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x2b000) = 0x7fe14677d000 1507 close(3) = 0 1507 openat(AT\_FDCWD, "/workspaces/os\_base/lab4/src/libc.so.6", O\_RDONLY|O\_CLOEXEC) = -1 ENOENT (No such file or directory) 1507 openat(AT\_FDCWD, "/lib/x86\_64-linux-gnu/libc.so.6", O\_RDONLY|O\_CLOEXEC) = 3 1507 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20t\2\0\0\0\0\0"..., 832) = 832 1507 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 1507 newfstatat(3, "", {st\_mode=S\_IFREG|0755, st\_size=1922136, ...}, AT\_EMPTY\_PATH) = 0 1507 mmap(NULL, 8192, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe146750000 1507 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 1507 mmap(NULL, 1970000, PROT\_READ, MAP\_PRIVATE|MAP\_DENYWRITE, 3, 0) = 0x7fe14656f000 1507 mmap(0x7fe146595000, 1396736, PROT\_READ|PROT\_EXEC, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x26000) = 0x7fe146595000 1507 mmap(0x7fe1466ea000, 339968, PROT\_READ, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x17b000) = 0x7fe1466ea000 1507 mmap(0x7fe14673d000, 24576, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_DENYWRITE, 3, 0x1ce000) = 0x7fe14673d000 1507 mmap(0x7fe146743000, 53072, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_FIXED|MAP\_ANONYMOUS, -1, 0) = 0x7fe146743000 1507 close(3) = 0 1507 mmap(NULL, 12288, PROT\_READ|PROT\_WRITE, MAP\_PRIVATE|MAP\_ANONYMOUS, -1, 0) = 0x7fe14656c000 1507 arch\_prctl(ARCH\_SET\_FS, 0x7fe14656c740) = 0 1507 set\_tid\_address(0x7fe14656ca10) = 1507 1507 set\_robust\_list(0x7fe14656ca20, 24) = 0 1507 rseq(0x7fe14656d060, 0x20, 0, 0x53053053) = 0 1507 mprotect(0x7fe14673d000, 16384, PROT\_READ) = 0 1507 mprotect(0x7fe14677d000, 4096, PROT\_READ) = 0 1507 mprotect(0x7fe14685c000, 4096, PROT\_READ) = 0 1507 mprotect(0x7fe146aba000, 45056, PROT\_READ) = 0 1507 mprotect(0x7fe146ad6000, 4096, PROT\_READ) = 0 1507 mprotect(0x7fe146adb000, 4096, PROT\_READ) = 0 1507 mprotect(0x403000, 4096, PROT\_READ) = 0 1507 mprotect(0x7fe146b0f000, 8192, PROT\_READ) = 0 1507 prlimit64(0, RLIMIT\_STACK, NULL, {rlim\_cur=8192\*1024, rlim\_max=RLIM64\_INFINITY}) = 0 1507 munmap(0x7fe146acc000, 25258) = 0 1507 futex(0x7fe146ac873c, FUTEX\_WAKE\_PRIVATE, 2147483647) = 0 1507 getrandom("\xd6\x56\xa4\x40\xb7\xf0\xd4\x1a", 8, GRND\_NONBLOCK) = 8 1507 brk(NULL) = 0x2c4f5000 1507 brk(0x2c516000) = 0x2c516000 1507 newfstatat(1, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0 1507 write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265 \320\272\320\276\320\274\320\260\320\275\320\264\321\203 (0"..., 160) = 160 1507 newfstatat(0, "", {st\_mode=S\_IFCHR|0620, st\_rdev=makedev(0x88, 0), ...}, AT\_EMPTY\_PATH) = 0 1507 read(0, 0x2c5076c0, 1024) = ? ERESTARTSYS (To be restarted if SA\_RESTART is set) 1507 --- SIGINT {si\_signo=SIGINT, si\_code=SI\_KERNEL} --- 1507 +++ killed by SIGINT +++** |

**Вывод**

В ходе выполнения лабораторной работы были успешно созданы и использованы динамические библиотеки для реализации заданных функций. Разработанные тестовые программы демонстрируют два подхода к использованию динамических библиотек: статическую линковку во время компиляции и динамическую загрузку во время исполнения. Использование RPATH в Makefile позволило упростить процесс поиска библиотек, избавив от необходимости ручной настройки переменной окружения LD\_LIBRARY\_PATH. Анализ с помощью strace подтвердил корректность загрузки библиотек в обоих подходах. Работа показала преимущества динамической загрузки, такие как гибкость и возможность изменения функциональности без пересборки программ, а также подчеркнула важность правильной настройки путей поиска библиотек для обеспечения их корректной загрузки.