Кафедра прикладной математики и кибернетики

Выполнили:

Студенты 3 курса группы ИП-111  
Корнилов А.А.,  
Попов М.И.,

Толкач А.А.

Проверил:

Профессор кафедры ПМиК  
Малков Е.А.

ЛАБОРАТОРНАЯ РАБОТА №7

По дисциплине: «Операционные системы»

Сибирский государственный университет  
телекоммуникаций и информатики

Новосибирск, 2023

Министерство цифрового развития, связи  
и массовых коммуникаций Российской Федерации

**Задание:** разработайте программу с динамической загрузкой библиотеки совместного доступа, созданную в лабораторной 5. Проконтролируйте загрузку и выгрузку библиотеки, используя файл /proc/<идентификатор процесса>/maps

**Цель:** знакомство со структурой адресного пространства процесса

**Выполнение работы:**

Для выполнения лабораторной работы была взята программа №5 лабораторной работы (листинг 1).

|  |
| --- |
| #include <stdio.h>  #include "lab05.h"  int main() {  system("clear");  int choice;  while (1) {  printf("\nМеню:\n");  printf("1. Добавить запись\n");  printf("2. Редактировать запись\n");  printf("3. Удалить запись\n");  printf("4. Поиск записи\n");  printf("5. Просмотр всех записей\n");  printf("6. Экспорт базы данных в файл\n");  printf("7. Импорт базы данных из файла\n");  printf("0. Выйти\n");  printf("Выберите действие (1-0): ");  scanf("%d", &choice);  switch (choice) {  case 1:  addRecord();  break;  case 2:  editRecord();  break;  case 3:  deleteRecord();  break;  case 4:  searchRecord();  break;  case 5:  viewAllRecords();  break;  case 6:  exportDatabaseToFile();  break;  case 7:  importDatabaseFromFile();  break;  case 0:  printf("Выход из программы.\n");  return 0;  default:  printf("Неверный выбор. Попробуйте снова.\n");  }  }  return 0;  } |

Листинг 1 – программа lab05.c

Для динамической загрузки библиотек необходимо использовать библиотеку <dlfcn.h>, необходимо открыть .so файл с флагом «RTLD\_LAZY» для разрешения неопределенных символов в коде.

|  |
| --- |
| void\* handle = dlopen("./liblab05.so", RTLD\_LAZY);  if (!handle) {  fprintf(stderr, " нету .so файла, проверьте название файла 🦣 %s\n", dlerror());  return 1;  }; |

Далее необходимо вызвать для каждого выбора в меню вызывать необходимую функцию из библиотеки, для этого создаем указатель на не типизированную функцию и в него заносим в помощью «dlsym» указывавшем куда мы открыли .so файл и какую функцию хотим вызвать. Так делаем для всего меню.

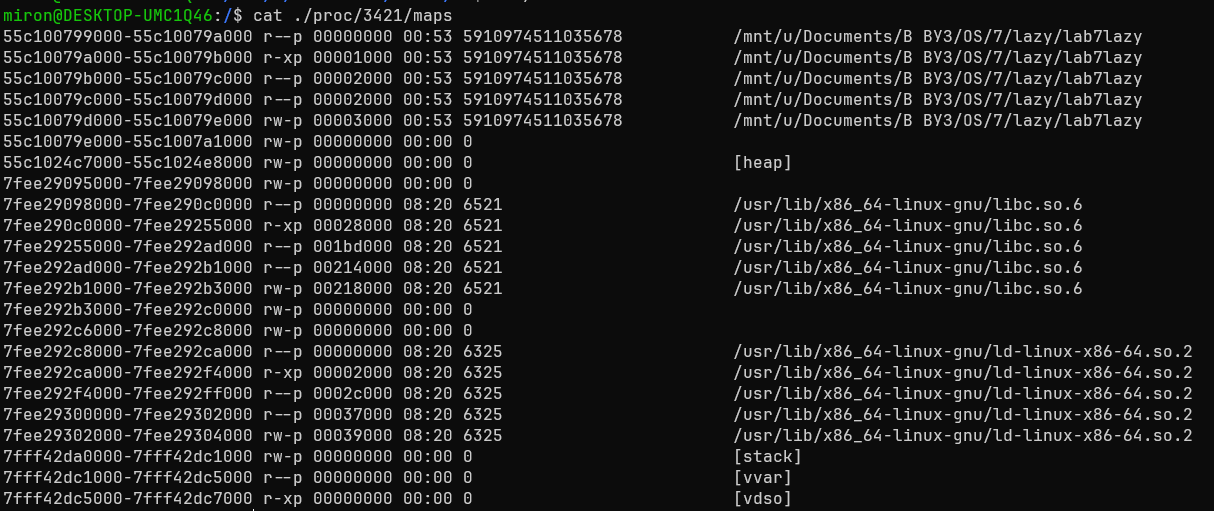
|  |
| --- |
| void\* (\*func1)();  /\*------сокращение кода функции------\*/  switch (choice) {  case 1:  func1 = dlsym(handle, "addRecord");  (\*func1)();  break; |

В конце не забываем закрывать библиотеку «dlclose(handle);».

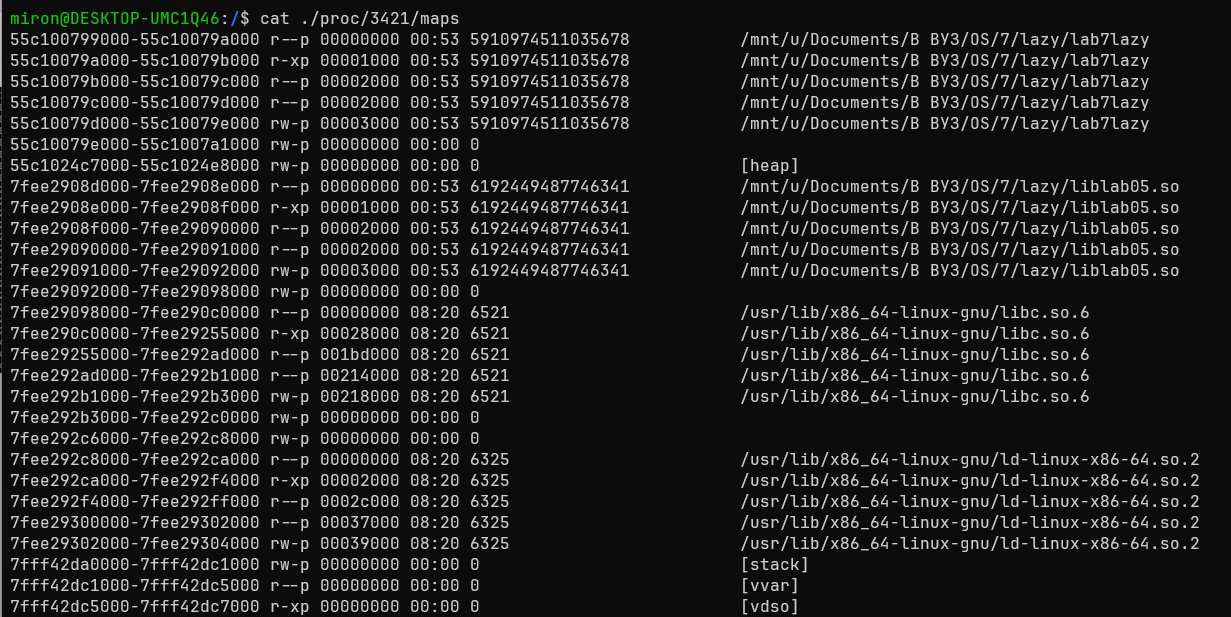
При компиляции и компоновке не забываем указать флаг библиотеки -ldl.

|  |
| --- |
| .PHONY: all  all:  gcc -Wall -ldl lab07s.c -o lab7lazy |

Проверяем загрузку и выгрузку библиотеки с помощью maps файла



До загрузки .so файла



После загрузки

Итоговый листинг программы:

|  |
| --- |
| #include <stdio.h>  #include <dlfcn.h>  //#include "lab07.h"  struct Person {  char name[50];  int age;  char address[100];  };  struct Person database[100];  int databaseSize = 0;  char\* menu = "Меню";  char\* n1 = "1. Добавить запись";  char\* n2 = "2. Редактировать запись";  char\* n3 = "3. Удалить запись";  char\* n4 = "4. Поиск записи";  char\* n5 = "5. Просмотр всех записей";  char\* n6 = "6. Экспорт базы данных в файл";  char\* n7 = "7. Импорт базы данных из файла";  char\* n0 = "0. Выйти";  char\* action = "Выберите действие (1-0): ";  char\* exit\_text = "Выход из программы.";  char\* wrong\_text = "Неверный выбор. Попробуйте снова.";  int main() {  system("clear");  printf("%s\n", "для загрузки .so нажмите Enter");  getchar();  void\* handle = dlopen("./liblab05.so", RTLD\_LAZY);  if (!handle) {  fprintf(stderr, " нету .so файла, проверьте название файла 🦣 %s\n", dlerror());  return 1;  };  void\* (\*func1)();  int choice;  while (1) {  printf("\n%s\n",menu);  printf("%s\n", n1);  printf("%s\n", n2);  printf("%s\n", n3);  printf("%s\n", n4);  printf("%s\n", n5);  printf("%s\n", n6);  printf("%s\n", n7);  printf("%s\n", n0);  printf("%s\n", action);  scanf("%d", &choice);  switch (choice) {  case 1:  func1 = dlsym(handle, "addRecord");  (\*func1)();  break;  case 2:  func1 = dlsym(handle, "editRecord");  (\*func1)();  break;  case 3:  func1 = dlsym(handle, "deleteRecord");  (\*func1)();  break;  case 4:  func1 = dlsym(handle, "searchRecord");  (\*func1)();  break;  case 5:  func1 = dlsym(handle, "viewAllRecords");  (\*func1)();  break;  case 6:  func1 = dlsym(handle, "exportDatabaseToFile");  (\*func1)();  break;  case 7:  func1 = dlsym(handle, "importDatabaseFromFile");  (\*func1)();  break;  case 0:  printf("%s\n", exit\_text);  dlclose(handle);  getchar();  return 0;  default:  printf("%s\n", wrong\_text);  }  }  return 0;  } |