

Московский Авиационный Институт
(Национальный Исследовательский Университет)
Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

**Лабораторная работа №5 по курсу
«Операционные системы»**

Студент: Рокотянский А.Е.
Группа: М8О-201Б-21
Вариант: 36
Преподаватель: Миронов Евгений Сергеевич
Оценка: _____
Дата: _____
Подпись: _____

Москва, 2022

Содержание

1. Репозиторий
2. Постановка задачи
3. Общие сведения о программе
4. Общий метод и алгоритм решения
5. Исходный код
6. Демонстрация работы программы
7. Выводы

Репозиторий

<https://github.com/Sly-al/OS-labs>

Постановка задачи

Цель работы

Целью является приобретение практических навыков в:

- Создание динамических библиотек
- Создание программ, которые используют функции динамических библиотек
- Работа со сборочной системой

Задание

Требуется создать динамические библиотеки, которые реализуют определенный 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 использует одну из библиотек, используя знания полученные на этапе компиляции;
- Тестовая программа (программа №2), которая загружает библиотеки, используя только их местоположение и контракты.

Провести анализ двух типов использования библиотек.

Общие сведения о программе

Динамические библиотеки с функциями компилируются из файлов func1.c и func2.c соответственно. Основные программы компилируются из файлов prog1.c и prog2.c с использованием заголовочного файла lab5.h (описание функций из динамических библиотек). Во второй программе используются библиотечные вызовы dlopen, dlsym, dlclose – открытие динамической библиотеки, получение функции и закрытие соответственно.

Общий метод и алгоритм решения

Во второй программе при получении 0 на вход происходит следующее: закрывается текущая динамическая библиотека, открывается новая (имена библиотек берутся из массива размера N по формуле $(n + 1) \% N$, где n – текущая библиотека) и из неё загружаются функции вместо функций старой.

Исходный код

func1.c

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "lab5.h"

void IntSwap(int* a, int* b) {
    int t = *a;
    *a = *b;
    *b = t;
}

void CharSwap(char* a, char* b){
    char temp = *a;
    *a = *b;
    *b = temp;
}

char* Translation(long x){
    char* str = malloc(sizeof(int)*8);
    int p = 0;

    while(x > 0){
        str[p] = '0' + x % 2;
        x = x / 2;
        p++;
    }

    str[p] = '\\0';
    int i = 0;

    while(i < p){
        p--;
        CharSwap(str + i, str + p);
        i++;
    }

    return str;
}

void Sort(int* array, unsigned long size) {
    for (unsigned long i = 0; i < size; ++i) {
        for (unsigned long j = 0; j < size - 1; ++j) {
            if (array[j] > array[j + 1]) {
                IntSwap(array + j, array + j + 1);
            }
        }
    }
}
```

func2.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "lab5.h"

int Compare(const void* a, const void* b) {
    return (*(int*)a - *(int*)b);
}
```

```

}

void CharSwap(char* a, char* b){
    char temp = *a;
    *a = *b;
    *b = temp;
}

char* Translation(long x){
    char* str = malloc(sizeof(int)*8);
    int p = 0;

    while(x > 0){
        str[p] = '0' + x % 3;
        x = x / 3;
        p++;
    }

    str[p] = '\0';
    int i = 0;

    while(i < p){
        p--;
        CharSwap(str + i, str + p);
        i++;
    }

    return str;
}

void Sort(int* array, unsigned long size) {
    qsort(array, size, sizeof(int), Compare);
}

```

prog1.c

```

#include<stdio.h>
#include <stdlib.h>
#include"lab5.h"

int main(){

    int t;
    printf("1 - translation, 2 - sort of array\n");

    while(scanf("%d", &t) != EOF){

        switch (t) {
            case 1:{

                printf("Enter x = ");
                int x;
                scanf("%d", &x);
                char* result = Translation(x);
                printf("Translation is %s\n", result);
                free(result);
                break;

            }
            case 2:{

                printf("Enter size of array = ");
                unsigned long size;

```

```

        scanf("%lu", &size);
        int* arr = malloc(size * sizeof(int));
        printf("Start input:\n");

        for (unsigned long i = 0; i < size; ++i){
            scanf("%d", arr + i);
        }

        printf("Sorted: \n");
        Sort(arr, size);

        for (unsigned long i = 0; i < size; ++i){
            printf("%d ", arr[i]);
        }

        printf("\n");
        free(arr);
        break;
    }
    default:{

        printf("1 - translation, 2 - sort\n");
        break;
    }
}

}
return 0;
}

```

prog2.c

```

#include <stdio.h>
#include <dlfcn.h>
#include <stdlib.h>
#include <lab5.h>

int main(){

    int numberOfLyb = 2;
    char* names[] = {"./libdyn1.so", "./libdyn2.so"};

    int n = 0;
    void* handle;
    handle = dlopen(names[n], RTLD_LAZY);

    if (!handle){
        printf("dlopen error\n");
        return -1;
    }

    TranslationType trans;
    SortType sort;
    #pragma GCC diagnostic push
    #pragma GCC diagnostic ignored "-Wpedantic"
    trans = (TranslationType)dlsym(handle, "Translation");
    sort = (SortType)dlsym(handle, "Sort");
    #pragma GCC diagnostic pop

    int t;
    printf("0 - switch library 1 - translation, 2 - sort of array\n");
}

```

```

while(scanf("%d", &t) != EOF){

    switch (t) {
        case 0:{

            n = (n + 1) % numberOfLyb;

            if (dlclose(handle) != 0){
                perror("dlclose error");
                return -1;
            }

            if (!(handle = dlopen(names[n], RTLD_LAZY))){
                printf("dlopen error\n");
                return -1;
            }

            #pragma GCC diagnostic push
            #pragma GCC diagnostic ignored "-Wpedantic"
            trans = (TranslationType)dlsym(handle, "Translation");
            sort = (SortType)dlsym(handle, "Sort");
            #pragma GCC diagnostic pop
            printf("Switched\n");
            break;

        }
        case 1:{

            printf("Enter x = ");
            int x;
            scanf("%d", &x);
            char* result = (*trans)(x);
            printf("Translation is %s\n", result);
            free(result);
            break;

        }
        case 2:{

            printf("Enter size of array = ");
            unsigned long size;
            scanf("%lu", &size);
            int* arr = malloc(size * sizeof(int));
            printf("Start input:\n");

            for (unsigned long i = 0; i < size; ++i){
                scanf("%d", arr + i);
            }

            printf("Sorted: \n");
            (*sort)(arr, size);

            for (unsigned long i = 0; i < size; ++i){
                printf("%d ", arr[i]);
            }

            printf("\n");
            free(arr);
            break;

        }
        default:{

            printf("0 - switch library 1 - translation, 2 - sort of array\n");
            break;

        }
    }
}

```

```
}  
return 0;  
  
}
```

Демонстрация работы программы

```
alex@alex-VirtualBox:~/Рабочий стол/OS-labs/build/lab5$ ./prog11
```

```
1 - translation, 2 - sort of array
```

```
1
```

```
Enter x = 20
```

```
Translation is 10100
```

```
2
```

```
Enter size of array = 3
```

```
Start input:
```

```
10000000
```

```
-4627
```

```
12
```

```
Sorted:
```

```
-4627 12 10000000
```

```
^C
```

```
alex@alex-VirtualBox:~/Рабочий стол/OS-labs/build/lab5$ ./prog12
```

```
1 - translation, 2 - sort of array
```

```
1
```

```
Enter x = 20
```

```
Translation is 202
```

```
2
```

```
Enter size of array = 3
```

```
Start input:
```

```
1000
```

```
-2342
```

```
152632
```

```
Sorted:
```

```
-2342 1000 152632
```

```
alex@alex-VirtualBox:~/Рабочий стол/OS-labs/build/lab5$ ./prog2
```

```
0 - switch library 1 - translation, 2 - sort of array
```

```
1
```

```
Enter x = 20
```

```
Translation is 10100
```

```
2
```

```
Enter size of array = 3
```



```
Start input:
3723
-183874
1213
Sorted:
-183874 1213 3723
0
Swithced
1
Enter x = 20
Translation is 202
2
Enter size of array = 3
Start input:
283737
-2482484
121
Sorted:
-2482484 121 283737
```

Выводы

Составлены и отлажены программы на языке Си, осуществляющие работу с динамическими библиотеками. Одна подключает библиотеки на этапе линковки, другая во время работы по средствам ОС.