

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-211БВ-24

Студент: Мутагиров Т.Р.

Преподаватель: Бахарев В.Д.

Оценка: \_\_\_\_\_

Дата: 26.11.25

Москва, 2025

# Постановка задачи

**Вариант 29.**

**Функция 1:**

Расчет значения числа  $\pi$  при заданной длине ряда ( $k$ ):

Сигнатура функции: float pi(int k);

- Реализация №1: Ряд Лейбница
- Реализация №2: Формула Валлиса

**Функция 2:**

Перевод числа  $x$  из десятичной системы счисления в другую:

Сигнатура функции: char \*convert(int x);

- Реализация №1: Перевод в двоичную
- Реализация №2: Перевод в троичную

## Общий метод и алгоритм решения

- void \*dlopen(const char \*filename, int flags); – загружает в память и открывает динамическую библиотеку.
- void \*dlsym(void \*handle, const char \*symbol); – возвращает адрес функции или переменной из загруженной библиотеки.
- int dlclose(void \*handle); – выгружает из памяти ранее загруженную динамическую библиотеку.

Я создал две библиотеки, которые реализуют два контракта двух функций. Также я сделал два вида подключения этой библиотеки в программу. Первый способ заключается в том, что подключаю библиотеку во время линковки программы. Второй способ заключается в том, что я подключаю библиотеку в runtime через интерфейс ОС для работы с динамическими библиотеками.

## Код программы

Lib1.c

```
#include "lib.h"

// в двоичную
char *convert(int x) {
    char *res = (char *)calloc(34, 1);
    char tmp[32];
    char *ptr = tmp, *ptr_res = res;
    if(!x) *ptr_res++ = '0';
    int neg = 0;
    if(x < 0) {
        neg = 1;
        x = -x;
    }
    while(x) {
        *ptr++ = (x % 2) + '0';
        x /= 2;
    }
}
```

```

    if(neg) *ptr_res++ = '-';
    while(ptr != tmp) {
        *ptr_res++ = *--ptr;
    }
    return res;
}

float pi(int k) {
    float res = 0;
    for(int i = 0; i < k; i++) {
        res += (i & 1 ? -1.: 1.) / (2 * i + 1);
    }
    res *= 4;
    return res;
}

```

## Lib2.c

```

#include "lib.h"

// в троичную
char *convert(int x) {
    char *res = (char *)calloc(33, 1);
    char tmp[32];
    char *ptr = tmp, *ptr_res = res;
    if(!x) *ptr_res++ = '0';
    int neg = 0;
    if(x < 0) {
        neg = 1;
        x = -x;
    }
    while(x) {
        *ptr++ = (x % 3) + '0';
        x /= 3;
    }
    if(neg) *ptr_res++ = '-';
    while(ptr != tmp) {
        *ptr_res++ = *--ptr;
    }
    return res;
}

float pi(int k) {
    float res = 1;
    for(int i = 1; i <= k; i++) {
        res *= 4. * i * i / (4 * i * i - 1);
    }
    res *= 2;
    return res;
}

```

### Example\_static.c

```
#include <stdio.h>
#include "lib.h"

int main() {
    int choise;
    printf("Введите номер вызываемой функции а потом аргументы\nДля выхода 0\n\n");
    scanf("%d", &choise);
    while(choise) {
        int arg;
        scanf("%d", &arg);
        if(choise == 1) {
            printf("%s\n", convert(arg));
        }
        if(choise == 2) {
            printf("%f\n", pi(arg));
        }
        scanf("%d", &choise);
    }
    return 0;
}
```

### Example\_dynamic.c

```
#include <stdio.h>
#include <stdlib.h>
#include <dlfcn.h>
// #include "lib.h"

typedef float (*pi_t)(int k);
typedef char* (*conv_t)(int x);

char *paths[] = {"./libown1.so", "./libown2.so"};

pi_t pi_func;
conv_t conv_func;
void * lib = NULL;

int open_and_init(int ind) {
    if(lib) dlclose(lib);
    lib = dlopen(paths[ind], RTLD_LAZY);
    pi_func = dlsym(lib, "pi");
    conv_func = dlsym(lib, "convert");
    if(!pi_func || !conv_func) return -1;
    return 0;
}
```

```
int main() {
    int choise;
    printf("Введите номер вызываемой функции а потом аргументы\nПереключить
реализации библиотек 0\nДля выхода -1\n");
    int ind = 0;
    if(open_and_init(ind) != 0) {
        printf("Ошибка при открытии или чтении функций из библиотеки\n");
        return 1;
    }
    scanf("%d", &choise);
    while(choise != -1) {
        int arg;
        if(!choise) {
            if(open_and_init(ind = !ind) != 0) {
                printf("Ошибка при открытии или чтении функций из библиотеки\n");
                return 1;
            }
        } if(choise == 1) {
            scanf("%d", &arg);
            printf("%s\n", conv_func(arg));
        } if(choise == 2) {
            scanf("%d", &arg);
            printf("%f\n", pi_func(arg));
        }
        scanf("%d", &choise);
    }
    dlclose(lib);
    return 0;
}
```

## Протокол работы программы

● rinrow@DESKTOP-S9IRGBR:~/2sem/os/os\_labs/4\$ ./example\_dynamic

Введите номер вызываемой функции а потом аргументы

Переключить реализаций библиотек 0

Для выхода -1

1 2

10

2 500

3.139593

0

1 2

2

2 500

3.140023

-1

● rinrow@DESKTOP-S9IRGBR:~/2sem/os/os\_labs/4\$ ./example\_static

Введите номер вызываемой функции а потом аргументы

Для выхода 0

2 9

3.060034

1 9

100

2 1000

3.140807

0

## Вывод

В ходе лабораторной работы я научился создавать динамические библиотеки, производить линковку, создавать программы, использующие динамические библиотеки.