

# Лабораторная работа №4

---

по курсу "Операционные системы"

Выполнил: Юнусов Рустам М8О-310Б-22. 5 Вариант

Преподаватель: Миронов Евгений Сергеевич

---

## 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.

## Листинг программы

realization1.c

```
#include "pi.h"
#include <math.h>

float SinIntegral(float A, float B, float e) {
    float result = 0;
    for (float i = A; i <= B; i += e) {
        result += sin(i) * e;
    }
    return result;
}

float Pi(int K) {
    float result = 0;
    for (int i = 0; i <= K; i++) {
```

```

    result += (i % 2 == 0 ? 1.0 : -1.0) / (2 * i + 1);
}
return result * 4;
}

```

realization2.c

```

#include "pi.h"
#include <math.h>

float SinIntegral(float A, float B, float e) {
    float result = (sin(A) + sin(B)) / 2;
    for (float i = A + e; i <= B - e; i += e) {
        result += sin(i);
    }
    return e * result;
}

float Pi(int K) {
    float result = 1;
    for (int i = 1; i <= K; i++) {
        result *= (4.0 * i * i) / (4 * i * i - 1);
    }
    return result * 2;
}

```

using as static library

```

#include "pi.h"
#include <stdio.h>

int main() {
    int cmd;
    while (
        printf("Enter the command (1 - calculating integral of sin, 2 -
calculating Pi number): ") > 0 &&
        scanf("%d", &cmd) != EOF)
    {
        if (cmd == 1) {
            float A, B, e;
            printf("Enter the arguments (begin of the segment, end of the
segment, step): ");
            scanf("%f %f %f", &A, &B, &e);
            printf("Result: %f\n", SinIntegral(A, B, e));
        } else if (cmd == 2) {
            int K;
            printf("Enter the quantity of elements in the row: ");
            scanf("%d", &K);

```

```

        printf("Result: %f\n", Pi(K));
    } else {
        printf("Wrong command! Try again!\n");
    }
}
}

```

using as dynamic library

```

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

int main() {
    void *handler = dlopen("./librealization1.so", RTLD_LAZY);
    if (!handler) {
        fputs(dlerror(), stderr);
        exit(1);
    }
    int realization = 1;
    int cmd;
    char *error;
    while (printf("Enter the command (0 - changing realization (library),
1 - "
                "calculating integral of sin, 2 - calculating Pi number):
") > 0 &&
        scanf("%d", &cmd) != EOF) {
        if (cmd == 0) {
            if (dlclose(handler) != 0) {
                perror("dlclose");
                exit(1);
            }
            realization = (realization == 1 ? 2 : 1);
            handler = (realization == 1 ? dlopen("./librealization1.so",
RTLD_LAZY)
                : dlopen("./librealization2.so",
RTLD_LAZY));
            if (!handler) {
                fputs(dlerror(), stderr);
                exit(1);
            }
            printf("Realization was changed from realization%d to
realization%d.\n",
                (realization == 1 ? 2 : 1), realization);
        } else if (cmd == 1) {
            float (*SinIntegral)(float, float, float);
            *(float **)&SinIntegral = dlsym(handler, "SinIntegral");
            if ((error = dlerror()) != NULL) {
                fprintf(stderr, "%s\n", error);
                exit(1);
            }
        }
    }
}

```

```

    }
    float A, B, e;
    printf("Enter the arguments (begin of the segment, end of the
segment, "
        "step): ");
    scanf("%f %f %f", &A, &B, &e);
    printf("Result: %f\n", SinIntegral(A, B, e));
} else if (cmd == 2) {
    float (*Pi)(int);
    *(float **>(&Pi) = dlsym(handler, "Pi");
    if ((error = dlerror()) != NULL) {
        fprintf(stderr, "%s\n", error);
        exit(1);
    }
    int K;
    printf("Enter the quantity of elements in the row: ");
    scanf("%d", &K);
    printf("Result: %f\n", Pi(K));
} else {
    printf("Wrong command! Try again!\n");
}
}
if (dlclose(handler) != 0) {
    perror("dlclose");
    exit(1);
}
exit(0);
}

```

## Вывод

В этой лабораторной работе я создал две динамические библиотеки, каждая из которых реализует функции для вычисления интеграла от синуса и числа пи разными способами. Первая тестовая программа использует одну из библиотек на этапе компиляции, а вторая программа работает с библиотеками динамически, загружая их в память во время выполнения через интерфейс `dlopen` и `dlsym`.

Так же динамические библиотеки позволяют использовать различные функции и методы в разных языках при лишь одной реализации!