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-210БВ-24

Студент: Самойлов Д. А.

Преподаватель: Бахарев В.Д.

Оценка: _____

Дата: 21.11.25

Москва, 2025

# Постановка задачи

**Вариант 25.**

**Функция 1:** Подсчёт наибольшего общего делителя для двух натуральных

Сигнатура функции: int gcd(int a, int b);

● Реализация №1: Алгоритм Евклида

● Реализация №2: Наивный алгоритм: пытаться разделить числа на

все числа, что меньше a и b.

**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.

Две динамические библиотеки (libcustom1.so libcustom2.so) (названия задаются пользователем), которые реализуют контракты двух функций: gcd(int a, int b) для вычисления наибольшего общего делителя заданных чисел и char* convert(int x) для перевода числа x в строковое представление в конкретной системе счисления.

Первая библиотека реализует алгоритм Евклида для нахождения НОД и перевод числа в двоичную систему счисления, а вторая – наивный алгоритм и перевод в троичную систему счисления.

Я реализовал два способа использования этих библиотек:

1. static - линковка на этапе компиляции. Библиотека libcustom1.so подключается через флаг -lcustom1 при компиляции. Информация о зависимости записывается в исполняемый файл, и динамический загрузчик операционной системы автоматически загружает библиотеку при запуске. Поскольку библиотека находится не в стандартных системных путях поиска библиотек (/lib, /usr/lib), нам необходимо указать текущую директорию, сделать это мы можем через переменную окружения LD_LIBRARY_PATH=. при запуске исполняемого файла.

2. dynamic - загрузка во время выполнения. Библиотеки загружаются программно через интерфейс операционной системы для работы с динамическими библиотеками: функции dlopen(), dlsym(), dlclose(). Это позволяет переключаться

между реализациями (libcustom1.so и libcustom2.so) во время работы программы по команде пользователя, что невозможно при первом способе, так как библиотека загружается автоматически при запуске программы и не может быть заменена во время её выполнения.

# Код программы

## contracts.h

```
#ifndef CONTRACTS_H
#define CONTRACTS_H

int gcd(int a, int b);

char* convert(int x);

#endif //CONTRACTS_H
```

## interactive_input.h

```
#ifndef INTERACTIVE_INPUT_H
#define INTERACTIVE_INPUT_H

#include <stdio.h> // snprintf

#include <unistd.h>   // read, write
#include <stdlib.h>   // malloc, free, strtol
#include <string.h>   // strlen

typedef int   (*gcd_func)(int a, int b);
typedef char* (*convert_func)(int num);

static inline ssize_t putstr(const char *s) {
    return write(1, s, strlen(s));
}

static inline int read_line(char *buf, size_t size) {
    ssize_t n = read(0, buf, size - 1);
    if (n <= 0) return 0;
    buf[n] = '\0';

    // убираем \n
    char *nl = strchr(buf, '\n');
    if (nl) *nl = '\0';

    return 1;
}

static inline int read_int(int *out) {
    char buf[64];
    if (!read_line(buf, sizeof(buf))) return 0;

    char *end;
    long v = strtol(buf, &end, 10);
```

```
    if (end == buf) return 0; // не число

    *out = (int)v;
    return 1;
}
```

## library1.c

```
#include <stdlib.h>

#include "contracts.h"

// First realization of functions


void swap(int* a, int* b) {
    int tmp = *a;
    *a = *b;
    *b = tmp;
}

// Euclid's algorithm
int gcd(int a, int b) {
    while (b) {
        a = a % b;
        swap(&a, &b);
    }
    return a;
}

// Convert to binary integer system
char* convert(int x) {
    if (!x) {
        char* str = (char*)malloc(sizeof(char) + 1);
        str[0] = '0';
        str[1] = 0;
        return str;
    }
    int n = 8 * sizeof(int);
    char* str = (char*) malloc(n + 1);

    int sz = 0;
    for (int i = n - 1; i >= 0; --i) {
        if (x & (1U << i)) {
            sz = i + 1;
            break;
        }
    }
    unsigned int ux = (unsigned int)x;
    for (int i = 0; i < sz; ++i) {
        str[i] = (ux & (1 << (sz - i - 1))) > 0 ? '1' : '0';
    }
    str[sz] = 0;
    return str;
}
```

## library2.c

```
#include <stdlib.h>
```

```
#include "../include/contracts.h"

int min(int a, int b) {
    return a < b ? a : b;
}

void swap(char* a, char* b) {
    char tmp = *a;
    *a = *b;
    *b = tmp;
}

// Native algorithm
int gcd(int a, int b) {
    int mn = min(a, b);
    for (int d = mn; d > 1; --d) {
        if (a % d == 0 && b % d == 0) {
            return d;
        }
    }
    return 1;
}

// Convert to ternary system
char* convert(int x) {
    int capacity = 16;
    int size = 0;

    char* str = (char*)malloc(capacity * sizeof(char));

    if (!x) {
        str[0] = '0';
        str[1] = 0;
        return str;
    }
    int n = x > 0 ? x : -x;
    while (n) {
        if (size + 1 >= capacity) {
            capacity *= 2;
            char* new_str = (char*)realloc(str, capacity *
sizeof(char));
            if (!new_str) {
                free(str);
                return 0;
            }
            str = new_str;
        }
        str[size++] = '0' + (n % 3);
        n /= 3;
    }

    if (x < 0) {
        if (size + 1 >= capacity) {
            ++capacity;
            char* new_str = (char*)realloc(str, capacity *
sizeof(char));
            if (!new_str) {
                free(str);
                return 0;
```

```
            }
            str = new_str;
        }
        str[size++] = '-';
    }

    for (int i = 0; i < size / 2; ++i) {
        swap(&str[i], &str[size - i - 1]);
    }
    str[size] = 0;
    return str;
}
```

## test_static_library.c

```
#include "interactive_input.h"
#include "contracts.h"


int main() {
    putstr("Commands list: 1 - convert, 2 - gcd, -1 - exit\n");

    while (1) {
        putstr("Enter command: ");

        int command;
        if (!read_int(&command) || command == -1)
            break;

        switch (command) {
        case 1: {
            putstr("Enter number: ");
            int num;
            if (!read_int(&num)) break;

            char *res = convert(num);
            putstr("Converted to number system: ");
            putstr(res);
            putstr("\n");
            free(res);
            break;
        }

        case 2: {
            int a, b;

            putstr("Enter first number: ");
            if (!read_int(&a)) break;

            putstr("Enter second number: ");
            if (!read_int(&b)) break;

            char outbuf[128];
            int n = snprintf(outbuf, sizeof(outbuf),
                             "GCD of %d and %d is %d\n",
                             a, b, gcd(a, b));
            write(1, outbuf, n);
            break;
        }
```

```
        default:
            putstr("Unknown command\n");
        }
    }
    exit(EXIT_SUCCESS);
}
```

## test_dynamic_library.c

```
#include "interactive_input.h"

#include <dlfcn.h>


typedef int   (*gcd_func)(int a, int b);
typedef char* (*convert_func)(int num);

convert_func convert = NULL;

gcd_func gcd = NULL;


void open_library(void** library, const char* const* paths, size_t
size) {
    if (!library) return;

    static int id = 0;

    const char* path = paths[id++ % size];

    *library = dlopen(path, RTLD_LOCAL | RTLD_NOW);
    if (!*library) {
        const char msg[] = "Failed to load library";
        write(STDERR_FILENO, msg, sizeof(msg));
        exit(EXIT_FAILURE);
    }

    convert = dlsym(*library, "convert");
    if (!convert) {
        const char msg[] = "Failed to find convert";
        write(STDERR_FILENO, msg, sizeof(msg));
        exit(EXIT_FAILURE);
    }
    gcd = dlsym(*library, "gcd");
    if (!gcd) {
        const char msg[] = "Failed to find gcd";
        write(STDERR_FILENO, msg, sizeof(msg));
        exit(EXIT_FAILURE);
    }
}


int main(int argc, char* argv[]) {
    if (argc != 3) {
        const char msg[] = "Usage: ./test_dynamic_library
<path_to_library1> <path_to_library2>";
        write(STDERR_FILENO, msg, sizeof(msg));
        exit(EXIT_FAILURE);
```

```
    }
    char* library_path1 = argv[1];
    char* library_path2 = argv[2];

    const char* paths[] = {library_path1, library_path2};
    size_t paths_size = sizeof(paths) / sizeof(paths[0]);


    void* library = NULL;
    open_library(&library, paths, paths_size);

    putstr("Commands list: 0 - swap library, 1 - convert, 2 - gcd, -
1 - exit\n");

    while (1) {
        putstr("Enter command: ");

        int command;
        if (!read_int(&command) || command == -1)
            break;

        switch (command) {
        case 0:
        {
            dlclose(library);
            library = NULL;

            open_library(&library, paths, paths_size);
            putstr("Successfully loaded new library\n");
            break;
        }
        case 1: {
            putstr("Enter number: ");
            int num;
            if (!read_int(&num)) break;

            char *res = convert(num);
            putstr("Converted to number system: ");
            putstr(res);
            putstr("\n");
            free(res);
            break;
        }

        case 2: {
            int a, b;

            putstr("Enter first number: ");
            if (!read_int(&a)) break;

            putstr("Enter second number: ");
            if (!read_int(&b)) break;

            char outbuf[128];
            int n = snprintf(outbuf, sizeof(outbuf),
                             "GCD of %d and %d is %d\n",
                             a, b, gcd(a, b));
            write(1, outbuf, n);
            break;
        }
```

```
        default:
            putstr("Unknown command\n");
        }
    }

    dlclose(library);
    exit(EXIT_SUCCESS);
}
```

# Протокол работы программы

**Тестирование:**

```
~/projects/OS-MAI-2025/lab4/build git:[main]
gcc -o test_static -L. -lcustom1 ../test_static_library.c

~/projects/OS-MAI-2025/lab4/build git:[main]
LD_LIBRARY_PATH=. ./test_static
Commands list: 1 - convert, 2 - gcd, -1 - exit
Enter command: 1
Enter number: 7
Converted to number system: 111
Enter command: 2
Enter first number: 33
Enter second number: 88
GCD of 33 and 88 is 11
Enter command: 1
Enter number: 16
Converted to number system: 10000
Enter command: 2
Enter first number: 1000000007
Enter second number: 51857185
GCD of 1000000007 and 51857185 is 1
Enter command: -1
```

```
~/projects/OS-MAI-2025/lab4/build git:[main]
gcc -o test_static -L. -lcustom2 ../test_static_library.c

~/projects/OS-MAI-2025/lab4/build git:[main]
LD_LIBRARY_PATH=. ./test_static
Commands list: 1 - convert, 2 - gcd, -1 - exit
Enter command: 1
Enter number: 7
Converted to number system: 21
Enter command: 2
Enter first number: 33
Enter second number: 88
GCD of 33 and 88 is 11
Enter command: 1
Enter number: 26
Converted to number system: 222
Enter command: 2
Enter first number: 1000000007
Enter second number: 51857185
GCD of 1000000007 and 51857185 is 1
Enter command: -1
```

```
~/projects/OS-MAI-2025/lab4/build git:[main]
gcc -o test_dynamic -ldl ../test_dynamic_library.c

~/projects/OS-MAI-2025/lab4/build git:[main]
LD_LIBRARY_PATH=. ./test_dynamic ./libcustom1.so ./libcustom2.so
Commands list: 0 - swap library, 1 - convert, 2 - gcd, -1 - exit
Enter command: 1
Enter number: 7
Converted to number system: 111
Enter command: 2
Enter first number: 33
Enter second number: 88
GCD of 33 and 88 is 11
Enter command: 1
Enter number: 26
Converted to number system: 11010
Enter command: 0
Successfully loaded new library
Enter command: 1
Enter number: 7
Converted to number system: 21
Enter command: 1
Enter number: 26
Converted to number system: 222
Enter command: 0
Successfully loaded new library
Enter command: 1
Enter number: 7
Converted to number system: 111
Enter command: 1
Enter number: 26
Converted to number system: 11010
Enter command: -1
```

## Вывод

При выполнении лабораторной работы я отработал на практике вышеперечисленные системные вызовы из интерфейса ОС для работы с динамическими библиотеками. Разобрался в статической и динамической линковке библиотек. Сложностей не возникло.