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-209БВ-24

Студент: Галич А.П.

Преподаватель: Миронов Е.С.

Оценка: _____

Дата: 16.12.25

Москва, 2025

Постановка задачи

Вариант 30.

Создание динамических библиотек, которые используются двумя способами:

1. Во время компиляции(на этапе “линковки/linking”)
2. Во время использования программы.

- Рассчитать значение числа Пи при заданной длине ряда (K):
  1. Ряд Лейбница
  2. Формула Валлиса
- Отсортировать целочисленный массив:
  1. Пузырьковая сортировка
  2. Сортировка Хоара

Общий метод и алгоритм решения

Использованные системные вызовы

• **openat()** — открытие файлов библиотек (`.so`) при выполнении функции dlopen() для динамической загрузки.

• **read()** — чтение данных из стандартного ввода.

• **mmap()** — отображение динамических библиотек в память и выделение больших блоков памяти

• **mprotect()** — установка прав доступа к сегментам загруженных библиотек.

• **close()** — закрытие файловых дескрипторов, связанных с загруженными библиотеками, при вызове dlclose() для выгрузки библиотек.

• **write()** — вывод результатов работы программы.

• **exit_group()** — завершение процесса при выходе из программы.

Алгоритм работы:

Решение состоит из двух программ: с статической и динамической загрузкой функций P(int K) и Sort(int* array, int size)

Статическая компоновка

1. Подготовка библиотек

- Библиотека lib1.c содержит первую реализацию функций: Pi() на основе ряда Лейбница, Sort() на основе пузырьковой сортировки
- Бибилиотека lib2.c содержит вторую реализацию функций: Pi() на основе формулы Валлиса, Sort() на основе сортировки Хоара

2. Компиляций и линковка

- Программа static.c линкуется с выбранной библиотекой на этапе сборки:

  gcc static.c lib1.c -o static1 (с певрой реализацией)

  gcc static.c lib2.c -o static2 (со второй реализацией)

Динамическая загрузка

1. Инициализация

Программа dynamic при запуске:

- Загружает первую библиотеку(lib1.c) с помощью dlopen()
- Получает указатель на функцию Pi и Sort через dlsym()
- Подготавливает систему для переключения между реализациями

2. Механизм для переключения

- При вводе команды 0:
  - Выгружается текущая библиотека через dlclose()
  - Загружается альтернативная библиотека
  - Обновляются указатели на функции

Код программы

lib1.c

```c
#include <stdlib.h>

float Pi(int K) {
    float pi = 0.0;
    int sign = 1;

    for (int i = 0; i < K; i++) {
        pi += sign * 4.0 / (2 * i + 1);
        sign = -sign;
    }

    return pi;
}
```

```
int* Sort(int* array, int size) {
    int* sorted = (int*)malloc(size * sizeof(int));
    for (int i = 0; i < size; i++) {
        sorted[i] = array[i];
    }

    for (int i = 0; i < size - 1; i++) {
        for (int j = 0; j < size - 1; j++) {
            if (sorted[j] > sorted[j + 1]) {
                int temp = sorted[j];
                sorted[j] = sorted[j + 1];
                sorted[j + 1] = temp;
            }
        }
    }

    return sorted;
}
```

lib2.c

```
#include <stdlib.h>

float Pi(int K) {
    float pi = 1.0;

    for (int i = 1; i <= K; i++) {
        float num = 4.0 * i * i;
        float den = 4.0 * i * i - 1.0;
        pi *= num / den;
    }

    return pi * 2;
}

void qsort_impl(int* arr, int low, int high) {
    if (low >= high) return;

    int pivot = arr[(low + high) / 2];
    int i = low, j = high;

    while (i <= j) {
        while (arr[i] < pivot) i++;
        while (arr[j] > pivot) j--;
        if (i <= j) {
            int tmp = arr[i];
            arr[i] = arr[j];
            arr[j] = tmp;
            i++; j--;
        }
    }
```

```
    qsort_impl(arr, low, j);
    qsort_impl(arr, i, high);
}

int* Sort(int* array, int size) {
    if (size <= 0) {
        int* empty = (int*)malloc(sizeof(int));
        return empty;
    }

    int* sorted = (int*)malloc(size * sizeof(int));
    for (int i = 0; i < size; i++) {
        sorted[i] = array[i];
    }

    if (size > 1) {
        qsort_impl(sorted, 0, size - 1);
    }

    return sorted;
}
```

lib.h

```
#ifndef LIB_H
#define LIB_H

float Pi(int K);
int* Sort(int* array, int size);

#endif
```

static.c

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

int main() {
    printf("  1 K          - вычислить Pi\n");
    printf("  2 a1 a2 ... - отсортировать числа\n");
    printf("  -1           - выход\n");
    printf("Введите команду: ");

    int cmd;

    while (1) {
        if (scanf("%d", &cmd) == EOF) break;

        if (cmd == -1) {
```

```
            printf("Выход\n");
            break;
        }

        if (cmd == 1) {
            int K;
            scanf("%d", &K);
            printf("Pi = %f\n", Pi(K));
        }

        else if (cmd == 2) {
            int numbers[100];
            int count = 0;
            int num;

            while (scanf("%d", &num) == 1 && count < 100) {
                numbers[count++] = num;
                if (getchar() == '\n') break;
            }

            if (count > 0) {
                printf("Исходный: ");
                for (int i = 0; i < count; i++) printf("%d ", numbers[i]);
                printf("\n");

                int* sorted = Sort(numbers, count);

                printf("Отсортированный: ");
                for (int i = 0; i < count; i++) printf("%d ", sorted[i]);
                printf("\n");

                free(sorted);
            }
        }

        else {
            printf("нет такой команды\n");
        }
    }

    return 0;
}
```

dynamic.c

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

int main() {
    char *libs[] = {"./lib1.so", "./lib2.so"};
    int cur = 0;
```

```c
    void *handle = dlopen(libs[cur], RTLD_LAZY);
    if (!handle) {
        printf("Ошибка загрузки библиотеки: %s\n", dlerror());
        return 1;
    }

    float (*PiFunc)(int) = dlsym(handle, "Pi");
    int* (*SortFunc)(int*, int) = dlsym(handle, "Sort");

    printf("  0           - переключить библиотеку\n");
    printf("  1 K         - вычислить Pi\n");
    printf("  2 a1 a2 ... - отсортировать числа\n");
    printf("  -1          - выход\n");
    printf("Введите команду: ");

    int cmd;

    while (1) {
        if (scanf("%d", &cmd) == EOF) break;

        if (cmd == -1) {
            printf("Выход\n");
            break;
        }

        if (cmd == 0) {
            dlclose(handle);
            cur = 1 - cur;

            handle = dlopen(libs[cur], RTLD_LAZY);
            if (!handle) {
                printf("Ошибка загрузки библиотеки: %s\n", dlerror());
                return 1;
            }

            PiFunc = dlsym(handle, "Pi");
            SortFunc = dlsym(handle, "Sort");

            printf("Библиотека переключена на реализацию %d\n", cur + 1);
        }

        else if (cmd == 1) {
            int K;
            scanf("%d", &K);
            printf("Pi = %f (библиотека %d)\n", PiFunc(K), cur + 1);
        }

        else if (cmd == 2) {
            int numbers[100];
            int count = 0;
            int num;

            while (scanf("%d", &num) == 1 && count < 100) {
                numbers[count++] = num;
```

```
                if (getchar() == '\n') break;
            }

            if (count > 0) {
                printf("Исходный: ");
                for (int i = 0; i < count; i++) printf("%d ", numbers[i]);
                printf("\n");

                int* sorted = SortFunc(numbers, count);

                printf("Отсортированный: ");
                for (int i = 0; i < count; i++) printf("%d ", sorted[i]);
                printf(" (библиотека %d)\n", cur + 1);

                free(sorted);
            }
        }

        else {
            printf("нет такой команды\n");
        }
    }

    dlclose(handle);
    return 0;
}
```

Протокол работы программы

```
kishaki@416:~/lab_OS4$ ./static
  1 K         - вычислить Pi
  2 a1 a2 ... - отсортировать числа
  -1          - выход
Введите команду: 1 1000
Pi = 3.140593
2 52 42 67 7
Исходный: 52 42 67 7
Отсортированный: 7 42 52 67
```

```
kishaki@416:~/lab_OS4$ ./dynamic
  0           - переключить библиотеку
  1 K         - вычислить Pi
  2 a1 a2 ... - отсортировать числа
  -1          - выход
Введите команду: 1 1000
Pi = 3.140593 (библиотека 1)
2 52 42 67 7
Исходный: 52 42 67 7
Отсортированный: 7 42 52 67  (библиотека 1)
0
Библиотека переключена на реализацию 2
1 1000
Pi = 3.140806 (библиотека 2)
2 52 42 67 7
Исходный: 52 42 67 7
Отсортированный: 7 42 52 67  (библиотека 2)
```

Strace:

Статическая версия (static):
strace -f ./static

execve("./static", ["./static"], 0x7ffe3eab2388 /* 29 vars */) = 0
brk(NULL) = 0x5fc304274000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7d765e9a5000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=37043, ...}) = 0
mmap(NULL, 37043, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7d765e99b000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7d765e600000
mmap(0x7d765e628000, 1605632, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7d765e628000
mmap(0x7d765e7b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x7d765e7b0000
mmap(0x7d765e7ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x7d765e7ff000
mmap(0x7d765e805000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7d765e805000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7d765e998000
arch_prctl(ARCH_SET_FS, 0x7d765e998740) = 0
set_tid_address(0x7d765e998a10) = 42005
set_robust_list(0x7d765e998a20, 24) = 0
rseq(0x7d765e999060, 0x20, 0, 0x53053053) = 0
mprotect(0x7d765e7ff000, 16384, PROT_READ) = 0
mprotect(0x5fc2e5247000, 4096, PROT_READ) = 0
mprotect(0x7d765e9dd000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7d765e99b000, 37043) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
getrandom("\xdd\x13\xfa\x28\x57\x5f\x31\x24", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x5fc304274000
brk(0x5fc304295000) = 0x5fc304295000
write(1, " 1 K - \320\262\321\213\321\207\320\270\321\201\320\273\320\270\321\202"..., 38 1 K - вычислить Pi
) = 38
write(1, " 2 a1 a2 ... - \320\276\321\202\321\201\320\276\321\200\321\202\320\270\321\200"..., 54 2 a1 a2 ... - отсортировать числа

) = 54
write(1, " -1   - \320\262\321\213\321\205\320\276\320\264\n", 27 -1   - выход
) = 27
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\272\320\276\320\274\320\260\320\275\320\264\321\203: ", 31Введите команду:) = 31
read(0, 1 1000
"1 1000\n", 1024) = 7
write(1, "Pi = 3.140593\n", 14Pi = 3.140593
) = 14
read(0, 2 52 42 67 7
"2 52 42 67 7\n", 1024) = 13
write(1, "\320\230\321\201\321\205\320\276\320\264\320\275\321\213\320\271: 52 42 67 7 \n", 30Исходный: 52 42 67 7
) = 30
write(1,
"\320\236\321\202\321\201\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\260\320\275\320\275\321\213\320\271: "..., 44Отсортированный: 7 42 52 67
) = 44
read(0, -1
"-1\n", 1024) = 3
write(1, "\320\222\321\213\321\205\320\276\320\264\n", 11Выход
) = 11
lseek(0, -1, SEEK_CUR) = -1 ESPIPE (Illegal seek)
exit_group(0) = ?
+++ exited with 0 +++

Динамическая версия (dynamic):
strace -f ./dynamic
execve("./dynamic", ["./dynamic"], 0x7ffdd8e1e428 /* 29 vars */) = 0
brk(NULL) = 0x5ad00f6fc000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x72a69cba7000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=37043, ...}) = 0
mmap(NULL, 37043, PROT_READ, MAP_PRIVATE, 3, 0) = 0x72a69cb9d000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\220\243\2\0\0\0\0"..., 832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
fstat(3, {st_mode=S_IFREG|0755, st_size=2125328, ...}) = 0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"..., 784, 64) = 784
mmap(NULL, 2170256, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x72a69c800000
mmap(0x72a69c828000, 1605632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x72a69c828000

mmap(0x72a69c9b0000, 323584, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b0000) = 0x72a69c9b0000
mmap(0x72a69c9ff000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1fe000) = 0x72a69c9ff000
mmap(0x72a69ca05000, 52624, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x72a69ca05000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x72a69cb9a000
arch_prctl(ARCH_SET_FS, 0x72a69cb9a740) = 0
set_tid_address(0x72a69cb9aa10) = 486
set_robust_list(0x72a69cb9aa20, 24) = 0
rseq(0x72a69cb9b060, 0x20, 0, 0x53053053) = 0
mprotect(0x72a69c9ff000, 16384, PROT_READ) = 0
mprotect(0x5acfe5de5000, 4096, PROT_READ) = 0
mprotect(0x72a69cbdf000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x72a69cb9d000, 37043) = 0
getrandom("\x02\x90\xfb\x2c\x5e\x1c\x7e\x34", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x5ad00f6fc000
brk(0x5ad00f71d000) = 0x5ad00f71d000
openat(AT_FDCWD, "./lib1.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=15576, ...}) = 0
getcwd("/home/kishaki/lab_OS4", 128) = 22
mmap(NULL, 16408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x72a69cba2000
mmap(0x72a69cba3000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x72a69cba3000
mmap(0x72a69cba4000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x72a69cba4000
mmap(0x72a69cba5000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x72a69cba5000
close(3) = 0
mprotect(0x72a69cba5000, 4096, PROT_READ) = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, " 0 - \320\277\320\265\321\200\320\265\320\272\320\273\321\216\321\207"..., 60 0 - переключить библиотеку
) = 60
write(1, " 1 K - \320\262\321\213\321\207\320\270\321\201\320\273\320\270\321\202"..., 38 1 K - вычислить Pi
) = 38
write(1, " 2 a1 a2 ... - \320\276\321\202\321\201\320\276\321\200\321\202\320\270\321\200"..., 54 2 a1 a2 ... - отсортировать числа
) = 54
write(1, " -1 - \320\262\321\213\321\205\320\276\320\264\n", 27 -1 - выход
) = 27
fstat(0, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0

write(1, "\320\222\320\262\320\265\320\264\320\270\321\202\320\265
\320\272\320\276\320\274\320\260\320\275\320\264\321\203: ", 31Введите команду:) = 31
read(0, 1 1000
"1 1000\n", 1024)   = 7
write(1, "Pi = 3.140593 (\320\261\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320"...,
39Pi = 3.140593 (библиотека 1)
) = 39
read(0, 2 52 42 67 7
"2 52 42 67 7\n", 1024)   = 13
write(1, "\320\230\321\201\321\205\320\276\320\264\320\275\321\213\320\271: 52 42 67 7 \n",
30Исходный: 52 42 67 7
) = 30
write(1,
"\320\236\321\202\321\201\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\260\320\27
5\320\275\321\213\320\271: "..., 69Отсортированный: 7 42 52 67 (библиотека 1)
) = 69
read(0, 0
"0\n", 1024)   = 2
munmap(0x72a69cba2000, 16408)   = 0
openat(AT_FDCWD, "./lib2.so", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=15664, ...}) = 0
getcwd("/home/kishaki/lab_OS4", 128)   = 22
mmap(NULL, 16416, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x72a69cba2000
**mmap(0x72a69cba3000, 4096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x72a69cba3000**
**mmap(0x72a69cba4000, 4096, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE,
3, 0x2000) = 0x72a69cba4000**
**mmap(0x72a69cba5000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x72a69cba5000**
close(3)   = 0
mprotect(0x72a69cba5000, 4096, PROT_READ) = 0
write(1, "\320\221\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320\272\320\260
\320\277\320\265\321\200\320\265\320\272\320"..., 72Библиотека переключена на реализацию 2
) = 72
read(0, 1 1000
"1 1000\n", 1024)   = 7
write(1, "Pi = 3.140806 (\320\261\320\270\320\261\320\273\320\270\320\276\321\202\320\265\320"...,
39Pi = 3.140806 (библиотека 2)
) = 39
read(0, 2 52 42 67 7
"2 52 42 67 7\n", 1024)   = 13
write(1, "\320\230\321\201\321\205\320\276\320\264\320\275\321\213\320\271: 52 42 67 7 \n",
30Исходный: 52 42 67 7
) = 30
write(1,
"\320\236\321\202\321\201\320\276\321\200\321\202\320\270\321\200\320\276\320\262\320\260\320\27
5\320\275\321\213\320\271: "..., 69Отсортированный: 7 42 52 67 (библиотека 2)
) = 69

```
read(0, -1
"-1\n", 1024)                   = 3
write(1, "\320\222\321\213\321\205\320\276\320\264\n", 11Выход
) = 11
munmap(0x72a69cba2000, 16416)           = 0
lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Illegal seek)
exit_group(0)                           = ?
+++ exited with 0 +++
```

Вывод

В ходе выполения лабораторной работы ыбла реализована программа с подррежкой статической и динамической загрузки функций через разделяемые библиотеки. Были созданы две реализациий математических функций в виде отдельных бибилиотек lib1.so и lib2.so.

Для статической линковки была разработана программа, которая на этапе компиляции включает в себя выбранную реализацию функций, создавая единый исполняемый файл. Этот подход обеспечивает портативность и независимость от наличия библиотек в системе, но приводит к увеличению размера программы и необходимости перекомпиляции при изменении реализации.

Для динамической загрузки была создана программа, которая во время выполнения может загружать и выгружать библиотеки, а также переключаться между разными реализациями. Этот подход демонстрирует гибкость, экономию памяти и возможность обновления компонентов без перекомпиляции основной программы.