МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

## Факультет информационных технологий и робототехники

Кафедра программного обеспечения информационных систем и технологий

Отчет по лабораторной работе № 9

по дисциплине:” Системное программирование”

на тему: ” Создание и использование библиотек в Linux ”

Вариант 6

Выполнили**:**  студент группы 10701222 Дяденькина Т.М.

Приняла**:** пр. Давыденко Н.В.

Минск 2024

**Цель работы**: Изучить и закрепить на практике создание и использование статически и динамически подключаемых пользовательских библиотек в операционных системах семейства Linux.

**Задание**

Задание 1

Произвести рефакторинг проекта предыдущей лабораторной работы по автосборке проекта: вынесите код функций бизнес логики в отдельную статическую библиотеку. Для автосборки проекта используйте утилиту make, обязательно.

При помощи программы ldd проверим зависимости, образовавшиеся между библиотеками.

Сравните сумму размеров объектных файлов проекта и исполняемого файла.

Результаты отобразите скриншотами.

Копию исполняемого файла расположите на «Рабочем столе», запустите программу. Какой будет результат?

Задание 2

Произвести рефакторинг проекта предыдущей лабораторной работы по автосборке проекта: вынесите код функций бизнес логики в отдельную динамическую библиотеку со статическим вызовом. Для автосборки проекта используйте утилиту make, обязательно.

При помощи программы ldd проверим зависимости, образовавшиеся между библиотеками.

Сравните сумму размеров объектных файлов проекта и исполняемого файла.

Результаты отобразите скриншотами.

Копию исполняемого файла расположите на «Рабочем столе», запустите программу. Какой будет результат? Если возникла проблема, решите ее и поясните каким способом. Сравните результаты первого и второго упражнения, сделайте вывод.

Задание 3

Произвести рефакторинг проекта предыдущей лабораторной работы по автосборке проекта: вынесите код функций бизнес логики в отдельную динамическую библиотеку с динамическим вызовом. Для автосборки проекта используйте утилиту make, обязательно.

При помощи программы ldd проверим зависимости, образовавшиеся между библиотеками.

Сравните сумму размеров объектных файлов проекта и исполняемого файла.

Результаты отобразите скриншотами.

Копию исполняемого файла расположите на «Рабочем столе», запустите программу. Какой будет результат? Если возникла проблема, решите ее, поясните причину проблемы и каким способом вы ее решили. Сравните результаты с предыдущими, сделайте вывод.

**Выполнение 1 задания**

**game.cpp**

#include "game.h"

#include <cstdlib>

int generate\_random\_number(int min, int max) {

return rand() % (max - min + 1) + min;

}

bool check\_guess(int guess, int target) {

return guess == target;

}

**game.h**

#pragma once

int generate\_random\_number(int min, int max);

bool check\_guess(int guess, int target);

**main.cpp**

#include <iostream>

#include "game.h"

int main() {

int minRange, maxRange, targetNumber, guess, attempts;

char playAgain;

std::cout << "Welcome to the Guess the Number game!" << std::endl;

do {

attempts = 0;

std::cout << "Enter the minimum range: ";

std::cin >> minRange;

std::cout << "Enter the maximum range: ";

std::cin >> maxRange;

std::cout << "Think of a number between " << minRange << " and " << maxRange << "." << std::endl;

// Определение загаданного числа пользователем

std::cout << "Enter your target number: ";

std::cin >> targetNumber;

int left = minRange;

int right = maxRange;

while (left <= right) {

guess = (left + right) / 2; // Оптимальный алгоритм двоичного поиска

attempts++;

if (check\_guess(guess, targetNumber)) {

std::cout << "The computer guessed your number " << targetNumber << " in " << attempts << " attempts." << std::endl;

break;

}

else if (guess < targetNumber) {

std::cout << "The computer guessed " << guess << ". It's lower!" << std::endl;

left = guess + 1;

}

else {

std::cout << "The computer guessed " << guess << ". It's higher!" << std::endl;

right = guess - 1;

}

}

if (left > right) {

std::cout << "The computer could not guess your number. You win!" << std::endl;

}

std::cout << "Do you want to play again? (Y/N): ";

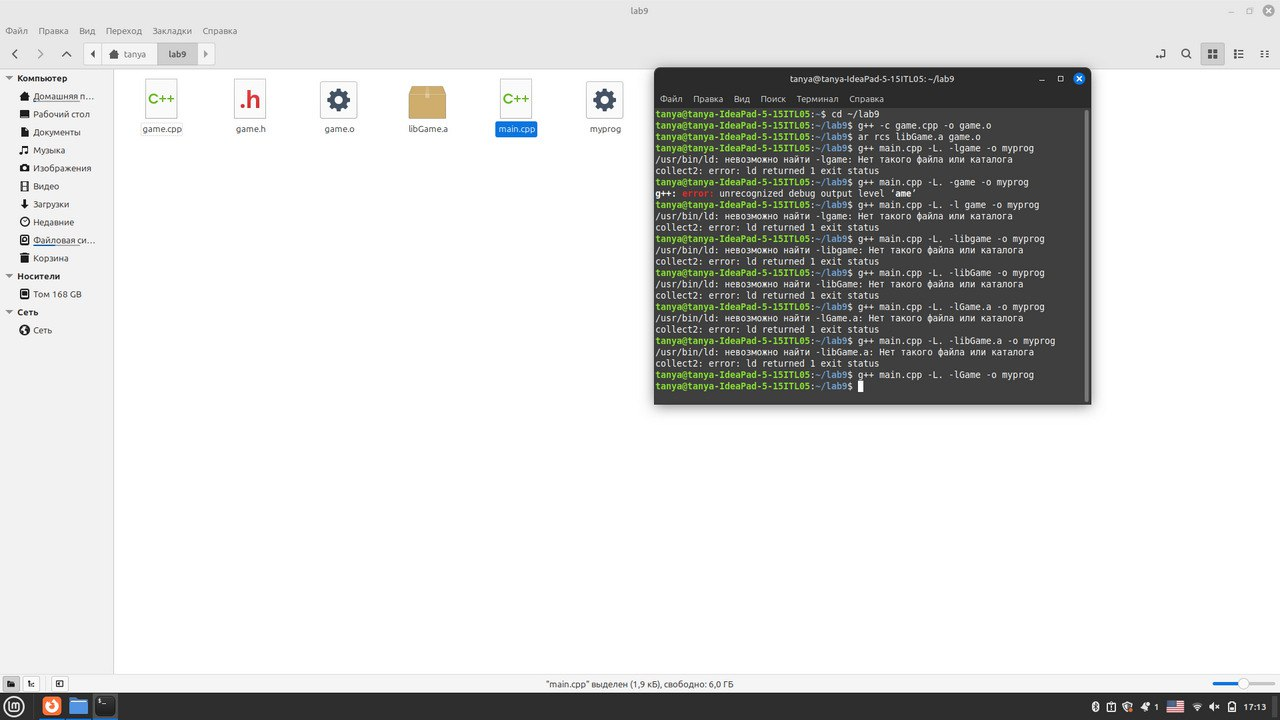
std::cin >> playAgain;

} while (playAgain == 'Y' || playAgain == 'y');

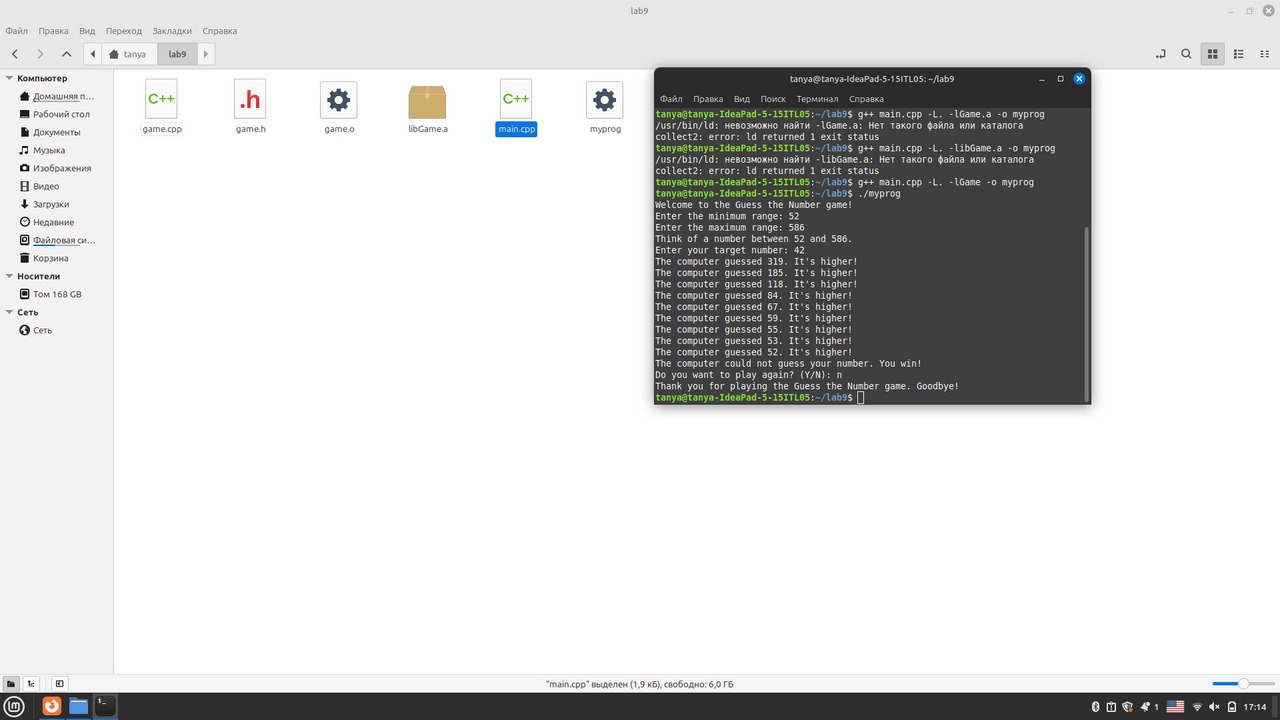
std::cout << "Thank you for playing the Guess the Number game. Goodbye!" << std::endl;

return 0;

}

Создаем объектный файл, затем статическую библиотеку (куда засунули коды функций бизнес логики) и компилируем исходный код с этой библиотекой с черт знает какого раза.

Запускаем программу



Автосборка проекта с утилитой make

**makefile**

CC = g++

CFLAGS = -Wall -Wextra -std=c++11

LDFLAGS = -L. -lGame

SRCS = main.cpp game.cpp

OBJS = $(SRCS:.cpp=.o)

EXEC = main

all: $(EXEC)

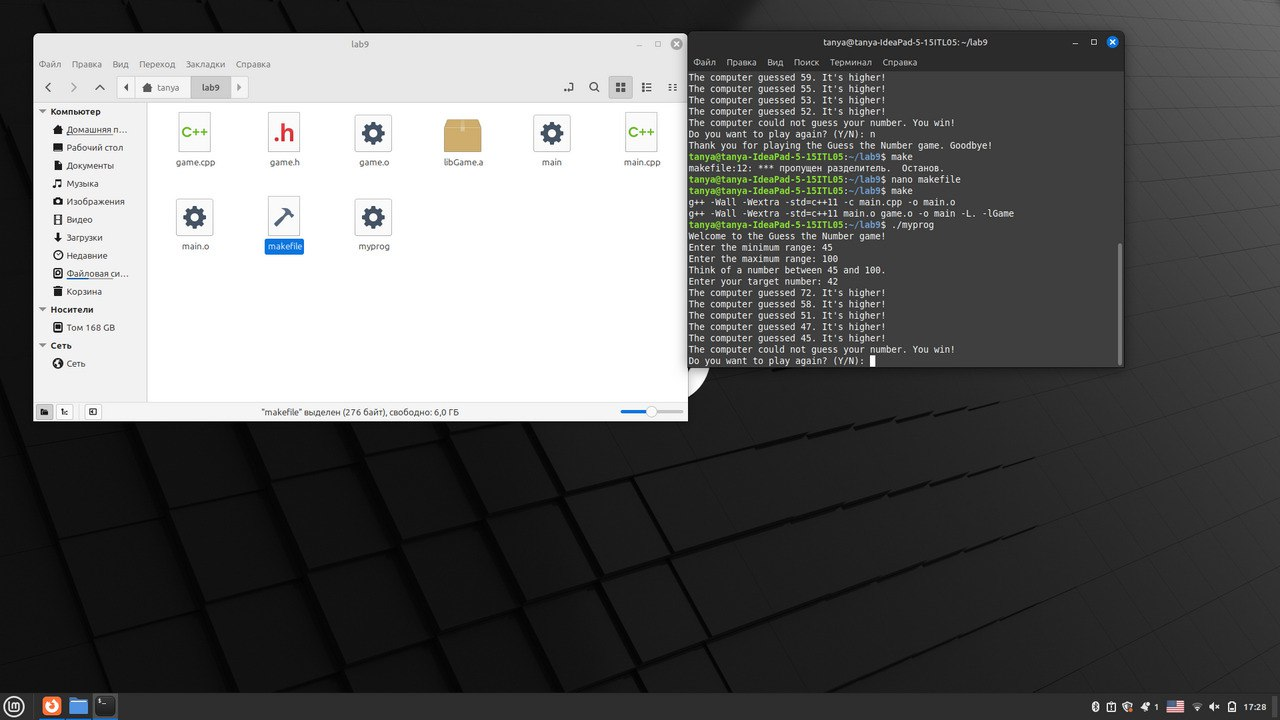
$(EXEC): $(OBJS)

$(CC) $(CFLAGS) $(OBJS) -o $(EXEC) $(LDFLAGS)

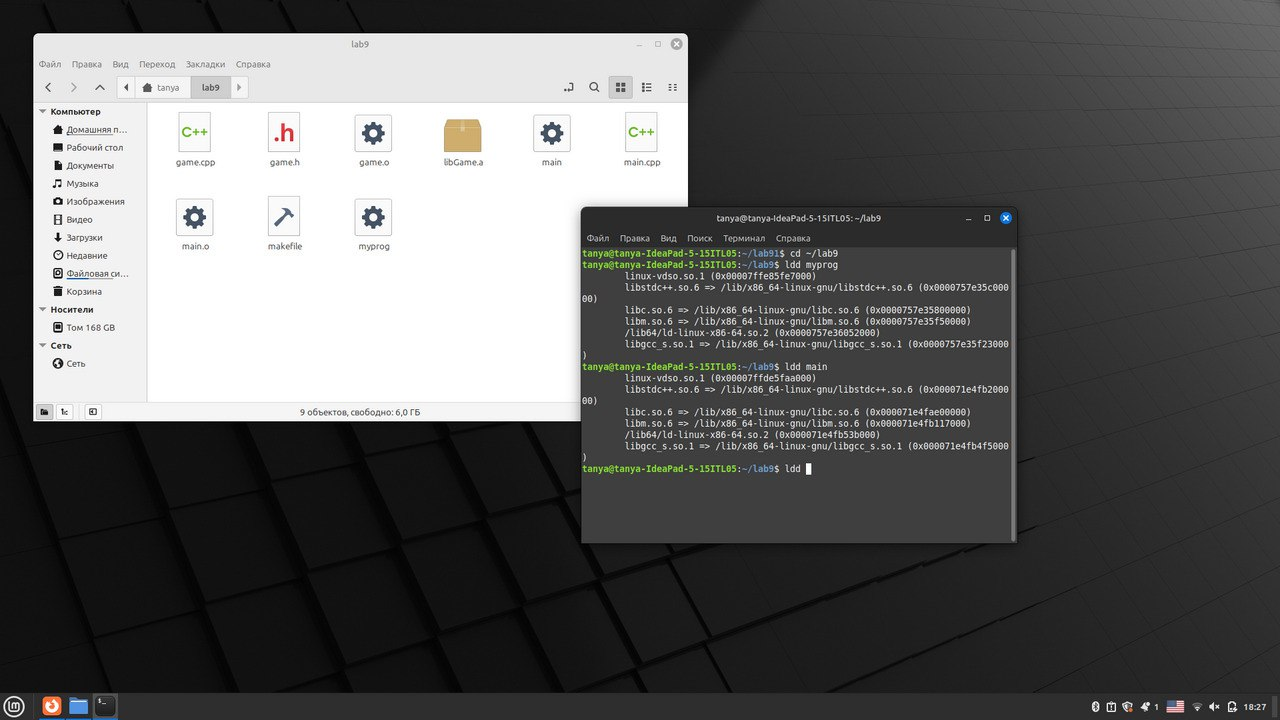
%.o: %.cpp

$(CC) $(CFLAGS) -c $< -o $@

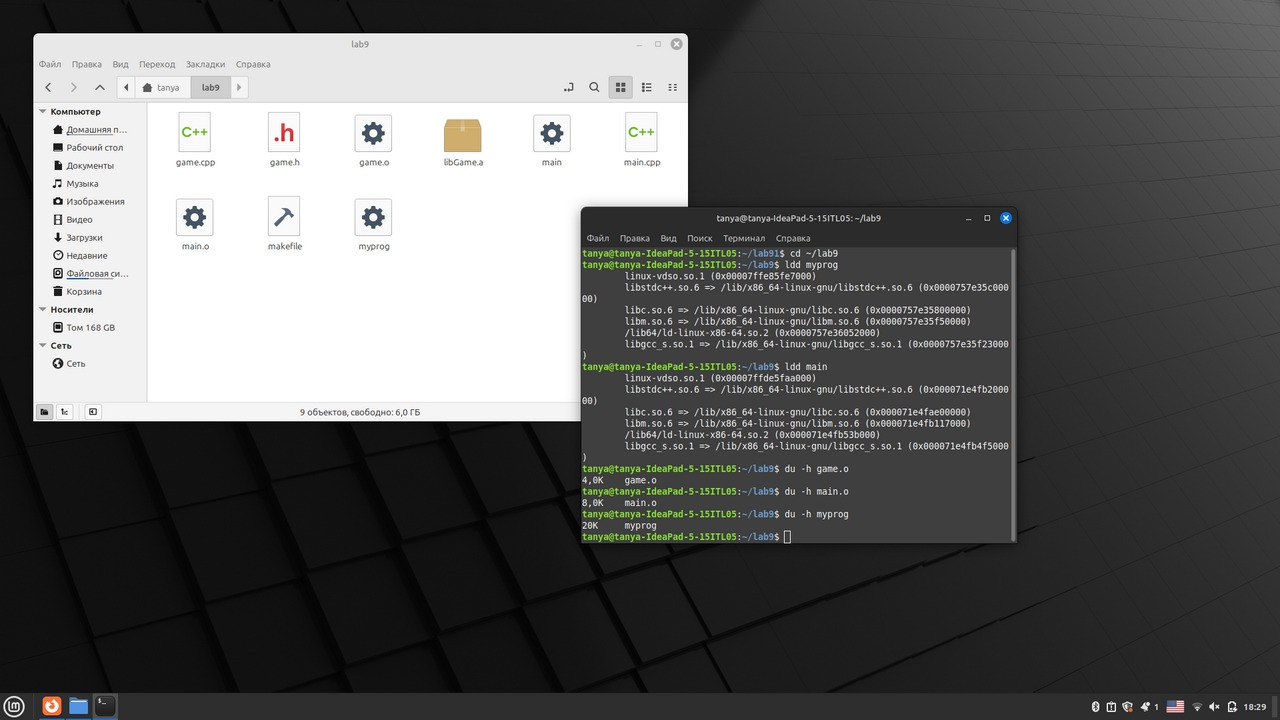
clean:

 rm -f $(OBJS) $(EXEC)

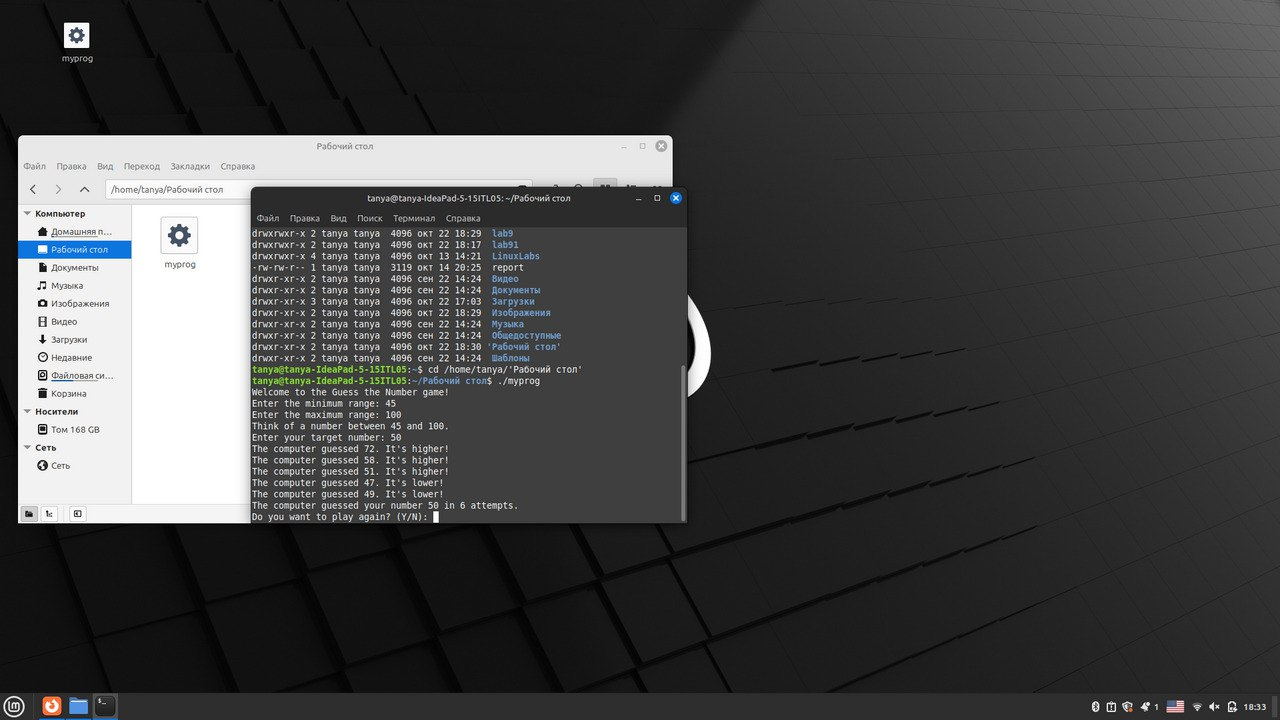
ldd исполняемого файла



Сравнение размеров исполняемого файла и суммарного размера объектных файлов



Задание с рабочим столом



**Контрольные вопросы**

На какой стадии происходит подключение библиотек (объектных файлов) к основному модулю программы?

Ответ: