

# **ЛАБОРАТОРНАЯ РАБОТА №4**

**Отчёт**

Глобин Никита Анатольевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
3.1	задание 1. написать программу Hello world! . . . . .	7
3.2	задание 2. использование транслятора NASM . . . . .	8
3.3	задание 3. использование компоновщика LD . . . . .	9
3.4	задание 4. использование компоновщика LD . . . . .	10
3.5	задание 5. самостоятельная работа . . . . .	10
<b>4</b>	<b>Выводы</b>	<b>13</b>

# Список иллюстраций

3.1	photo 1.1	7
3.2	photo 1.2	8
3.3	photo 2.1	8
3.4	photo 2.2	9
3.5	photo 3.1	9
3.6	photo 3.2	10
3.7	photo 4.1	10
3.8	photo 5.1	11
3.9	photo 5.2	11
3.10	photo 5.3	12
3.11	photo 5.4	12
3.12	photo 5.5	12

## **Список таблиц**

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## 2 Задание

написать программу Hello world!

использование транслятора NASM

использование компоновщика LD

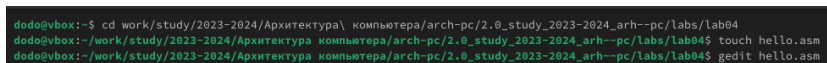
Запуск исполняемого файла

самостоятельная работа

## 3 Выполнение лабораторной работы

### 3.1 задание 1. написать программу Hello world!

1. Переходим в каталог и создаём там файл с расширением .asm `cd ~/work/arch-pc/lab04`



```
dodo@vbox:~$ cd work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/Labs/lab04
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/Labs/lab04$ touch hello.asm
dodo@vbox:~/work/study/2023-2024/Архитектура компьютера/arch-pc/2.0_study_2023-2024_arh--pc/Labs/lab04$ gedit hello.asm
```

Рис. 3.1: photo 1.1

2. открываем созданный файл и пишем в нём программу `gedit hello.asm`

; hello.asm SECTION .data ; Начало секции данных hello: DB 'Hello world!',10 ; 'Hello world!' плюс ; символ перевода строки helloLen: EQU \$-hello ; Длина строки hello SECTION .text ; Начало секции кода GLOBAL \_start \_start: ; Точка входа в программу mov eax,4 ; Системный вызов для записи (sys\_write) mov ebx,1 ; Описатель файла '1' - стандартный вывод mov ecx,hello ; Адрес строки hello в ecx mov edx,helloLen ; Размер строки hello int 80h ; Вызов ядра mov eax,1 ; Системный вызов для выхода (sys\_exit) mov ebx,0 ; Выход с кодом возврата '0' (без ошибок) int 80h ; Вызов ядра



Рис. 3.2: photo 1.2

## 3.2 задание 2. использование транслятора NASM

1. используя транслятор NASM начинаем работать с нашим файлом `nasm -f elf hello.asm`

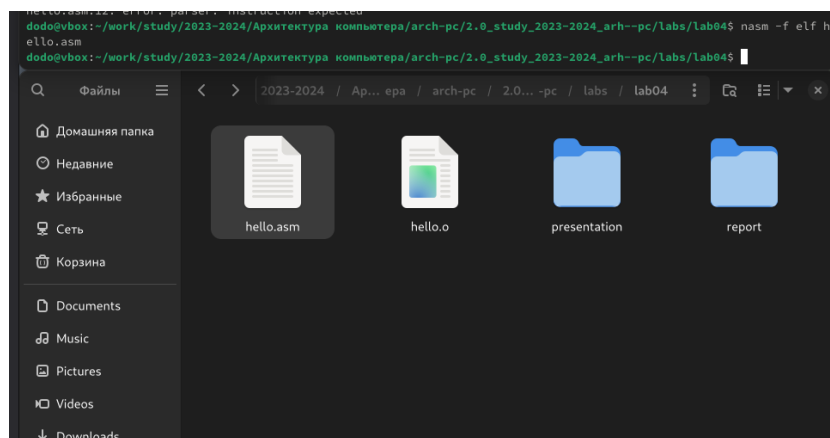


Рис. 3.3: photo 2.1

2. используем расширенный функционал NASM `nasm -o obj.o -f elf -g -l list.lst hello.asm`



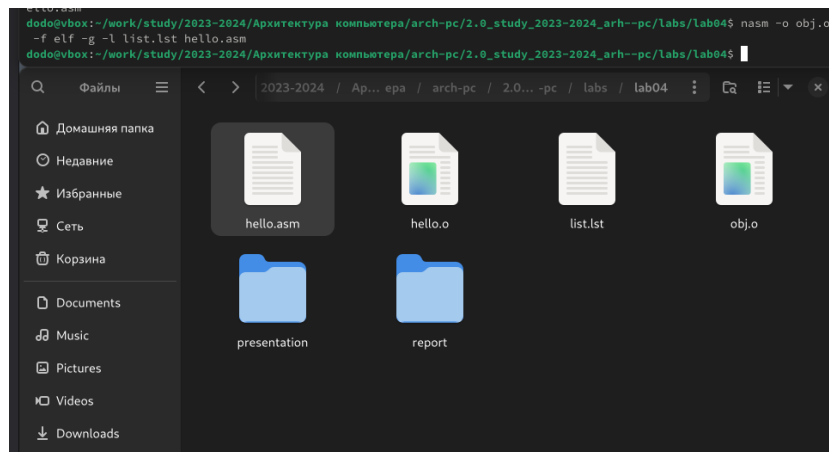


Рис. 3.4: photo 2.2

### 3.3 задание 3. использование компоновщика LD

1. отдаём файл компоновщику LD для получения исполняемую программу d  
-m elf\_i386 hello.o -o hello

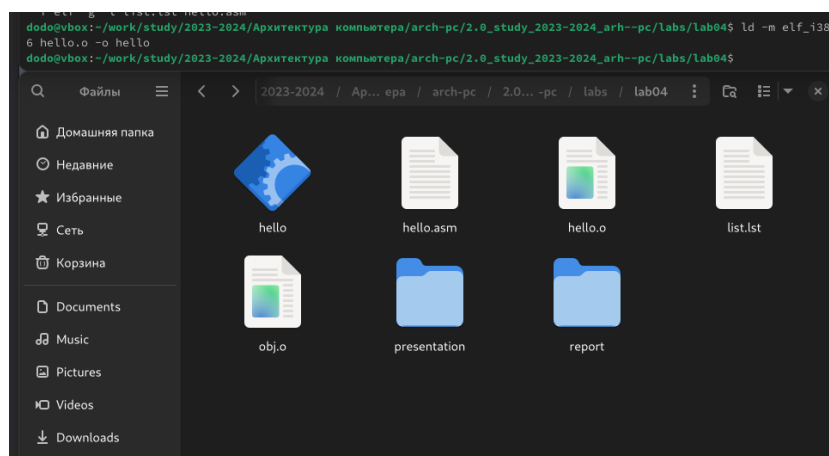


Рис. 3.5: photo 3.1

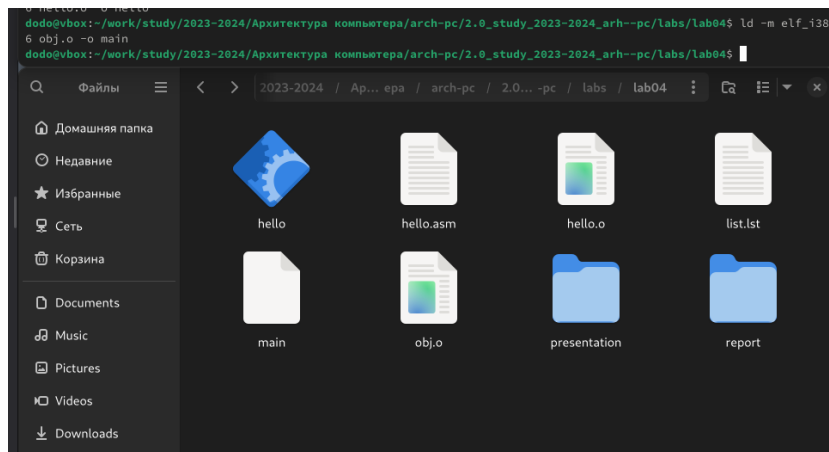


Рис. 3.6: photo 3.2

### 3.4 задание 4. использование компоновщика LD

1. запускаем файл ./hello

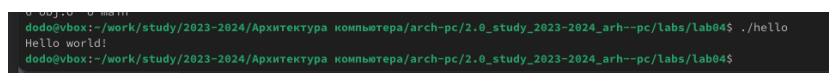


Рис. 3.7: photo 4.1

### 3.5 задание 5. самостоятельная работа

1. создаём копию файла при помощи команды cp и называем копию lab4.asm

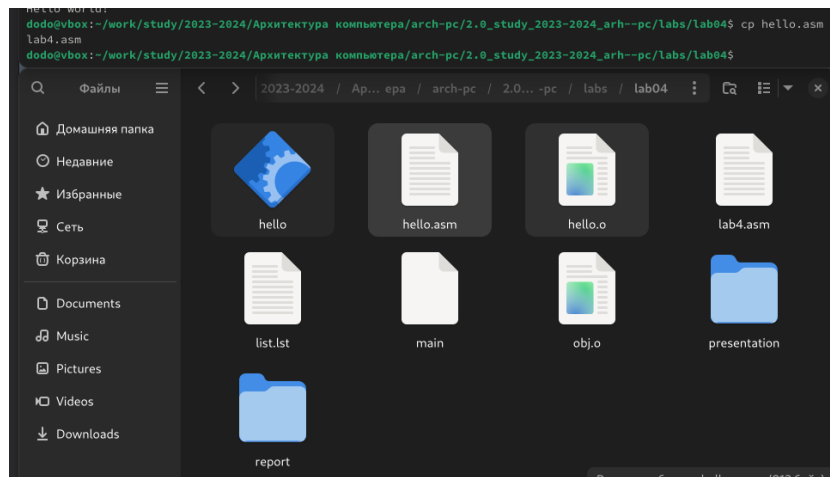


Рис. 3.8: photo 5.1

2. использую текстовый редактор переписываем программ так чтобы она выводила имя и фамилию

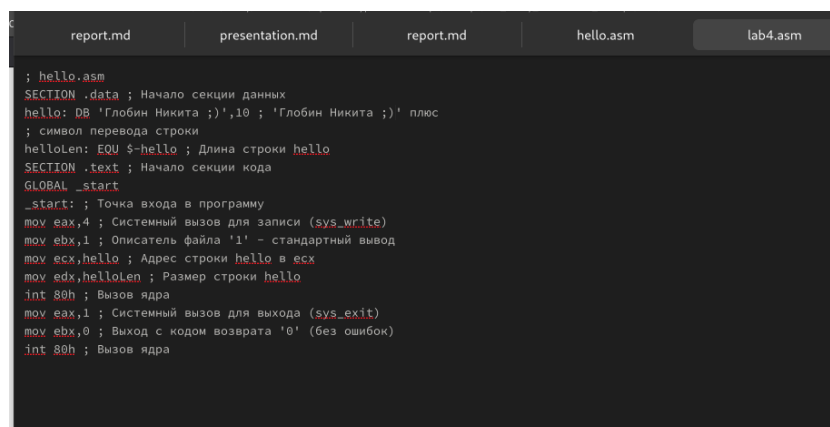


Рис. 3.9: photo 5.2

3. компилируем и запускаем нашу программу

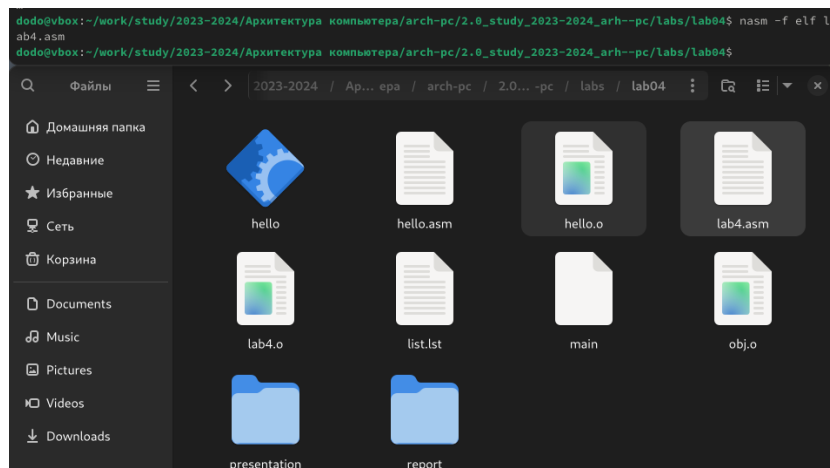


Рис. 3.10: photo 5.3

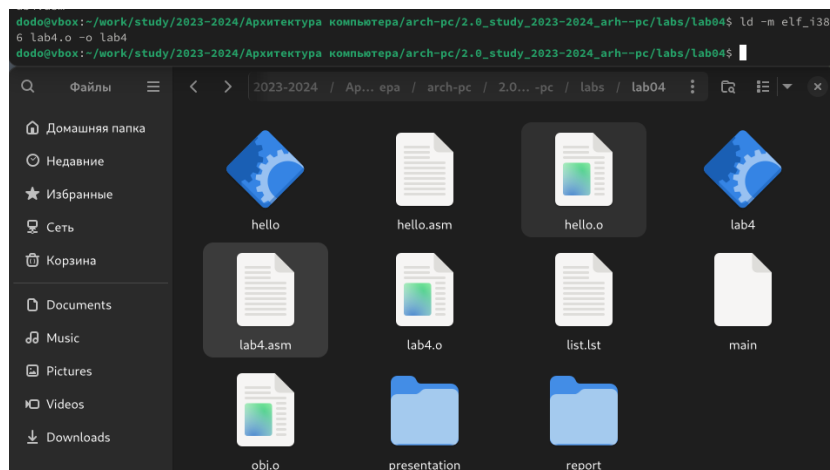


Рис. 3.11: photo 5.4

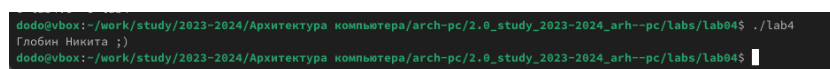


Рис. 3.12: photo 5.5

## **4 Выводы**

В ходе лабораторной работы я освоил процедуры компиляции и сборки программ, написанных на ассемблере NASM.