

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

дисциплина: *Архитектура компьютера*

Студент: Гасанова Шакира Чингизовна

Группа: НКАбд-05-24

МОСКВА

2024 г.

Содержание

1 Цель работы	3
2 Задание	4
3 Теоретическое введение	5
4 Выполнение лабораторной работы	
4.1 Создание программы Hello world!.....	7
4.2 Работа с транслятором NASM.....	8
4.3 Работа с расширенным синтаксисом командной строки NASM.....	9
4.4 Работа с компоновщиком LD.....	9
4.5 Запуск исполняемого файла.....	9
4.6 Выполнение заданий для самостоятельной работы.....	10
5 Выводы	12
6 Источники	13

1 Цель работы

Цель данной лабораторной работы - освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Задание

1. Создание программы Hello world!
2. Работа с транслятором NASM
3. Работа с расширенным синтаксисом командной строки NASM
4. Работа с компоновщиком LD
5. Запуск исполняемого файла
6. Выполнение заданий для самостоятельной работы.

3 Теоретическое введение

Основными компонентами любой ЭВМ (электронно-вычислительной машины) являются центральный процессор, память и периферийные устройства. Эти элементы взаимодействуют через общую шину, которая физически представляет собой набор проводников, соединяющих устройства. В современных компьютерах проводники выполнены в виде дорожек на материнской плате. Основная задача процессора заключается в обработке информации и координации всех компонентов системы.

Центральный процессор состоит из следующих элементов:

1. Арифметико-логическое устройство (АЛУ) — выполняет арифметические и логические операции для обработки данных, хранящихся в памяти.
2. Устройство управления (УУ) — управляет работой всех компонентов компьютера.
3. Регистры — быстрая внутренняя память процессора для хранения промежуточных результатов вычислений. Регистры делятся на два типа: общего назначения и специальные.

Для программирования на языке ассемблера важно знать, какие регистры доступны и как они используются, поскольку большинство команд работают с данными, находящимися в регистрах. В архитектуре x86 регистры имеют следующие обозначения:

- 64-битные: RAX, RCX, RDX, RBX, RSI, RDI
- 32-битные: EAX, ECX, EDX, EBX, ESI, EDI
- 16-битные: AX, CX, DX, BX, SI, DI
- 8-битные: AH, AL, CH, CL, DH, DL, BH, BL

Ещё одним важным компонентом ЭВМ является оперативная память (ОЗУ), которая представляет собой быстродействующее энергозависимое устройство, предназначенное для хранения программ и данных, с которыми процессор работает в данный момент. ОЗУ состоит из ячеек памяти, каждая из которых имеет свой адрес.

Периферийные устройства включают:

- Устройства внешней памяти для долгосрочного хранения данных.
- Устройства ввода-вывода для взаимодействия компьютера с внешней средой.

Компьютер работает на основе принципа программного управления, выполняя задачи как последовательность действий, записанных в программе. Команды процессора имеют двоичное представление и состоят из

операционной и адресной частей. Операционная часть хранит код команды, а адресная — данные или их адреса.

Командный цикл процессора включает следующие этапы:

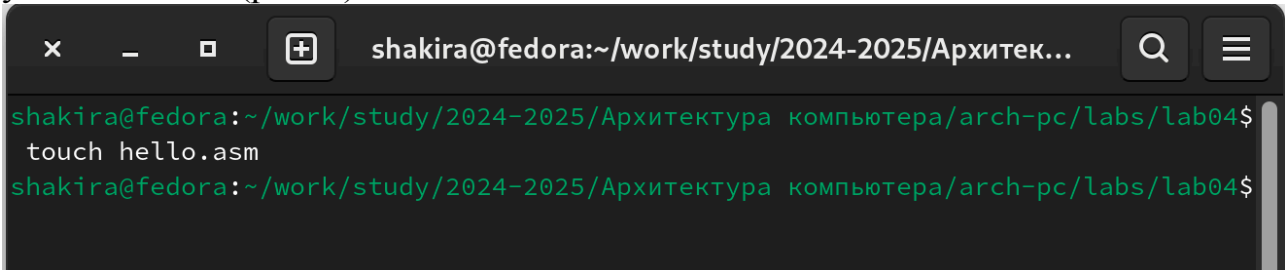
1. Формирование адреса команды в памяти.
2. Чтение команды и её декодирование.
3. Выполнение команды.
4. Переход к следующей команде.

Язык ассемблера (asm) — это машинно-ориентированный язык низкого уровня. NASM — это популярный ассемблер, поддерживающий синтаксис Intel и инструкции архитектуры x86-64, позволяющий создавать объектные файлы для различных операционных систем.

4 Выполнение лабораторной работы

4.1 Создание программы Hello world!

С помощью утилиты `cd` перемещаюсь в каталог, в котором буду работать. Создаю в текущем каталоге пустой текстовый файл `hello.asm` с помощью утилиты `touch` (рис. 1).

A terminal window with a dark background. The title bar shows the user 'shakira@fedora' and the current directory '~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04'. The prompt is 'shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04\$'. The user enters 'touch hello.asm' and presses enter. The prompt returns, indicating the command was successful.

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$  
touch hello.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис.1 Создание файла

Открываю созданный файл в `mousepad` (рис. 2).

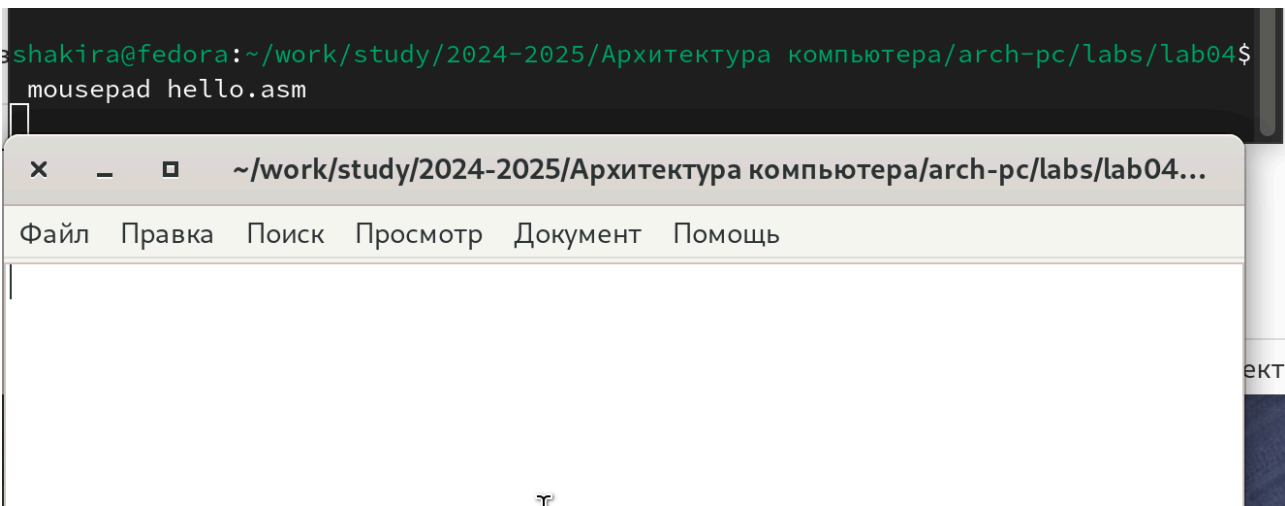
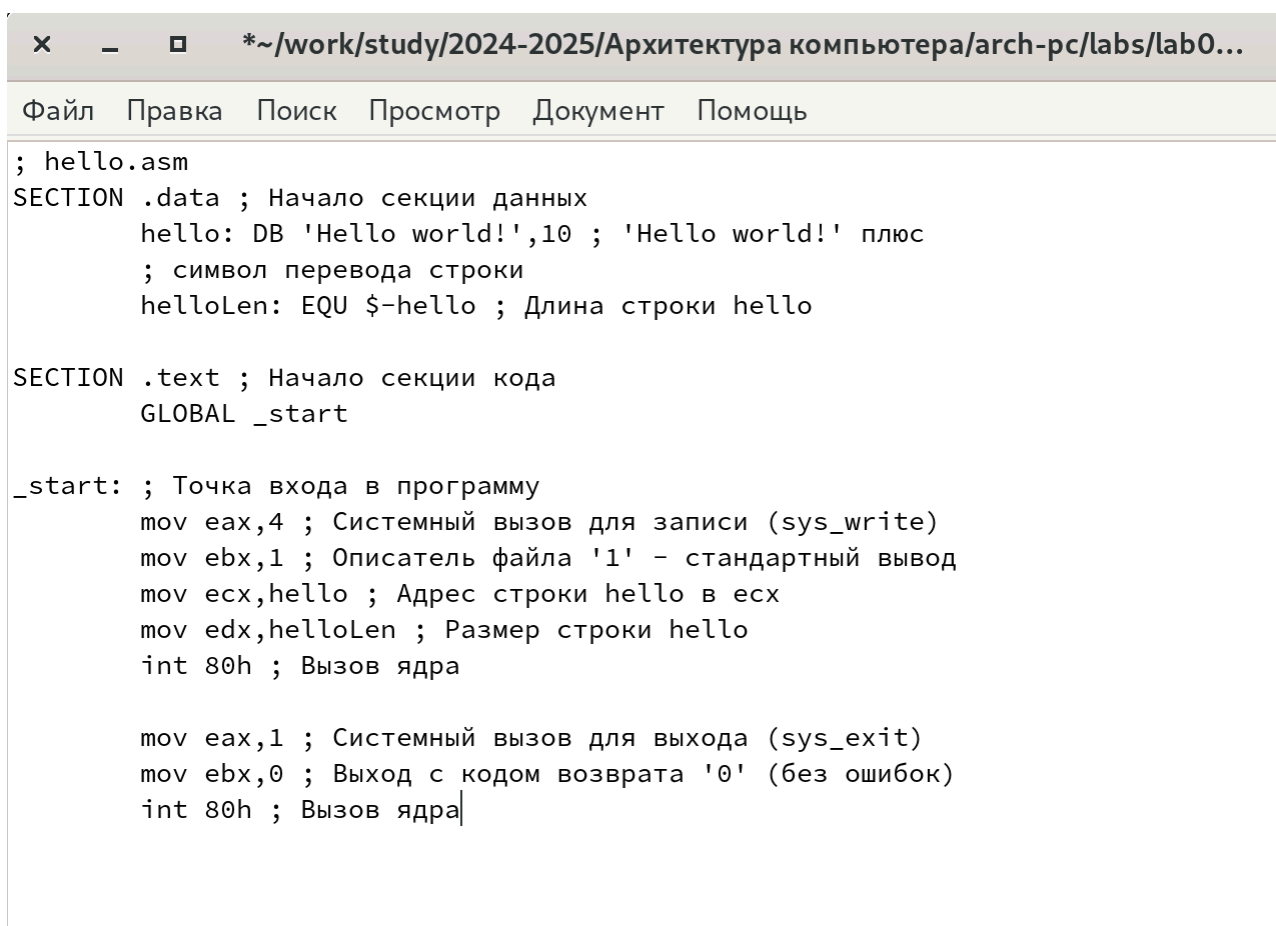


Рис.2 Открытие файла

Заполняю файл, вставляя в него программу для вывода “Hello word!” (рис. 3).



```
; 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

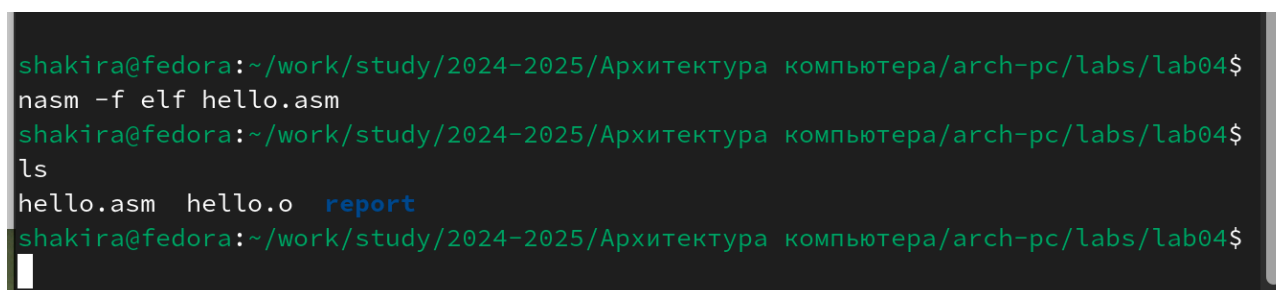
_start: ; Точка входа в программу
    mov eax,4 ; Системный вызов для записи (sys_write)
    mov ebx,1 ; Описатель файла '1' - стандартный вывод
    mov ecx,hello ; Адрес строки hello в ecx
    mov edx,helloLen ; Размер строки hello
    int 80h ; Вызов ядра

    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 Заполнение файла

4.2 Работа с транслятором NASM

Я трансформирую текст программы, выводящей "Hello, world!", в объектный код с помощью транслятора NASM, используя команду `nasm -f elf hello.asm`. Ключ `-f` указывает NASM на необходимость создания бинарного файла в формате `elf`. Затем с помощью утилиты `ls` проверяю, что команда выполнена корректно — действительно, создан файл "hello.o" (рис. 4).



```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
nasm -f elf hello.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
ls
hello.asm  hello.o  report
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис.4 Трансформация текста программы

4.3 Работа с расширенным синтаксисом командной строки NASM

Я ввожу команду для компиляции файла `hello.asm` в объектный файл `obj.o`, используя ключ `-g` для включения отладочных символов. Также с помощью ключа `-l` создается файл листинга `list.lst`. Затем с помощью утилиты `ls` проверяю корректность выполнения команды (рис. 5).

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$  
nasm -o obj.o -f elf -g -l list.lst hello.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$  
ls  
hello.asm  hello.o  list.lst  obj.o  report  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arch-pc/labs/lab04$
```

Рис.5 Копмиляция текста программы

4.4 Работа с компоновщиком LD

Я передаю объектный файл `hello.o` на обработку компоновщику LD для создания исполняемого файла `hello`. Ключ `-o` указывает имя создаваемого исполняемого файла. После этого с помощью утилиты `ls` проверяю корректность выполнения команды (рис. 6).

```
shakira@localhost-live:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04 — sudo dnf in gcc  
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ ld -m elf_i386 hello.o -o hello  
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$
```

Рис.6 Передача объектного файла на обработку компоновщику

Я выполняю следующую команду, в результате которой исполняемый файл получит имя `main`, так как после ключа `-o` указано значение `main`. Объектный файл, использованный для создания этого исполняемого файла, называется `obj.o` (рис. 7).

```
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ ld -m elf_i386 hello.o -o hello  
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ ld -m elf_i386 obj.o -o main  
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$
```

Рис.7 Передача объектного файла на обработку компоновщику

4.5 Запуск исполняемого файла

Запускаю на выполнение созданный исполняемый файл `hello` (рис. 8).

```
shakira@localhost-live:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04 — sudo dnf in gcc  
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ ld -m elf_i386 hello.o -o hello  
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ ld -m elf_i386 obj.o -o main  
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ ls  
hello  hello.asm  hello.o  list.lst  main  obj.o  
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ ./hello  
Hello, world!  
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$
```

Рис.8 Запуск файла

4.6 Выполнение заданий для самостоятельной работы

С помощью утилиты `cp` создаю в текущем каталоге копию файла `hello.asm` с именем `lab04.asm` и проверяю правильность (рис. 9).

```
Hello, world!
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ cp hello.asm lab04.asm
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ ls
hello  hello.asm  hello.o  lab04.asm  list.lst  main  obj.o
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$
```

Рис.9 Создание копии файла

С помощью текстового редактора `mouespad` открываю файл `lab04.asm` и вношу изменения в программу так, чтобы она выводила мои имя и фамилию (рис. 10)

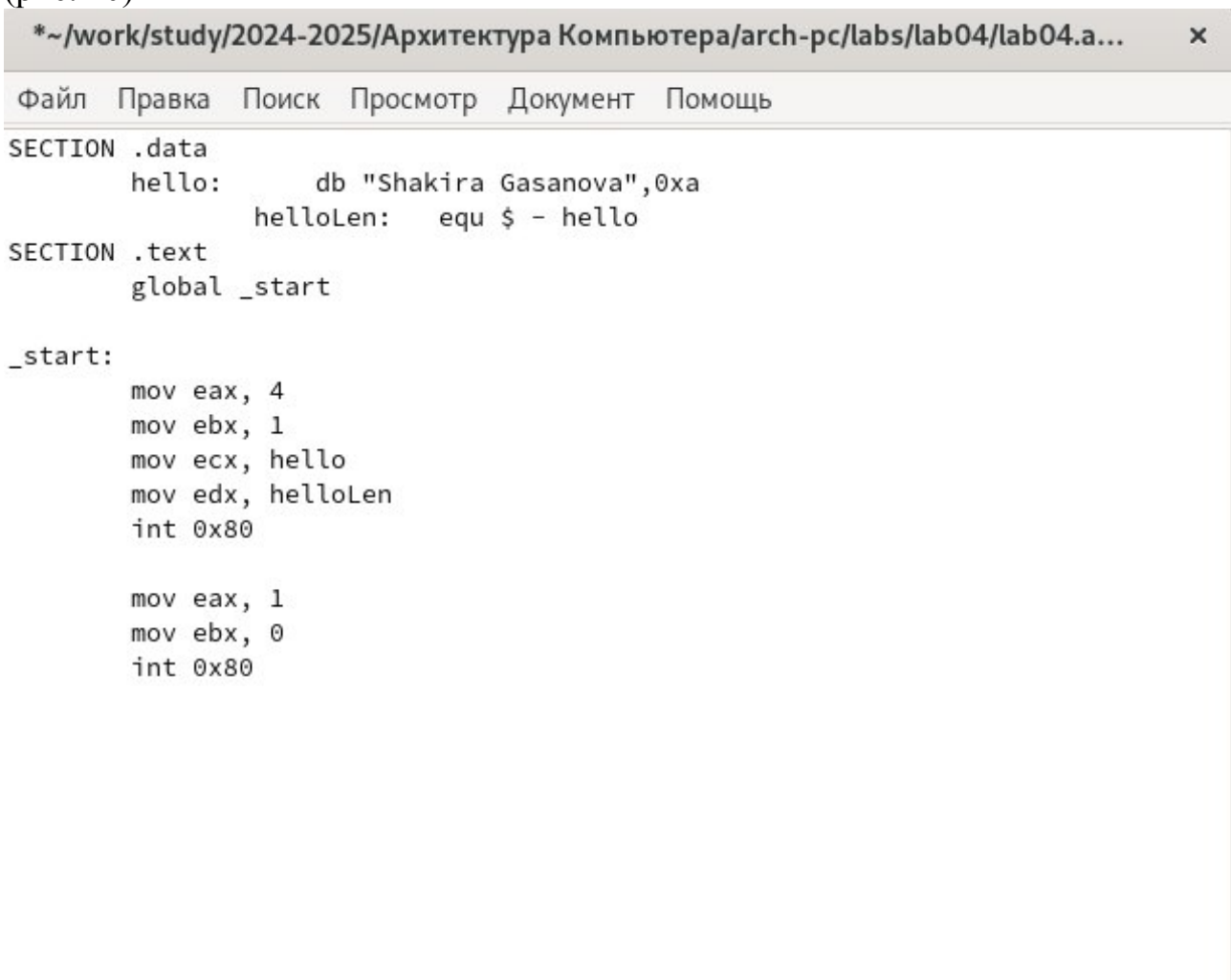


Рис. 10 Редактирование файла

Компилирую текст программы в объектный файл. Проверяю с помощью утилиты `ls`, что файл `lab04.o` создан (рис. 11).

```
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ nasm -f elf lab04.asm
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$ ls
hello  hello.asm  hello.o  lab04.asm  lab04.o  list.lst  main  obj.o
shakira@fedora:~/work/study/2024-2025/Архитектура Компьютера/arch-pc/labs/lab04$
```

Рис. 11 Компиляция текста программы

Передаю объектный файл lab04.o на обработку компоновщику LD, чтобы получить исполняемый файл lab04 (рис. 12).

```
shakira@fedora:~/work/study/2024-2025/Архитектура_Компьютера/arch-pc/labs/lab04$ ld -m elf_i386 lab04.o -o lab04
shakira@fedora:~/work/study/2024-2025/Архитектура_Компьютера/arch-pc/labs/lab04$
```

Рис. 12 Передача объектного файла на обработку компоновщику

Запускаю исполняемый файл lab04, на экран действительно выводятся мои имя и фамилия (рис. 13).

```
shakira@fedora:~/work/study/2024-2025/Архитектура_Компьютера/arch-pc/labs/lab04$ ./lab04
Shakira Gasanova
```

Рис. 13 Запуск файла

С помощью команд git add . и git commit добавляю файлы на GitHub, комментируя действие как добавление файлов для лабораторной работы №4 (рис. 14).

```
shakira@fedora:~/work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab04$ git add .
shakira@fedora:~/work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab04$ git commit -m "Add files to lab04"
[master b6f58c5] Add files to lab04
 9 files changed, 49 insertions(+)
 create mode 100755 labs/lab04/hello
 create mode 100644 labs/lab04/hello.asm
 create mode 100644 labs/lab04/hello.o
 create mode 100755 labs/lab04/lab04
 create mode 100644 labs/lab04/lab04.asm
 create mode 100644 labs/lab04/lab04.o
 create mode 100644 labs/lab04/list.lst
 create mode 100755 labs/lab04/main
 create mode 100644 labs/lab04/obj.o
shakira@fedora:~/work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab04$
```

Рис. 14 Добавление файлов на GitHub

Отправляю файлы на сервер с помощью команды git push (рис. 15).

```
shakira@fedora:~/work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab04$ git push -f origin master
Username for 'https://github.com': shakiragas
Password for 'https://shakiragas@github.com':
Перечисление объектов: 16, готово.
Подсчет объектов: 100% (16/16), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (13/13), готово.
Запись объектов: 100% (13/13), 2.95 КиБ | 2.95 МБ/с, готово.
Total 13 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To https://github.com/shakiragas/study_2024-2025_arh-pc.git
 20a873c..b6f58c5 master -> master
shakira@fedora:~/work/study/2024-2025/Архитектура_компьютера/arch-pc/labs/lab04$
```

Рис. 15 Отправка файлов

5 Выводы

При выполнении данной лабораторной работы я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

6 Источники

1. [Архитектура ЭВМ \(rudn.ru\)](http://rudn.ru)