

# **Отчёт по лабораторной работе №5**

**Дисциплина: архитектура компьютера**

Ганина Таисия Сергеевна, НКАбд-01-22

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>12</b>
	<b>Список литературы</b>	<b>13</b>

# Список иллюстраций

3.1	Рис. 1	6
4.1	Рис. 2	8
4.2	Рис. 3	8
4.3	Рис. 4	9
4.4	Рис. 5	9
4.5	Рис. 6	9
4.6	Рис. 7	9
4.7	Рис. 8	10
4.8	Рис. 9	10
4.9	Рис. 10	10
4.10	Рис. 11	10
4.11	Рис. 12	11
4.12	Рис. 13	11
4.13	Рис. 14	11

# 1 Цель работы

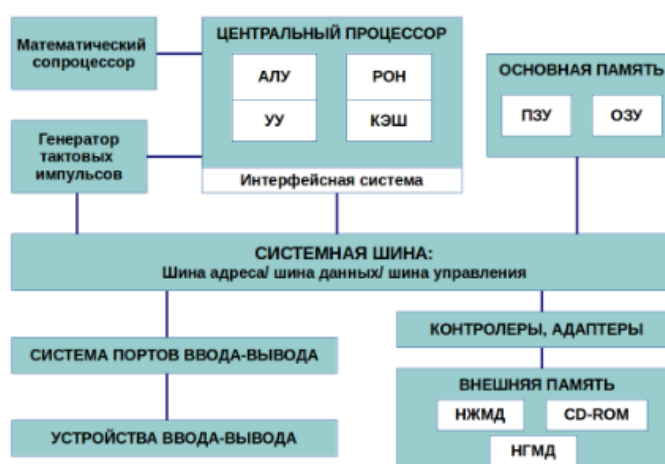
Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## 2 Задание

1. Создать каталог для работы с программами на языке ассемблера NASM.
2. Перейти в созданный каталог и создать текстовый файл с именем `hello.asm`
3. Открыть файл с помощью текстового редактора.
4. Внести в файл кода изменения и превратить его в объектный файл.
5. Скомпилировать исходный файл в `obj.o`, проверить, что всё выполнено.  
(есть `obj.o`, `list.lst`)
6. Скомпоновать файл, создать файл `main.o`, выяснить какое имя будет иметь исполняемый файл объектный, из которого он собран.
7. Запустить файл `hello`.
8. Создать копию файла `hello` с именем `lab5`.
9. Изменить текст так, чтобы выводилась строка с именем и фамилией.
10. Выполнить компоновку объектного файла и запустить получившийся исполняемый файл.

### 3 Теоретическое введение

Структурная схема ЭВМ (рис. 3.1).



3.1: Рис. 1

Основной задачей процессора является обработка информации, а также организация координации всех узлов компьютера. В состав центрального процессора (ЦП) входят следующие устройства:

- арифметико-логическое устройство (АЛУ) — выполняет логические и арифметические действия, необходимые для обработки информации, хранящейся в памяти;
- устройство управления (УУ) — обеспечивает управление и контроль всех устройств компьютера;
- регистры — сверхбыстрая оперативная память небольшого объема, входя-

щая в состав процессора, для временного хранения промежуточных результатов выполнения инструкций;

Регистры процессора делятся на два типа: регистры общего назначения и специальные регистры. Для того, чтобы писать программы на ассемблере, необходимо знать, какие регистры процессора существуют и как их можно использовать. Большинство команд в программах написанных на ассемблере используют регистры в качестве операндов. Практически все команды представляют собой преобразование данных хранящихся в регистрах процессора, это например пересылка данных между регистрами или между регистрами и памятью, преобразование (арифметические или логические операции) данных хранящихся в регистрах. Доступ к регистрам осуществляется не по адресам, как к основной памяти, а по именам. Каждый регистр процессора архитектуры x86 имеет свое название, состоящее из 2 или 3 букв латинского алфавита.

В качестве примера приведем названия основных регистров общего назначения именно эти регистры чаще всего используются при написании программ):

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

Например, AH (high AX) — старшие 8 бит регистра AX, AL (low AX) — младшие 8 бит регистра AX

## 4 Выполнение лабораторной работы

1. Создала каталог для работы с программами на языке ассемблера NASM (рис. 4.1).

```
[tsganina@fedora ~]$ cd ~/work/study/2022-2023/"Архитектура компьютера"/study_2022-2023_arh-pc/labs/lab05
[tsganina@fedora lab05]$ mkdir nasm_progs
[tsganina@fedora lab05]$ ls
nasm_progs  presentation  report
[tsganina@fedora lab05]$
```

4.1: Рис. 2

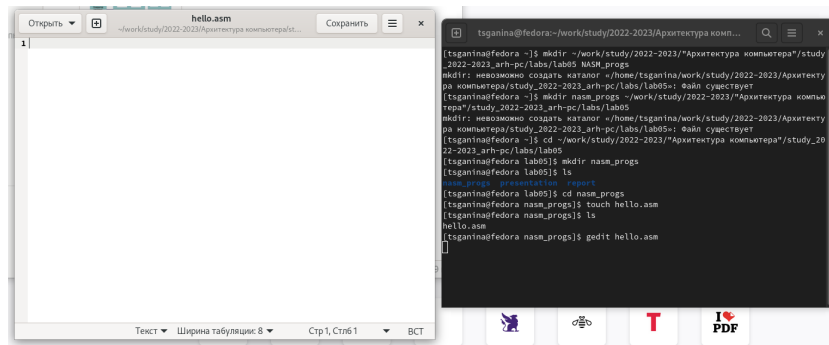
2. Перешла в созданный каталог и создала текстовый файл с именем hello.asm (рис. 4.2).

```
[tsganina@fedora lab05]$ cd nasm_progs
[tsganina@fedora nasm_progs]$ touch hello.asm
[tsganina@fedora nasm_progs]$ ls
hello.asm
[tsganina@fedora nasm_progs]$
```

4.2: Рис. 3

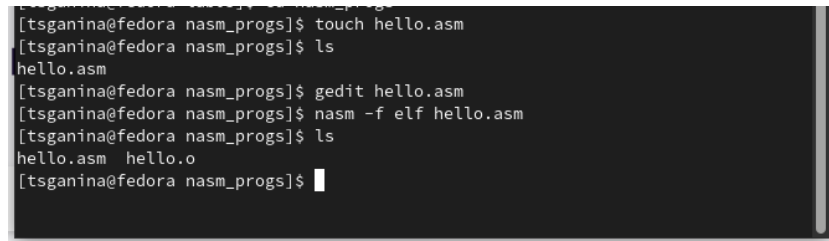
3. Открыла файл с помощью текстового редактора (рис. 4.3).





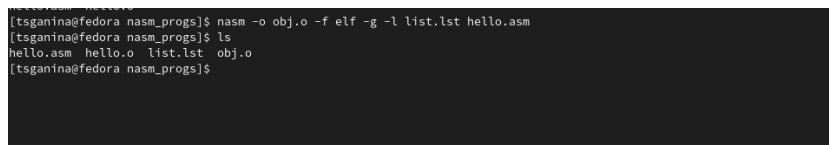
4.3: Рис. 4

4. Внесла в файл кода изменения и превратила его в объектный файл (рис. 4.4).



4.4: Рис. 5

5. Скомпилировала исходный файл в obj.o, проверила, что всё выполнено. (есть obj.o, list.lst) (рис. 4.5).



4.5: Рис. 6

6. Скомпоновала файл, создала файл main, выяснила какое имя будет иметь исполняемый файл и объектный, из которого он собран (рис. 4.6, 4.7).



4.6: Рис. 7

```
[tsganina@fedora nasm_progs]$ ld -m elf_i386 obj.o -o main
[tsganina@fedora nasm_progs]$ ls
hello hello.asm hello.o list.lst main obj.o
[tsganina@fedora nasm_progs]$
```

4.7: Рис. 8

У объектного файла имя obj.o, а у исполняемого файла - main.

7. Запустила файл hello (рис. 4.8).

```
[tsganina@fedora nasm_progs]$ ./hello
Hello world!
[tsganina@fedora nasm_progs]$
```

4.8: Рис. 9

8. Создала копию файла hello с именем lab5 (рис. 4.9).

```
[tsganina@fedora nasm_progs]$ cp hello.asm lab5.asm
[tsganina@fedora nasm_progs]$ ls
hello hello.asm hello.o lab5.asm list.lst main obj.o
[tsganina@fedora nasm_progs]$
```

4.9: Рис. 10

9. Изменила текст так, чтобы выводилась строка с именем и фамилией (рис. 4.10).

```
[tsganina@fedora nasm_progs]$ cp hello.asm lab5.asm
[tsganina@fedora nasm_progs]$ ls
hello hello.asm hello.o lab5.asm list.lst main obj.o
[tsganina@fedora nasm_progs]$
```

4.10: Рис. 11

10. Выполнила компоновку объектного файла и запустить получившийся исполняемый файл (рис. 4.11, 4.12, 4.13).

```

[tsganina@fedora nasm_progs]$ gedit lab5.asm
[tsganina@fedora nasm_progs]$ nasm -f elf lab5.asm
[tsganina@fedora nasm_progs]$ ls
hello hello.asm hello.o lab5.asm lab5.o list.lst main obj.o
[tsganina@fedora nasm_progs]$

```

4.11: Рис. 12

```

[tsganina@fedora nasm_progs]$ ld -m elf_i386 lab5.o -o lab5
[tsganina@fedora nasm_progs]$ ls
hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj.o
[tsganina@fedora nasm_progs]$

```

4.12: Рис. 13

```

hello hello.asm hello.o lab5 lab5.asm lab5.o list.lst main obj.o
[tsganina@fedora nasm_progs]$ ./lab5
Ganina Taisia!
[tsganina@fedora nasm_progs]$

```

4.13: Рис. 14

## 5 Выводы

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

## **Список литературы**