

# **Шаблон отчёта по лабораторной работе**

**Простейший вариант**

Сиссе Мохамед Ламин

# Содержание

1	Цель работы	5
2	Теоретическое введение	6
3	Выполнение лабораторной работы	8
4	Выводы	13
	Список литературы	14

## Список иллюстраций

3.1	Создайте каталог для работы . . . . .	8
3.2	вставить текст hello.asm . . . . .	9
3.3	Расширенный синтаксис командной строки NASM . . . . .	9
3.4	Запуск исполняемого файла . . . . .	10
3.5	копию файла . . . . .	10
3.6	текст программы . . . . .	11
3.7	lab5.asm . . . . .	12
3.8	Скопируйте файлы . . . . .	12

## **Список таблиц**

# 1 Цель работы

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

1. В каталоге `~/work/arch-pc/lab05` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab5.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab5.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab5.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab5.asm` в Ваш локальный репозиторий в каталог `~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05/`. Загрузите файлы на Github.

## 2 Теоретическое введение

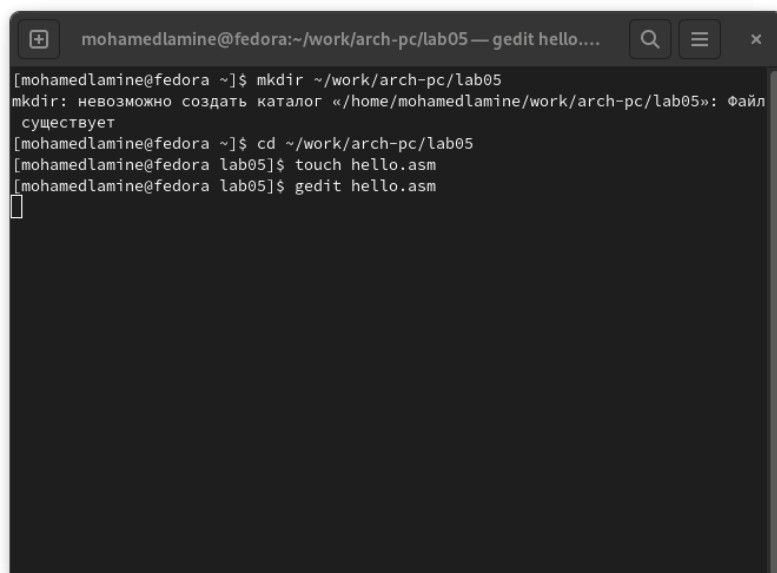
Основными функциональными элементами любой электронно-вычислительной машины (ЭВМ) являются центральный процессор, память и периферийные устройства (рис. 5.1). Взаимодействие этих устройств осуществляется через общую шину, к которой они подключены. Физически шина представляет собой большое количество проводников, соединяющих устройства друг с другом. В современных компьютерах проводники выполнены в виде электропроводящих дорожек на материнской (системной) плате. 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.

### 3 Выполнение лабораторной работы

1. Создайте каталог для работы с программами на языке ассемблера NASM.  
(рис. 3.1)



```
mohamedlamine@fedora:~/work/arch-pc/lab05 — gedit hello....
[mohamedlamine@fedora ~]$ mkdir ~/work/arch-pc/lab05
mkdir: невозможно создать каталог «/home/mohamedlamine/work/arch-pc/lab05»: файл
существует
[mohamedlamine@fedora ~]$ cd ~/work/arch-pc/lab05
[mohamedlamine@fedora lab05]$ touch hello.asm
[mohamedlamine@fedora lab05]$ gedit hello.asm
```

Рис. 3.1: Создайте каталог для работы

2. файл с помощью любого текстового редактора, например, gedit с помощью команды gedit hello.asm. (рис. 3.2)



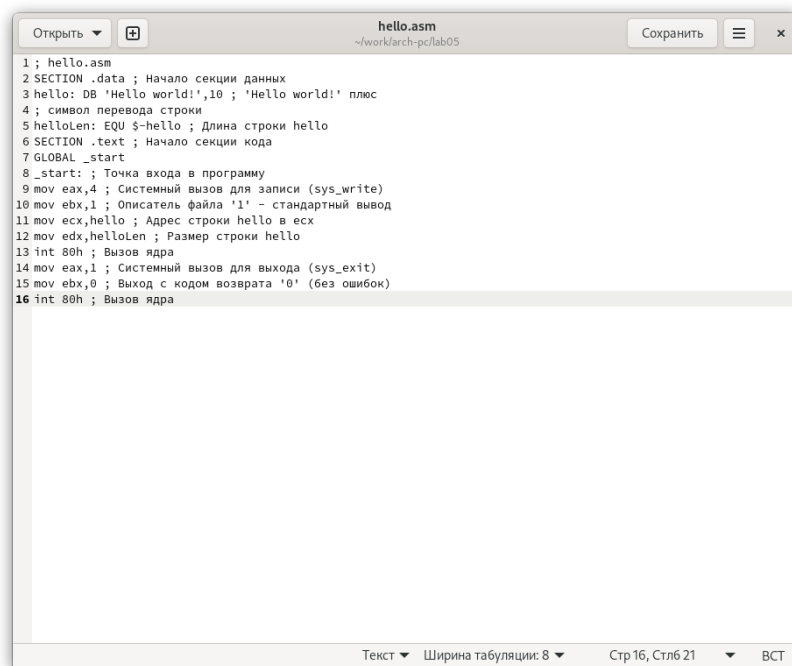


Рис. 3.2: вставить текст hello.asm

### 3. Транслятор NASMКомпоновщик LD. (рис. 3.3)

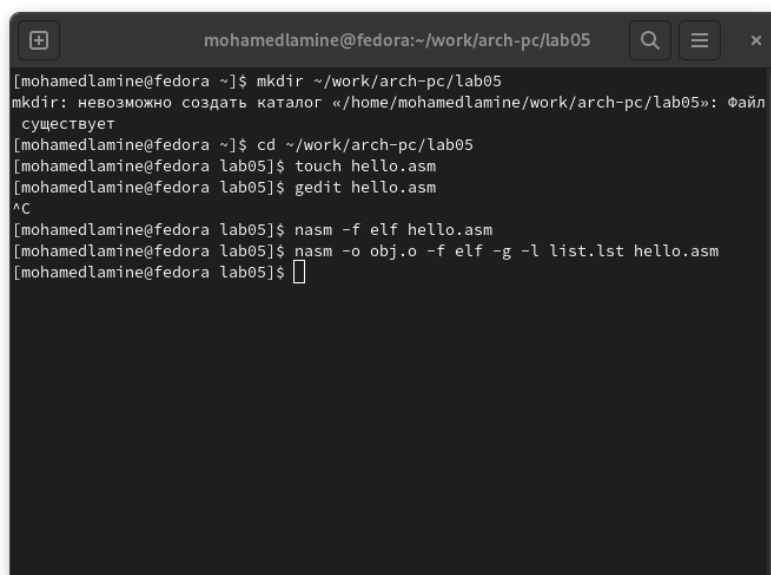
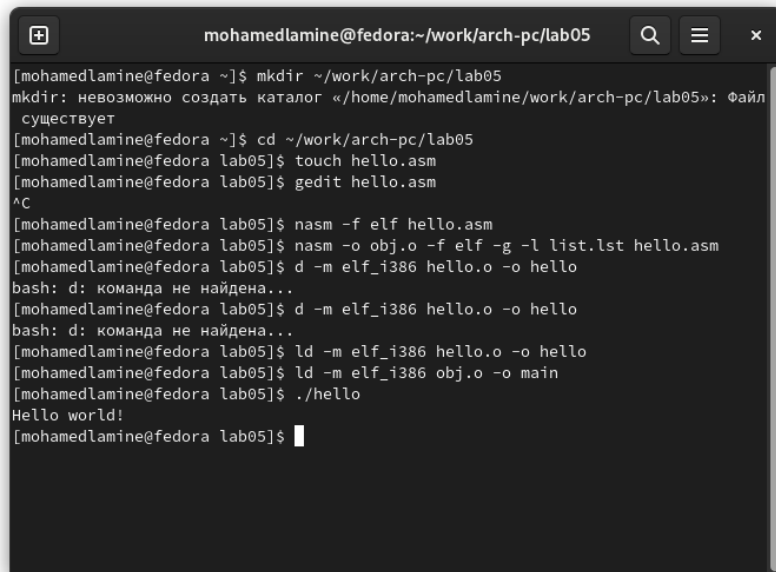


Рис. 3.3: Расширенный синтаксис командной строки NASM

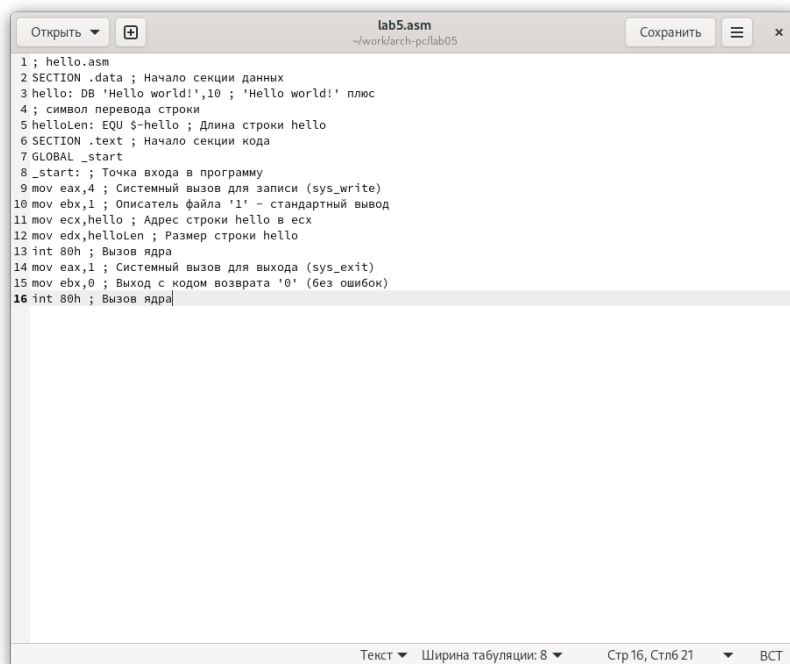
### 4. Компоновщик LD. (рис. 3.4)



```
mohamedlamine@fedora:~/work/arch-pc/lab05
[mohamedlamine@fedora ~]$ mkdir ~/work/arch-pc/lab05
mkdir: невозможно создать каталог «/home/mohamedlamine/work/arch-pc/lab05»: файл
существует
[mohamedlamine@fedora ~]$ cd ~/work/arch-pc/lab05
[mohamedlamine@fedora lab05]$ touch hello.asm
[mohamedlamine@fedora lab05]$ gedit hello.asm
^C
[mohamedlamine@fedora lab05]$ nasm -f elf hello.asm
[mohamedlamine@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[mohamedlamine@fedora lab05]$ d -m elf_i386 hello.o -o hello
bash: d: команда не найдена...
[mohamedlamine@fedora lab05]$ d -m elf_i386 hello.o -o hello
bash: d: команда не найдена...
[mohamedlamine@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[mohamedlamine@fedora lab05]$ ld -m elf_i386 obj.o -o main
[mohamedlamine@fedora lab05]$ ./hello
Hello world!
[mohamedlamine@fedora lab05]$
```

Рис. 3.4: Запуск исполняемого файла

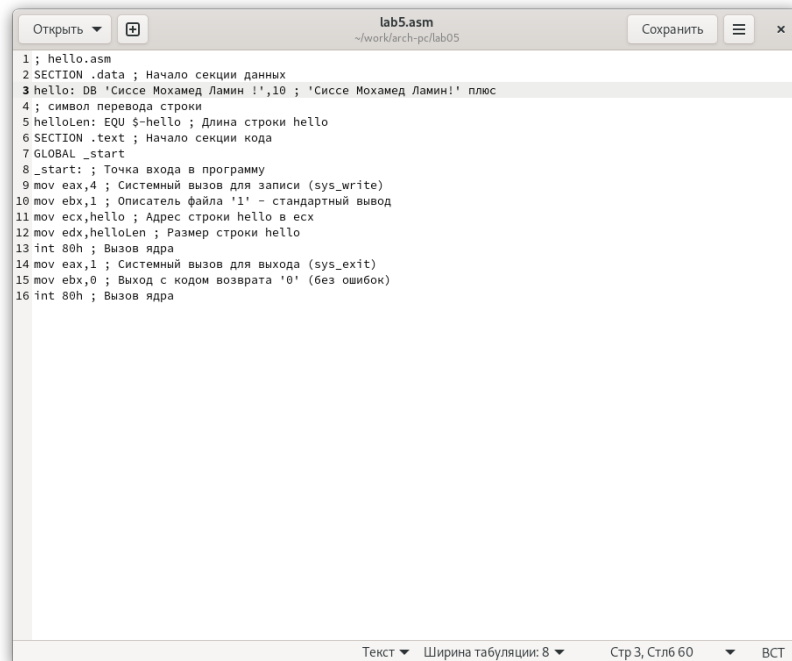
5. В каталоге `~/work/arch-pc/lab05` с помощью команды `cp` создайте копию файла `hello.asm` с именем `lab5.asm`. (рис. 3.5)



```
lab5.asm
~/work/arch-pc/lab05
1; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описание файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
12 mov edx,helloLen ; Размер строки hello
13 int 80h ; Вызов ядра
14 mov eax,1 ; Системный вызов для выхода (sys_exit)
15 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16 int 80h ; Вызов ядра
```

Рис. 3.5: копию файла

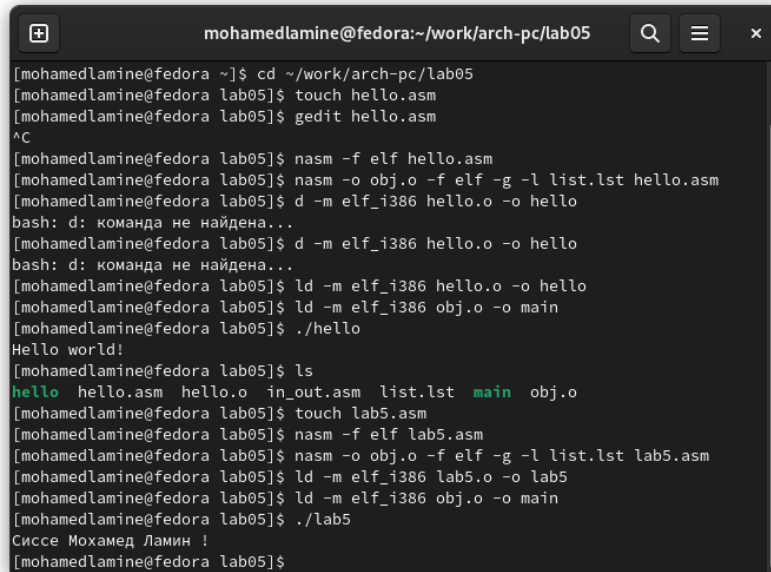
6. С помощью любого текстового редактора внесите изменения в текст программы в файле lab5.asm так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем. (рис. 3.6)



```
1; hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Сиссе Мохамед Ламин !',10 ; 'Сиссе Мохамед Ламин!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в есх
12 mov edx,helloLen ; Размер строки hello
13 int 80h ; Вызов ядра
14 mov eax,1 ; Системный вызов для выхода (sys_exit)
15 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16 int 80h ; Вызов ядра
```

Рис. 3.6: текст программы

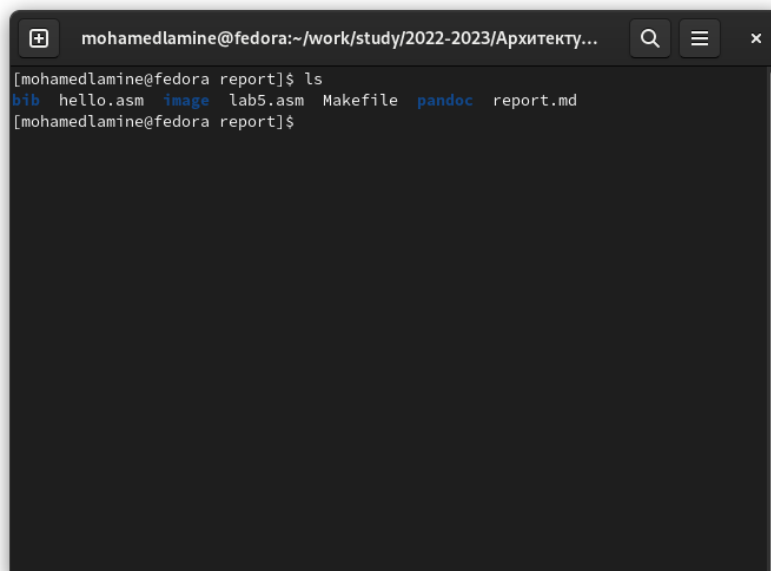
7. Оттранслируйте полученный текст программы lab5.asm в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл. (рис. 3.7)

A terminal window titled 'mohamedlamine@fedora:~/work/arch-pc/lab05'. The user enters several commands: 'cd ~/work/arch-pc/lab05', 'touch hello.asm', 'gedit hello.asm', 'nasm -f elf hello.asm', 'nasm -o obj.o -f elf -g -l list.lst hello.asm', 'd -m elf\_i386 hello.o -o hello' (which fails with 'bash: d: команда не найдена...'), 'ld -m elf\_i386 hello.o -o hello', 'ld -m elf\_i386 obj.o -o main', and './hello' (which outputs 'Hello world!'). Then the user runs 'ls' showing 'hello.asm', 'hello.o', 'in\_out.asm', 'list.lst', 'main', and 'obj.o'. Next, they create 'lab5.asm' and compile it with 'nasm -f elf lab5.asm', 'nasm -o obj.o -f elf -g -l list.lst lab5.asm', 'ld -m elf\_i386 lab5.o -o lab5', and 'ld -m elf\_i386 obj.o -o main'. Finally, they run './lab5' which outputs 'Сиссе Мохамед Ламин !'.

```
mohamedlamine@fedora:~/work/arch-pc/lab05
[mohamedlamine@fedora ~]$ cd ~/work/arch-pc/lab05
[mohamedlamine@fedora lab05]$ touch hello.asm
[mohamedlamine@fedora lab05]$ gedit hello.asm
^C
[mohamedlamine@fedora lab05]$ nasm -f elf hello.asm
[mohamedlamine@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[mohamedlamine@fedora lab05]$ d -m elf_i386 hello.o -o hello
bash: d: команда не найдена...
[mohamedlamine@fedora lab05]$ d -m elf_i386 hello.o -o hello
bash: d: команда не найдена...
[mohamedlamine@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[mohamedlamine@fedora lab05]$ ld -m elf_i386 obj.o -o main
[mohamedlamine@fedora lab05]$ ./hello
Hello world!
[mohamedlamine@fedora lab05]$ ls
hello.asm  hello.o  in_out.asm  list.lst  main  obj.o
[mohamedlamine@fedora lab05]$ touch lab5.asm
[mohamedlamine@fedora lab05]$ nasm -f elf lab5.asm
[mohamedlamine@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst lab5.asm
[mohamedlamine@fedora lab05]$ ld -m elf_i386 lab5.o -o lab5
[mohamedlamine@fedora lab05]$ ld -m elf_i386 obj.o -o main
[mohamedlamine@fedora lab05]$ ./lab5
Сиссе Мохамед Ламин !
[mohamedlamine@fedora lab05]$
```

Рис. 3.7: lab5.asm

8. Скопируйте файлы hello.asm и lab5.asm в Ваш локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05/, Загрузите файлы на Github. (рис .3.8)

A terminal window titled 'mohamedlamine@fedora:~/work/study/2022-2023/Архитекту...'. The user enters 'ls' in a directory named 'report', which shows 'bib', 'hello.asm', 'image', 'lab5.asm', 'Makefile', 'pandoc', and 'report.md'.

```
mohamedlamine@fedora:~/work/study/2022-2023/Архитекту...
[mohamedlamine@fedora report]$ ls
bib  hello.asm  image  lab5.asm  Makefile  pandoc  report.md
[mohamedlamine@fedora report]$
```

Рис. 3.8: Скопируйте файлы

## **4 Выводы**

В ходе этой лабораторной работы я приобрел практический навык в освоении процедур компиляции и ассемблера программ, написанных на ассемблере NASM.

# Список литературы

1. Расширенный ассемблер: NASM