

# Лабораторная работа № 5

## Создание и процесс обработки программ на языке ассемблера NASM

Шулуужук Айраана В.

### Содержание

1	Цель работы .....	1
2	Задание.....	1
3	Выполнение лабораторной работы .....	1
3.1	Программа Hello world! .....	1
3.2	Транслятор NASM .....	2
3.3	Расширенный синтаксис командной строки NASM .....	3
3.4	Компоновщик LD.....	4
3.5	Запуск исполняемого файла .....	4
4	Выполнение самостоятельной работы.....	5
4.1	Порядок выполнения самостоятельной работы .....	5
5	Выводы.....	7

### 1 Цель работы

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

### 2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

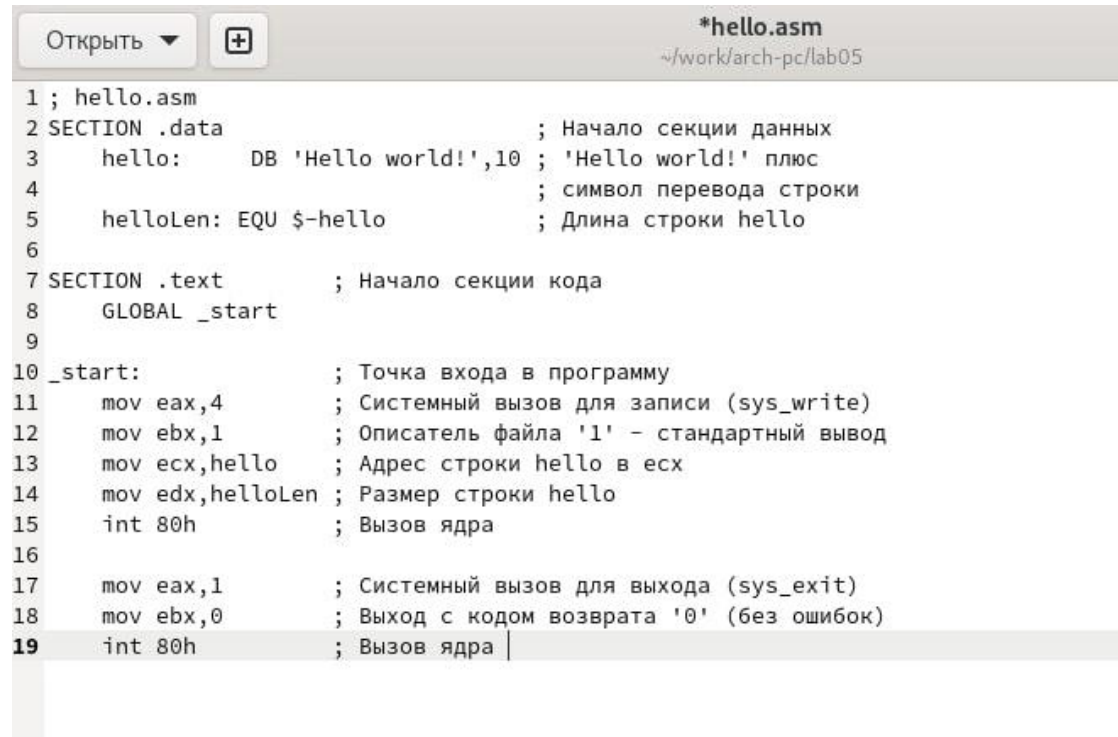
### 3 Выполнение лабораторной работы

#### 3.1 Программа Hello world!

Создаем каталог для работы с программами на языке ассемблера NASM.

Переходим в созданный каталог и создаем текстовый файл с именем `hello.asm`, далее открываем этот файл с помощью текстового редактора.

Введем текст программы (рис. 1)



```
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
7 SECTION .text                ; Начало секции кода
8     GLOBAL _start
9
10 _start:                    ; Точка входа в программу
11     mov eax,4              ; Системный вызов для записи (sys_write)
12     mov ebx,1              ; Описатель файла '1' - стандартный вывод
13     mov ecx,hello          ; Адрес строки hello в есх
14     mov edx,helloLen       ; Размер строки hello
15     int 80h                ; Вызов ядра
16
17     mov eax,1              ; Системный вызов для выхода (sys_exit)
18     mov ebx,0              ; Выход с кодом возврата '0' (без ошибок)
19     int 80h                ; Вызов ядра
```

Рис. 1: файл `hello.asm` в текстовом редакторе

## 3.2 Транслятор NASM

Скомпилируем текст программы «Hello World». Введем следующую команду:

```
nasm -f elf hello.asm
```

Транслятор преобразует текст программы из файла `hello.asm` в объектный код, который запишется в файл `hello.o` (рис. 2)

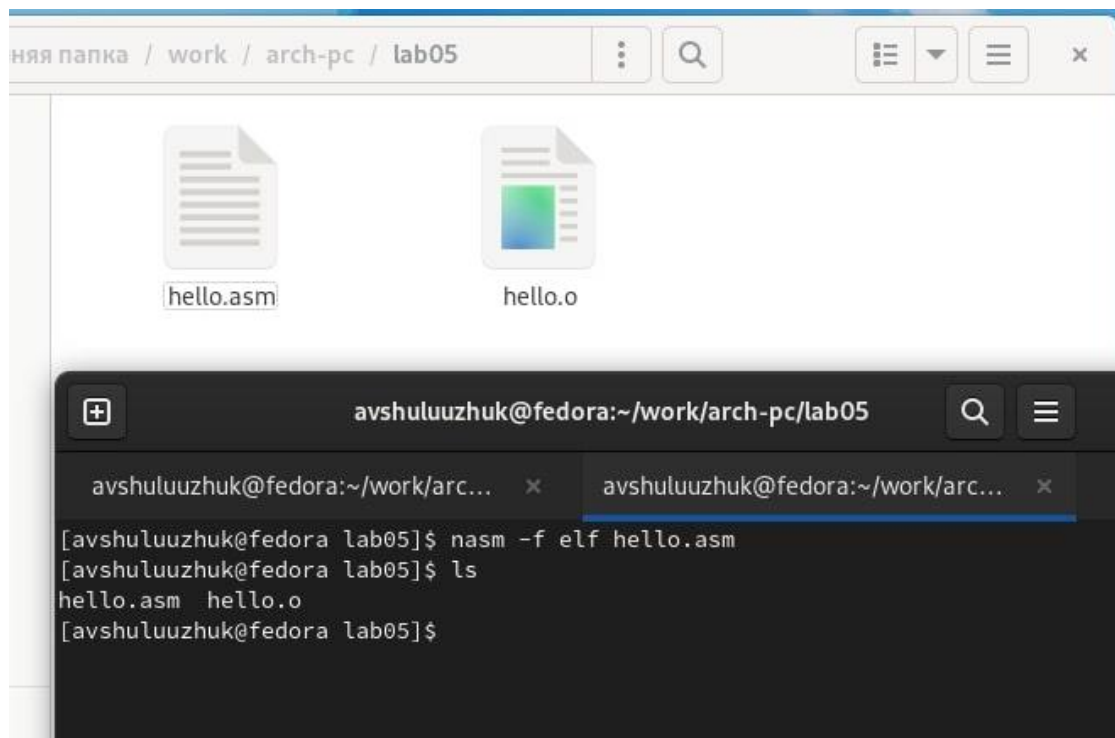


Рис. 2: объектный файл `hello.o`

### 3.3 Расширенный синтаксис командной строки NASM

Выполним следующую команду:

```
nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Эта команда скомпилирует файл `hello.asm` в `obj.o`. Также будет создан файл листинга `list.lst`. Далее проверим созданные файлы (рис. 3)

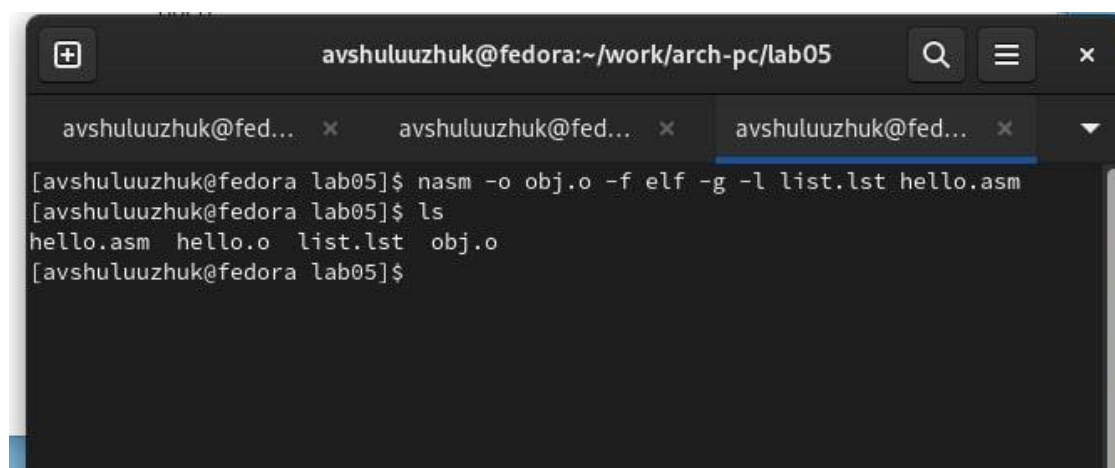


Рис. 3: созданные файлы `obj.o` и `list.lst`

### 3.4 Компоновщик LD

1. Передаем объектный файл на обработку компоновщику:

```
ld -m elf_i386 hello.o -o hello
```

Будет создан файл hello

2. Выполняя следующую команду:

```
ld -m elf_i386 obj.o -o main
```

из объектного файла obj.o будет создан исполняемый файл main (рис. 4)

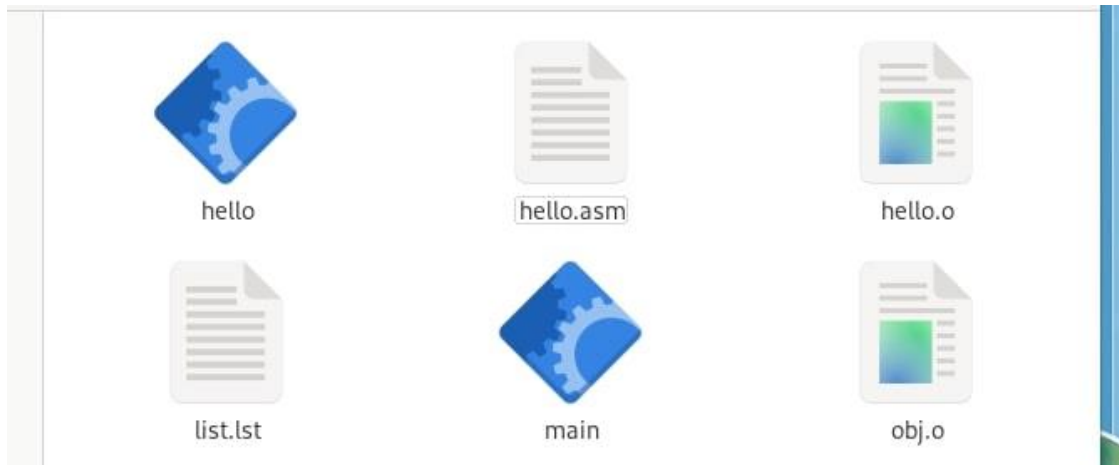


Рис. 4: созданные файлы

### 3.5 Запуск исполняемого файла

Набрав команду:

```
./hello
```

Запустим на выполнение созданный исполняемый файл (рис. 5)

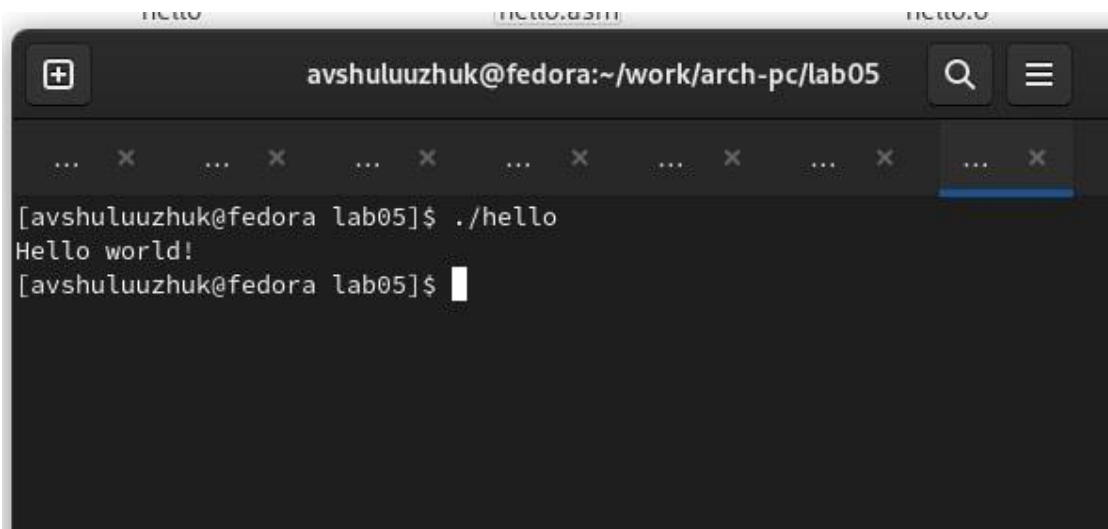
A screenshot of a terminal window with a dark background. The title bar at the top shows the user 'avshuluuzhuk' and the directory '~/work/arch-pc/lab05'. The terminal content shows a prompt '[avshuluuzhuk@fedora lab05]\$' followed by the command './hello', the output 'Hello world!', and another prompt '[avshuluuzhuk@fedora lab05]\$' with a cursor. The window has several tabs at the top, each with a close button 'x'.

Рис. 5: вывод программы "Hello world"

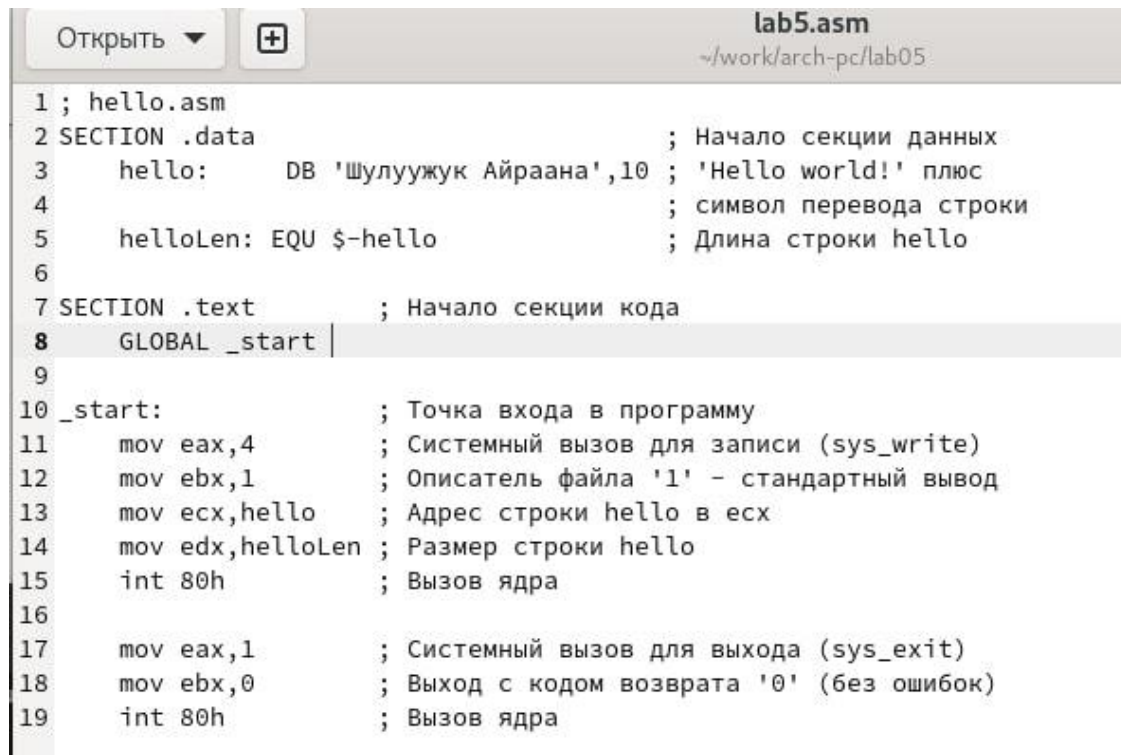
## 4 Выполнение самостоятельной работы

задание для самостоятельной работы: вывести фамилию и имя

1. Внести изменения в тексте программы с помощью текстового редактора. Вместо Hello world! вставить фамилию и имя.
2. Оттранслировать текст программы в объектный файл. Выполнить компоновку объектного файла и запустить файл
3. Загрузить файлы на Github

### 4.1 Порядок выполнения самостоятельной работы

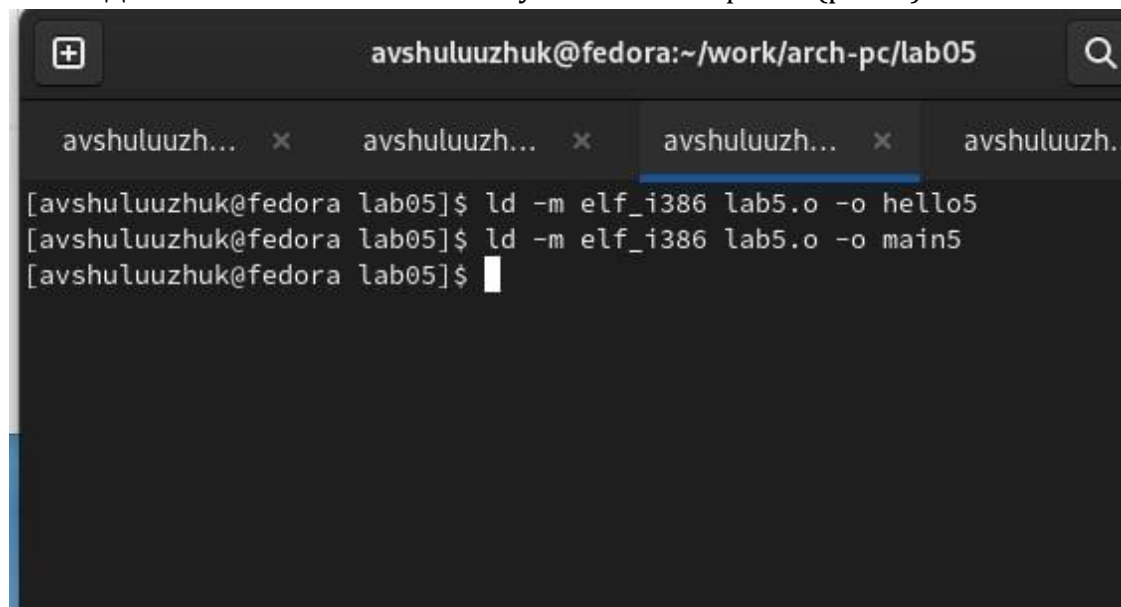
1. Создаем копию файла hello.asm с именем lab5.asm
2. С помощью текстового редактора внесем изменения в текст программы в файле ab5.asm, вносим фамилию и имя (рис. 6)



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

Рис. 6: файл *lab5.asm*

3. Оттранслируем полученный текст программы *lab5.asm* в объектный файл. Далее выполним компоновку объектного файла (рис. 7)

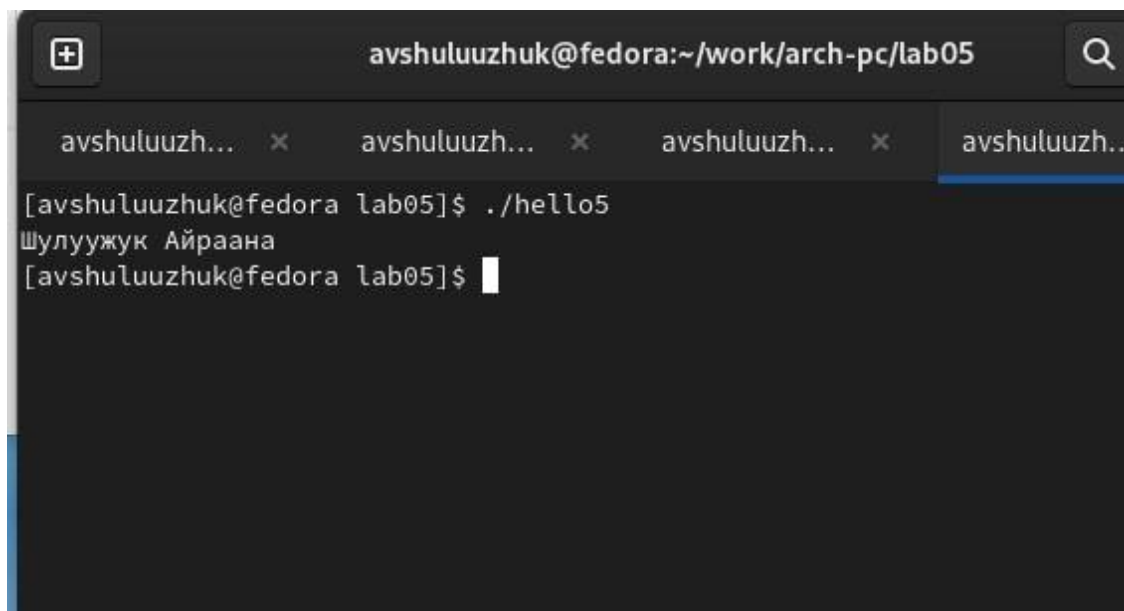


```
avshuluuzhuk@fedora:~/work/arch-pc/lab05
avshuluuzh... x avshuluuzh... x avshuluuzh... x avshuluuzh..
[avshuluuzhuk@fedora lab05]$ ld -m elf_i386 lab5.o -o hello5
[avshuluuzhuk@fedora lab05]$ ld -m elf_i386 lab5.o -o main5
[avshuluuzhuk@fedora lab05]$
```

Рис. 7: компоновка файла *lab5.o*

в итоге получаем файлы *hello5* и *main5*

4. Запускаем получившийся файл с выводом имени и фамилии (рис. 8)

A terminal window titled 'avshuluuzhuk@fedora:~/work/arch-pc/lab05'. The window has four tabs, all labeled 'avshuluuzh...'. The terminal shows the command './hello5' being executed, which outputs 'Шулуужук Айраана' (Shuluuzhuk Ayraana). The prompt '[avshuluuzhuk@fedora lab05]\$' is visible at the end of the line.

```
avshuluuzhuk@fedora:~/work/arch-pc/lab05
[avshuluuzhuk@fedora lab05]$ ./hello5
Шулуужук Айраана
[avshuluuzhuk@fedora lab05]$
```

Рис. 8: вывод фамилии и имени

5. Копируем файлы `hello.asm` и `lab5.asm` в локальный репозиторий и загружаем файла на github

## 5 Выводы

Были освоены процедуры компиляции и сборки текстов программ "Hello world!", а также текст программы с выводом имени и фамилии