

# Отчёта по лабораторной работе №4

## Дисциплина: Архитектура компьютера

Камалиева Лия Дамировна.

### Содержание

1	Цель работы .....	1
2	Задание.....	1
3	Теоретическое введение .....	2
4	Выполнение лабораторной работы .....	2
4.1	1.1 Программа Hello world!.....	2
4.2	1.2 Транслятор NASM.....	2
4.3	1.3 Расширенный синтаксис командной строки NASM .....	3
4.4	1.4 Компоновщик LD .....	3
4.5	1.5 Запуск исполняемого файла.....	4
4.6	1.6 Задание для самостоятельной работы.....	4
4.6.1	№1 .....	4
4.6.2	№2 .....	4
4.6.3	№3 .....	4
4.6.4	№4 .....	5
5	Выводы.....	6
	Список литературы.....	6

### 1 Цель работы

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

### 2 Задание

1.1 Программа Hello world! 1.2 Транслятор NASM 1.3 Расширенный синтаксис командной строки NASM 1.4 Компоновщик LD 1.5 Запуск исполняемого файла 1.6 Задание для самостоятельной работы

### 3 Теоретическое введение

NASM (Netwide Assembler) — свободный (LGPL и лицензия BSD) ассемблер для архитектуры Intel x86. Используется для написания 16-, 32- и 64-разрядных программ.

## 4 Выполнение лабораторной работы

### 4.1 1.1 Программа Hello world!

Шаг 1. Создаем новый каталог lab04

```
ldkamalieva@ldkamalieva-VirtualBox:~$ mkdir -p ~/work/arch-pc/lab04
```

*lab04 рис.4.1*

Шаг 2. При помощи команды touch создаем файл hello.asm

```
ldkamalieva@ldkamalieva-VirtualBox:~$ cd ~/work/arch-pc/lab04
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ touch hello.asm
```

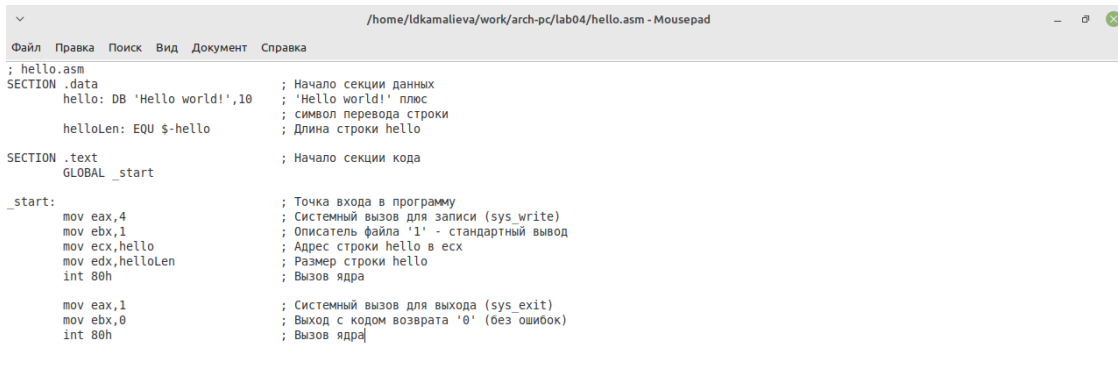
*создание файла hello.asm рис.4.2*

Шаг 3. Открываем его в текстовом редакторе

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ mousepad hello.asm
```

*открываем в mousepad рис.4.3*

Шаг 4. Вводим в mousepad программу hello world!



```

/home/ldkamalieva/work/arch-pc/lab04/hello.asm - Mousepad
Файл  Правка  Поиск  Вид  Документ  Справка
; hello.asm
SECTION .data
    hello: DB 'Hello world!',10      ; 'Hello world!' плюс
                                       ; символ перевода строки
    helloLen: EQU $-hello           ; Длина строки hello
SECTION .text
    GLOBAL _start                  ; Начало секции кода
_start:
    mov eax,4                      ; Точка входа в программу
    mov ebx,1                      ; Системный вызов для записи (sys_write)
    mov ecx,hello                  ; Описатель файла '1' - стандартный вывод
    mov edx,helloLen               ; Адрес строки hello в ecx
    int 80h                       ; Размер строки hello
                                       ; 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
```

*программа hello world! рис.4.4*

### 4.2 1.2 Транслятор NASM

Шаг 1. Компилируем наш код при помощи команды `nasm -f elf hello.asm`

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
```

*компилируем код рис.4.5*

Шаг 2. Сделаем проверку

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
```

*команда ls рис.4.6*

### 4.3 1.3 Расширенный синтаксис командной строки NASM

Шаг 1. скомпилируем наш файл hello.asm в obj.o при помощи команды `nasm -o obj.o -f elf -g -l list.lst hello.asm` и сделаем проверку с помощью команды `ls`

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g
-l list.lst hello.asm
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
```

*компиляция и проверка рис.4.7*

### 4.4 1.4 Компоновщик LD

Шаг 1. Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику при помощи команды `ld -m elf_i386 hello.o -o hello`

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o
-o hello
```

*передача объектного файла рис.4.8*

Шаг 2. Проверяем созданся ли файл hello

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

*ls рис.4.9*

Шаг 3. Далее вводим команду `ld -m elf_i386 obj.o -o main` и создаем файл main, который также проверяем с помощью команды `ls`

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o
main
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$
```

*создание файла main рис.4.10*

## 4.5 1.5 Запуск исполняемого файла

Шаг 1. Запускаем на выполнение созданный файл, находящийся в текущем каталоге, при помощи команды `./hello`

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ./hello
Hello world!
```

*запускаем на выполнение файл рис.4.11*

## 4.6 1.6 Задание для самостоятельной работы

### 4.6.1 №1

Шаг 1. В каталоге `~/work/arch-pc/lab04` с помощью команды `cp` создаем копию файла `hello.asm` с именем `lab4.asm`

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
```

*создание lab4.asm рис.4.12*

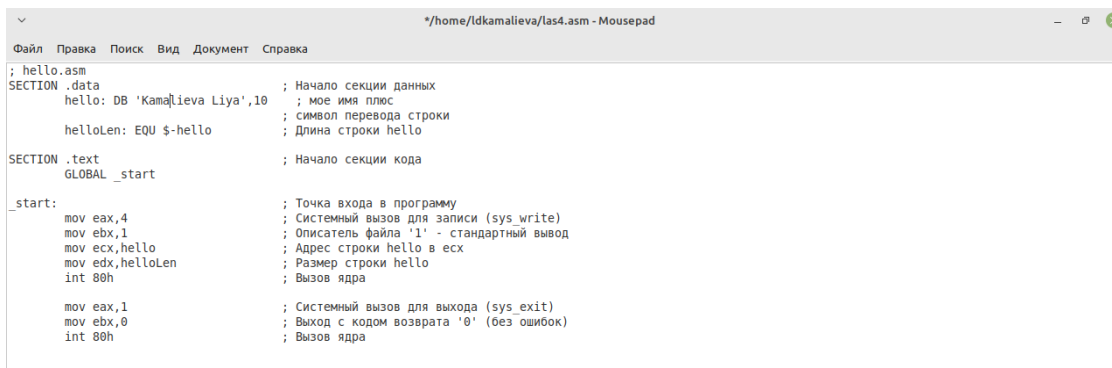
Шаг 2. Делаем проверку

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm list.lst main obj.o
```

*проверка рис.4.13*

### 4.6.2 №2

Шаг 1. С помощью текстового редактора в программе `Hello world!` заменяем на `Kamalieva Liya`



```

; hello.asm
SECTION .data
    hello: DB 'Kamalieva Liya',10
    helloLen: EQU $-hello
SECTION .text
    GLOBAL _start
_start:
    mov eax,4
    mov ebx,1
    mov ecx,hello
    mov edx,helloLen
    int 80h
    mov eax,1
    mov ebx,0
    int 80h
```

*замена рис.4.14*

### 4.6.3 №3

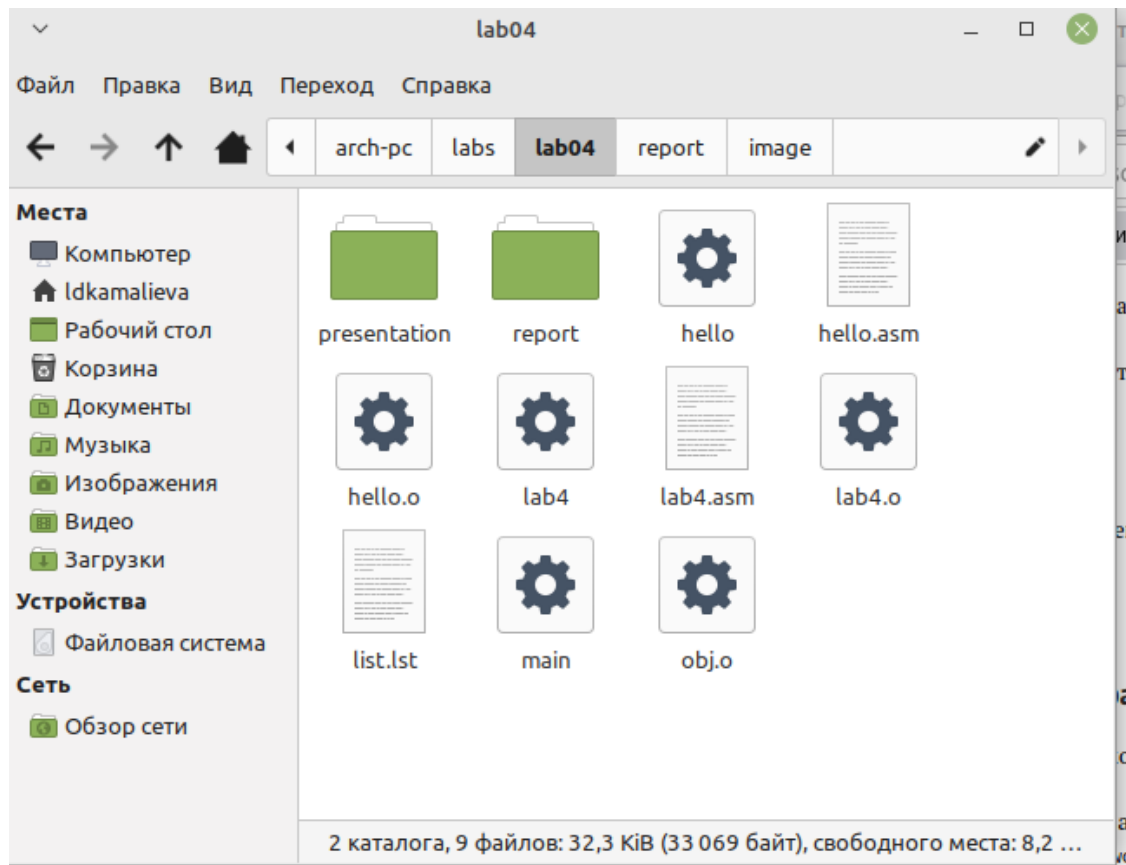
Шаг 1. Делаем все действия, что, и с программой `Hello world!`, также делаем проверку

```
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4.asm lab4.o list.lst main obj.o
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -
o lab4
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$ ./lab4
Kamaliev Liya
ldkamalieva@ldkamalieva-VirtualBox:~/work/arch-pc/lab04$
```

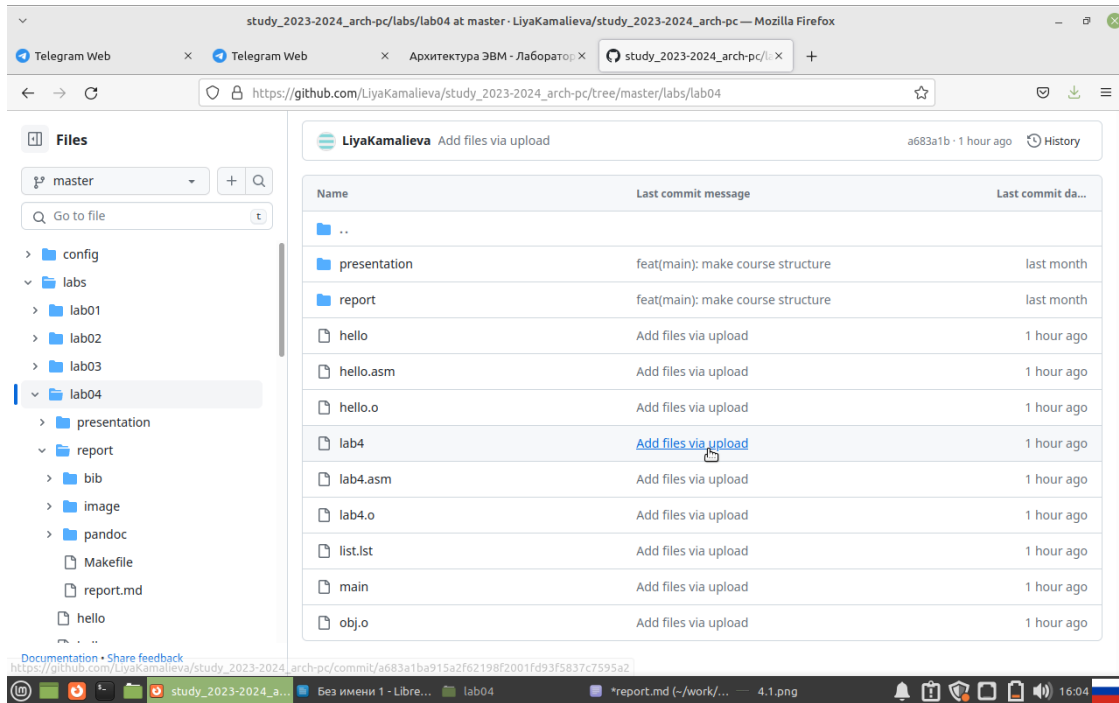
настройка программы рис.4.15

#### 4.6.4 №4

Шаг 1. Скопируем файлы hello.asm и lab4.asm в локальный репозиторий в каталог ~/work/study/2023-2024/“Архитектура компьютера”/arch-pc/labs/lab04. Загрузим файлы на Github



репозиторий рис.4.16



*github пус.4.17*

## 5 Выводы

Мы изучили основные команды для работы с языком NASM.

## Список литературы