

**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: Ванюшкина Т.В.

Группа: НКАбд-01-24

Студ.билет: 1132246713

**МОСКВА**

2024г

## СОДЕРЖАНИЕ

1. Цель работы.....	3
2. Теоретическое введение.....	4
3. Выполнение лабораторной работы.....	5-9
3.1 Программа Hello world!.....	5-6
3.2 Транслятор NASM.....	6
3.3 Расширенный синтаксис командной строки NASM.....	6
3.4 компоновщик LD.....	6-7
3.5 Запуск исполняемого файла.....	7
3.6 Задания для самостоятельной работы.....	7-10
4. Выводы.....	11
5. Список литературы.....	12

## **1. Цель работы**

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

## 2. Теоретическое введение

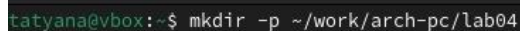
**Язык ассемблера** — представление команд процессора в виде, доступном для чтения человеком. Язык ассемблера считается языком программирования низкого уровня в противовес высокоуровневым языкам, не привязанным к конкретной реализации вычислительной системы. Программы, написанные на языке ассемблера, однозначным образом переводятся в инструкции конкретного процессора и в большинстве случаев не могут быть перенесены без значительных изменений для запуска на машине с другой системой команд. Ассемблером также называется программа-компилятор, преобразующая код на языке ассемблера в машинный код. Программа, выполняющая обратную задачу, называется дизассемблером

**NASM** – свободный ассемблер для архитектуры intel x86. Используется для написания 16-, 32- и 64-разрядных программ

### 3. Выполнение лабораторной работы

#### 3.1 Программа Hello world!


Создаю каталог для работы с программами на языке ассемблера NASM:



```
tatyana@vbox:~$ mkdir -p ~/work/arch-pc/lab04
```

Рис.1

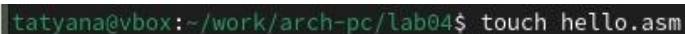
Перехожу в созданный каталог:



```
tatyana@vbox:~$ cd ~/work/arch-pc/lab04  
tatyana@vbox:~/work/arch-pc/lab04$
```

Рис .2

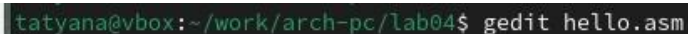
Создаю текстовый файл с именем hello.asm:



```
tatyana@vbox:~/work/arch-pc/lab04$ touch hello.asm
```

Рис.3

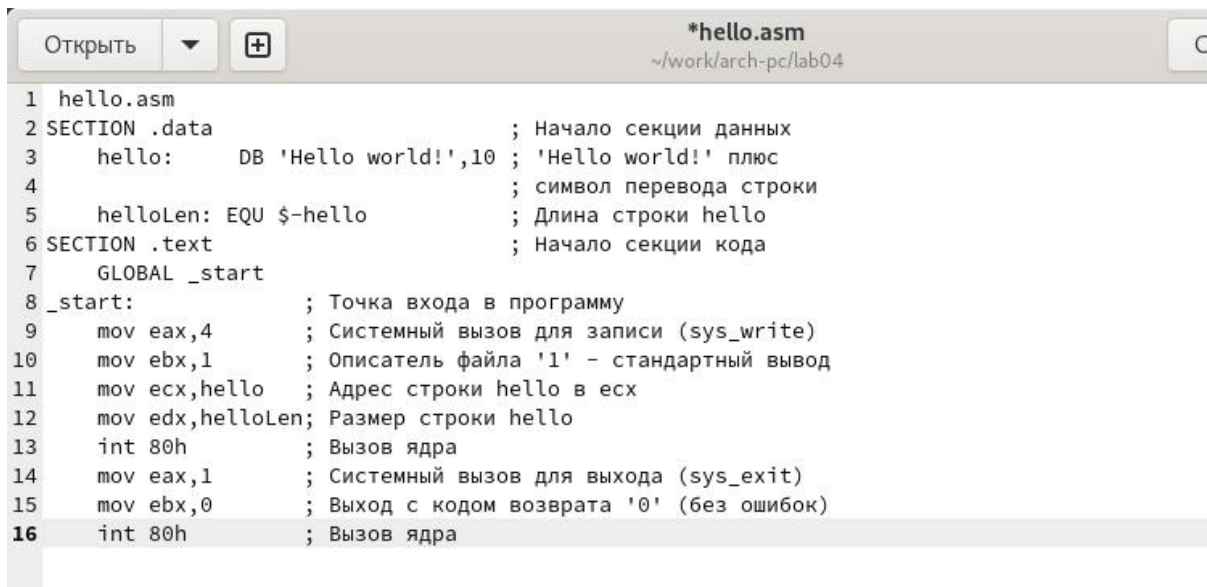
Открываю этот файл с помощью текстового редактора gedit:



```
tatyana@vbox:~/work/arch-pc/lab04$ gedit hello.asm
```

Рис.4

Ввожу в него следующий текст:

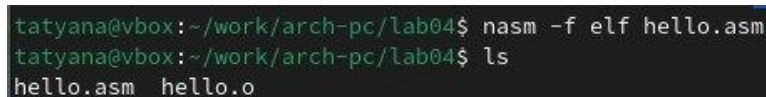


```
1 hello.asm
2 SECTION .data                ; Начало секции данных
3     hello:    DB 'Hello world!',10 ; 'Hello world!' плюс
4                                     ; символ перевода строки
5     helloLen: EQU $-hello      ; Длина строки hello
6 SECTION .text                ; Начало секции кода
7     GLOBAL _start
8 _start:                      ; Точка входа в программу
9     mov eax,4                ; Системный вызов для записи (sys_write)
10    mov ebx,1                ; Описатель 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.5

## 3.2. Транслятор NASM

NASM превращает текст программы в объектный код. Для компиляции приведённого выше текста программы «Hello World» пишу команду `nasm -f elf hello.asm` и с помощью команды `ls` проверяю, что объектный файл был создан:

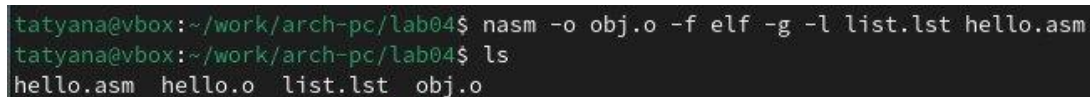


```
tatyana@vbox:~/work/arch-pc/lab04$ nasm -f elf hello.asm
tatyana@vbox:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
```

Рис.6

## 3.3. Расширенный синтаксис командной строки NASM

Выполняю следующую команду и с помощью команды `ls` проверяю, что файлы были созданы:



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

Рис.7

## 3.4. Компоновщик LD

Объектный файл передаю на обработку компоновщику и с помощью команды ls проверяю, что исполняемый файл hello был создан.:

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

Рис.8

Выполняю следующую команду:

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

Рис.9

### 3.5 Запуск исполняемого файла

Выполняю следующую команду:

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

Рис.10

### 3.6 Задание для самостоятельной работы

В каталоге ~/work/arch-pc/lab04 с помощью команды cp создаю копию файла hello.asm с именем lab4.asm:

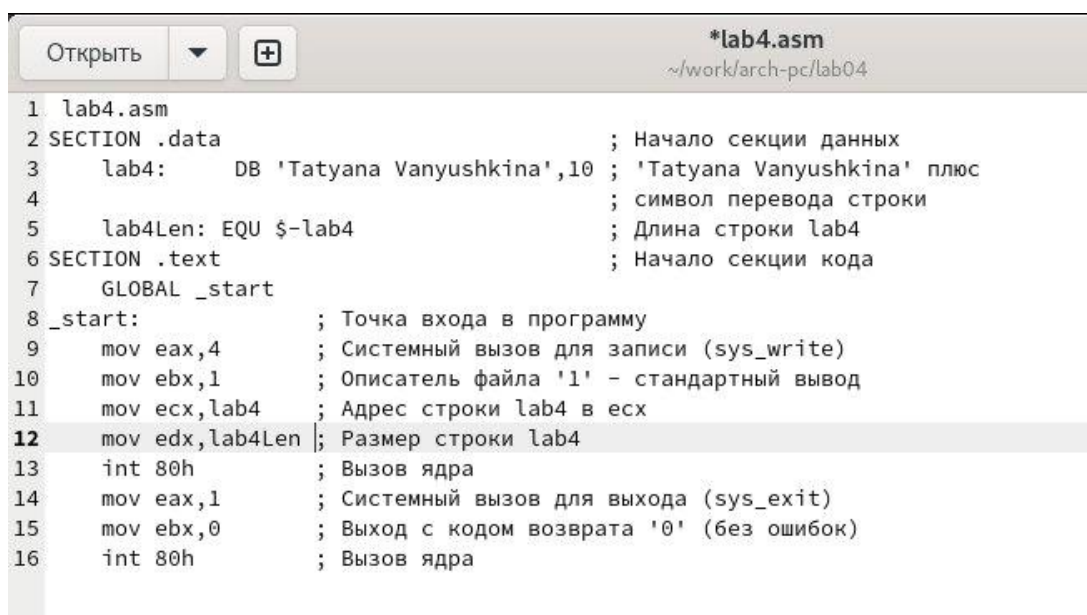
```
tatyana@vbox:~/work/arch-pc/lab04$ cp hello.asm lab4.asm
```

Рис.11

2. С помощью текстового редактора gedit вношу изменения в текст программы в файле lab4.asm так, чтобы вместо Hello world! на экран выводилась строка с моими фамилией и именем:

```
tatyana@vbox:~/work/arch-pc/lab04$ gedit lab4.asm
```

Рис.12



```
*lab4.asm
~/work/arch-pc/lab04

1 lab4.asm
2 SECTION .data                ; Начало секции данных
3     lab4:    DB 'Tatyana Vanyushkina',10 ; 'Tatyana Vanyushkina' плюс
4                                           ; символ перевода строки
5     lab4Len: EQU $-lab4        ; Длина строки lab4
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,lab4              ; Адрес строки lab4 в ecx
12    mov edx,lab4Len           ; Размер строки lab4
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.13

Компилирую текст программы в объектный файл:

```
tatyana@vbox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
```

Рис.14

С помощью команды ls проверяю, что файл был создан:

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

Рис.15

Выполняю следующую команду:

```
tatyana@vbox:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst lab4.asm
```

Рис.16

С помощью команды ls проверяю, что файлы были созданы:

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

Рис.17



Объектный файл передаю на обработку компоновщику:

```
tatyana@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
```

Рис.18

Проверяю с помощью команды ls:

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

Рис.18

Выполняю следующую команду:

```
tatyana@vbox:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
tatyana@vbox:~/work/arch-pc/lab04$ ls
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
```

Рис.19

Запускаю файл:

```
tatyana@vbox:~/work/arch-pc/lab04$ ./lab4
Tatyana Vanyushkina
```

Рис.20

Копирую файлы hello.asm и lab4.asm в локальный репозиторий в каталог ~/work/study/2023-2024/"Архитектура компьютера"/arch-pc/labs/lab04/:

```
tatyana@vbox:~/work/arch-pc/lab04$ cp * ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/lab04
```

Рис.21

Проверяю с помощью команды ls:

```
tatyana@vbox:~/work/arch-pc/lab04$ ls ~/work/study/2024-2025/"Архитектура компьютера"/arch-pc/lab04
hello hello.asm hello.o lab4 lab4.asm lab4.o list.lst main obj.o
tatyana@vbox:~/work/arch-pc/lab04$
```

Рис.22

Удаляю лишние файлы с помощью команды rm:

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

Рис.23

Добавляю файлы на GitHub:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ git add .  
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ git commit -m "Add fales for lab04"  
[master 3174892] Add fales for lab04  
9 files changed, 51 insertions(+)
```

Рис. 24

Отправляю файлы на сервер:

```
tatyana@vbox:~/work/study/2024-2025/Архитектура компьютера/arch-pc/lab04$ git push  
Перечисление объектов: 13, готово.  
Подсчет объектов: 100% (13/13), готово.  
При сжатии изменений используется до 2 потоков  
Сжатие объектов: 100% (12/12), готово.  
Запись объектов: 100% (12/12), 3.35 КиБ | 3.35 МБ/с, готово.  
Total 12 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (6/6), completed with 1 local object.  
To github.com:tatyana952/study_2024-2025_arh-pc.git  
69dc912..3174892 master -> master
```

Рис.25

**4.Вывод:** В ходе выполнения лабораторной работы я освоила процедуру компиляции и сборки программ, написанных на ассемблере NASM.

## **5.Список литературы:**

Курс: Архитектура компьютеров и операционные системы.  
Раздел "Архитектура компьютеров" (02.03.00, УГСН) (rudn.ru)