

# **Отчет по лабораторной работе № 4**

**Дисциплина: архитектура компьютеров**

Казазаев Даниил Михайлович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задания Лабораторной работы</b>	<b>5</b>
<b>3</b>	<b>Задания Самостоятельной работы</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
4.1	Выполнение заданий самостоятельной работы . . . . .	9
<b>5</b>	<b>Вывод</b>	<b>12</b>

## Список иллюстраций

4.1	Переход в нужную директорию . . . . .	7
4.2	Создание файла hello.asm и проверка того, что он создан . . . . .	7
4.3	Введенный текст . . . . .	7
4.4	Установка NASM . . . . .	8
4.5	Перенос в объектный файл . . . . .	8
4.6	Компиляция файла, создание листинг файла и проверка . . . . .	8
4.7	Передача компоновщику и проверка созданного файлаисполнителя	8
4.8	Запущенный исполнительный файл . . . . .	8
4.9	Скопированный файл lab04.asm . . . . .	9
4.10	Отредактированный файл . . . . .	10
4.11	Перенос в объектный файл . . . . .	10
4.12	Компиляция файла, создание листинг файла и проверка . . . . .	10
4.13	Передача компоновщику и проверка созданного файлаисполнителя	10
4.14	Запущенный исполнительный файл . . . . .	11

# 1 Цель работы

Освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM

## 2 Задания Лабораторной работы

1. Создать текстовый файл с именем `hello.asm`
2. Ввести текст программы
3. Перевести текст программы в объектный код с помощью NASM
4. Скомпилировать файл `hello.asm` в `obj.o`
5. Передать объектный файл на обработку компоновщику
6. Запустить файл

### 3 Задания Самостоятельной работы

1. В каталоге `~/work/arch-рс/lab04` с помощью команды `ср` создайте копию файла `hello.asm` с именем `lab4.asm`
2. С помощью любого текстового редактора внесите изменения в текст программы в файле `lab4.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с вашими фамилией и именем.
3. Оттранслируйте полученный текст программы `lab4.asm` в объектный файл. Выполните компоновку объектного файла и запустите получившийся исполняемый файл.
4. Скопируйте файлы `hello.asm` и `lab4.asm` в Ваш локальный репозиторий в каталог `~/work/study/2023-2024/“Архитектура компьютера”/arch-рс/labs/lab04/`. Загрузите файлы на Github.

## 4 Выполнение лабораторной работы

Перехожу в нужную директорию для начала выполнения лабораторной работы.  
(рис. [4.1])

```
dmkazazaev@Ubuntu:~$ cd ~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/labs/lab04/  
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/labs/lab04$
```

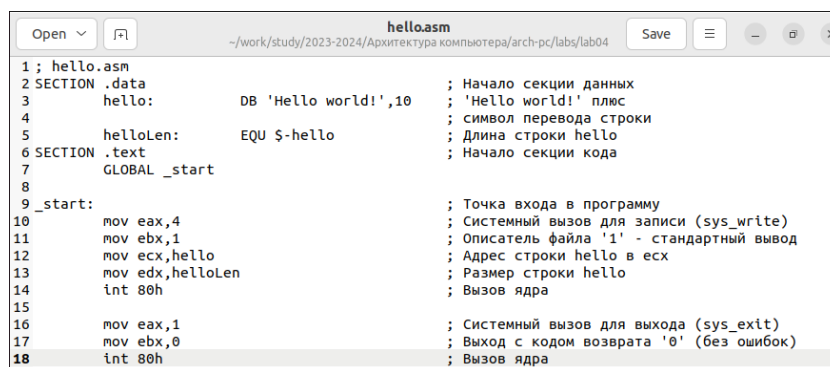
Рис. 4.1: Переход в нужную директорию

Создаю файл hello.asm. (рис. [4.2])

```
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/labs/lab04$ touch hello.asm  
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура\ компьютера/arch-pc/labs/lab04$ ls  
hello.asm presentation report
```

Рис. 4.2: Создание файла hello.asm и проверка того, что он создан

Открываю файл и ввожу в него текст программы. (рис. [4.3])



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

Рис. 4.3: Введенный текст

Устанавливаю транслятор NASM. (рис. [4.4])

```
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ sudo
apt install nasm
[sudo] password for dmkazazaev:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  nasm
0 upgraded, 1 newly installed, 0 to remove and 10 not upgraded.
Need to get 375 kB of archives.
After this operation, 3 345 kB of additional disk space will be used.
Get:1 http://ru.archive.ubuntu.com/ubuntu jammy/universe amd64 nasm amd64 2.15.05-1 [375
kB]
Fetched 375 kB in 1s (671 kB/s)
Selecting previously unselected package nasm.
(Reading database ... 425311 files and directories currently installed.)
Preparing to unpack .../nasm_2.15.05-1_amd64.deb ...
Unpacking nasm (2.15.05-1) ...
Setting up nasm (2.15.05-1) ...
Processing triggers for man-db (2.10.2-1) ...
```

Рис. 4.4: Установка NASM

Переношу файл hello.asm в объектный файл hello.o. (рис. [4.5])

```
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ nasm
-f elf hello.asm
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm hello.o presentation report
```

Рис. 4.5: Перенос в объектный файл

Компилирую файл и создаю файл листинга list.lst. (рис. [4.6])

```
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ nasm
-o obj.o -f elf -g -l list.lst hello.asm
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello.asm hello.o list.lst obj.o presentation report
```

Рис. 4.6: Компиляция файла, создание листинг файла и проверка

Передаю файл hello.o на обработку компоновщику. (рис. [4.7])

```
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ld -
m elf_i386 hello.o -o hello
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello hello.asm hello.o list.lst obj.o presentation report
```

Рис. 4.7: Передача компоновщику и проверка созданного файлаисполнителя

Запускаю исполняемый файл. (рис. [4.8])

```
dmkazazaev@Ubuntu:~/work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ./hello
Hello world!
```

Рис. 4.8: Запущенный исполнительный файл

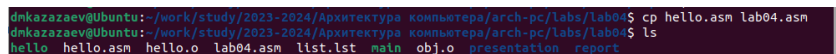


Листинг программы hello:

```
1                                     ; hello.asm
2                                SECTION .data
3 00000000 48656C6C6F20776F72-      hello:      DB 'Hello world!',10
3 00000009 6C64210A
4
5                                helloLen:  EQU $-hello
6                                SECTION .text
7                                GLOBAL _start
8
9                                _start:
10 00000000 B804000000              mov eax,4
11 00000005 BB01000000              mov ebx,1
12 0000000A B9[00000000]            mov ecx,hello
13 0000000F BA0D000000              mov edx,helloLen
14 00000014 CD80                    int 80h
15
16 00000016 B801000000              mov eax,1
17 0000001B BB00000000              mov ebx,0
18 00000020 CD80                    int 80h
```

## 4.1 Выполнение заданий самостоятельной работы

Копирую файл hello.asm с названием lab04. (рис. [4.9])



```
dnkazaev@Ubuntu: /work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ cp hello.asm lab04.asm
dnkazaev@Ubuntu: /work/study/2023-2024/Архитектура компьютера/arch-pc/labs/lab04$ ls
hello  hello.asm  hello.o  lab04.asm  list.lst  main  obj.o  presentation  report
```

Рис. 4.9: Скопированный файл lab04.asm

Редактирую файл. (рис. [4.10])

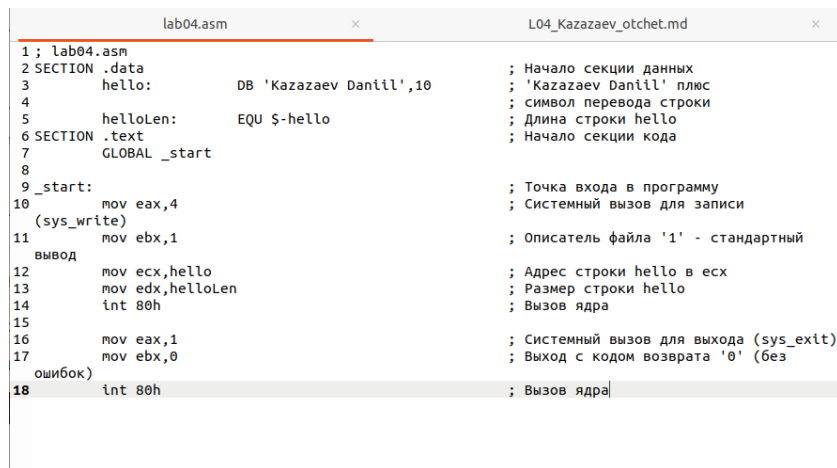


Рис. 4.10: Отредактированный файл

Переношу файл lab04.asm в объектный файл lab04.o. (рис. [4.11])

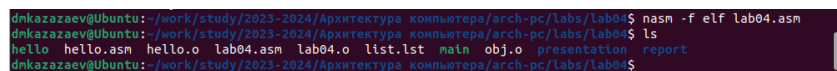


Рис. 4.11: Перенос в объектный файл

Компилирую файл и создаю файл листинга list\_lab04.lst. (рис. [4.12])

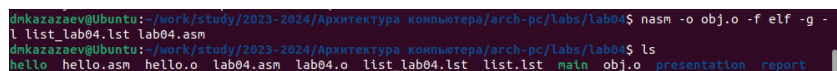


Рис. 4.12: Компиляция файла, создание листинг файла и проверка

Передаю файл lab04.o на обработку компоновщику. (рис. [4.13])

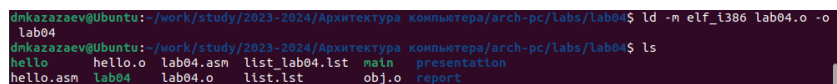


Рис. 4.13: Передача компоновщику и проверка созданного файлаисполнителя

Запускаю исполняемый файл. (рис. [4.14])

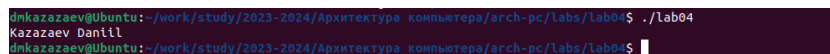


Рис. 4.14: Запущенный исполнительный файл

Листинг программы lab04:

```
1                                ; lab04.asm
2                                SECTION .data
3 00000000 4B617A617A61657620-    hello:      DB 'Kazazaev Daniil',10
3 00000009 44616E69696C0A
4
5                                helloLen:  EQU $-hello
6                                SECTION .text
7                                GLOBAL _start
8
9                                _start:
10 00000000 B804000000            mov eax,4
11 00000005 BB01000000            mov ebx,1
12 0000000A B9[00000000]          mov ecx,hello
13 0000000F BA10000000            mov edx,helloLen
14 00000014 CD80                  int 80h
15
16 00000016 B801000000            mov eax,1
17 0000001B BB00000000            mov ebx,0
18 00000020 CD80                  int 80h
```

## 5 Вывод

Итогом данной работы стала первая программа, которую я написал на языке ассемблера NASM