

# **Отчёт по лабораторной работе 4**

## **Архитектура компьютеров**

Алхоев Абдулмалик-Салим Гапурович

# Содержание

<b>1 Цель работы</b>	<b>5</b>
<b>2 Выполнение лабораторной работы</b>	<b>6</b>
2.1 Программа Hello world! . . . . .	6
2.2 Транслятор NASM . . . . .	7
2.3 Компоновщик LD . . . . .	8
2.4 Выполнение заданий для самостоятельной работы. . . . .	9
<b>3 Выводы</b>	<b>11</b>

# **Список иллюстраций**

2.1 Создание каталога и файла . . . . .	6
2.2 Программа hello.asm . . . . .	7
2.3 Трансляция hello.asm . . . . .	7
2.4 Трансляция hello.asm с дополнительными опциями . . . . .	8
2.5 Линковка программы . . . . .	8
2.6 Линковка программы . . . . .	8
2.7 Запуск программ . . . . .	9
2.8 Код программы в файле lab4.asm . . . . .	10
2.9 Запуск программы lab4.asm . . . . .	10

# **Список таблиц**

# **1 Цель работы**

Целью работы является освоение процедуры компиляции и сборки программ, написанных на ассемблере NASM.

## 2 Выполнение лабораторной работы

### 2.1 Программа Hello world!

Создаю каталог lab04 командой mkdir, перехожу в него с помощью команды cd, создаю файл hello.asm. (рис. 2.1)

```
alhoev@ubuntu-vm:~$  
alhoev@ubuntu-vm:~$ mkdir -p ~/work/arch-pc/lab04  
alhoev@ubuntu-vm:~$ cd ~/work/arch-pc/lab04  
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ touch hello.asm  
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ █
```

Рисунок 2.1: Создание каталога и файла

Открыл файл и написал код программы по заданию.(рис. 2.2)

The screenshot shows a text editor window titled "hello.asm" with the file path "~/work/arch-pc/lab04". The code is written in NASM assembly language:

```
1 SECTION .data
2     hello:      db "Hello, world!",0xa
3     helloLen:   equ $ - hello
4 SECTION .text
5     global _start
6
7 _start:
8     mov eax, 4
9     mov ebx, 1
10    mov ecx, hello
11    mov edx, helloLen
12    int 0x80
13
14    mov eax, 1
15    mov ebx, 0
16    int 0x80
17
```

Рисунок 2.2: Программа hello.asm

## 2.2 Транслятор NASM

Транслирую файл командой nasm. Получился объектный файл hello.o (рис. 2.3)

```
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ nasm -f elf hello.asm
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
alhoev@ubuntu-vm:~/work/arch-pc/lab04$
```

Рисунок 2.3: Трансляция hello.asm

Транслирую файл командой nasm с дополнительными опциями. (рис. 2.4)

Получился файл листинга list.lst, объектный файл obj.o, в программу добавилась отладочная информация.

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

Рисунок 2.4: Трансляция hello.asm с дополнительными опциями

## 2.3 Компоновщик LD

Выполняю линковку командой ld и получил исполняемый файл. (рис. 2.5)

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

Рисунок 2.5: Линковка программы

Еще раз выполняю линковку для объектного файла obj.o и получаю исполняемый файл main.(рис. 2.6)

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

Рисунок 2.6: Линковка программы

Запускаю исполняемые файлы.(рис. 2.7)

```
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ ./hello
Hello, world!
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ ./main
Hello, world!
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ █
```

Рисунок 2.7: Запуск программ

## 2.4 Выполнение заданий для самостоятельной работы.

Копирую программу в новый файл.

Изменяю сообщение Hello world на свое имя (рис. 2.8) и запускаю новую программу. (рис. 2.9)

The screenshot shows a text editor window with the file 'lab4.asm' open. The code is written in NASM assembly syntax. It defines a string 'hello' with the value 'Malik', calculates its length 'helloLen', and then prints it using the BIOS interrupt 0x80. The program then exits with status code 1. The code is numbered from 1 to 17.

```
1 SECTION .data
2     hello:      db "Malik",0xa
3     helloLen:    equ $ - hello
4 SECTION .text
5     global _start
6
7 _start:
8     mov eax, 4
9     mov ebx, 1
10    mov ecx, hello
11    mov edx, helloLen
12    int 0x80
13
14    mov eax, 1
15    mov ebx, 0
16    int 0x80
17
```

Рисунок 2.8: Код программы в файле lab4.asm

The terminal session shows the compilation of 'lab4.asm' to an ELF executable 'lab4'. The assembly code is assembled using 'nasm -f elf lab4.asm', and then linked with 'ld -m elf\_i386 lab4.o -o lab4'. Finally, the program is run with './lab4', outputting the string 'Malik'.

```
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
alhoev@ubuntu-vm:~/work/arch-pc/lab04$ ./lab4
Malik
alhoev@ubuntu-vm:~/work/arch-pc/lab04$
```

Рисунок 2.9: Запуск программы lab4.asm

## **3 Выводы**

При выполнении данной лабораторной работы я освоил процесс компиляции и сборки программ, написанных на ассемблере nasm.