

Отчёт по лабораторной работе 4

дисциплина: Архитектура компьютера

Фархад Ахамд Камран

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Программа Hello world!	6
2.2	Трансляция кода с помощью NASM	7
2.3	Линковка с использованием LD	8
2.4	Выполнение заданий для самостоятельной работы	9
2.5	Выводы	10

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	9
2.9	Запуск программы lab4.asm	10

Список таблиц

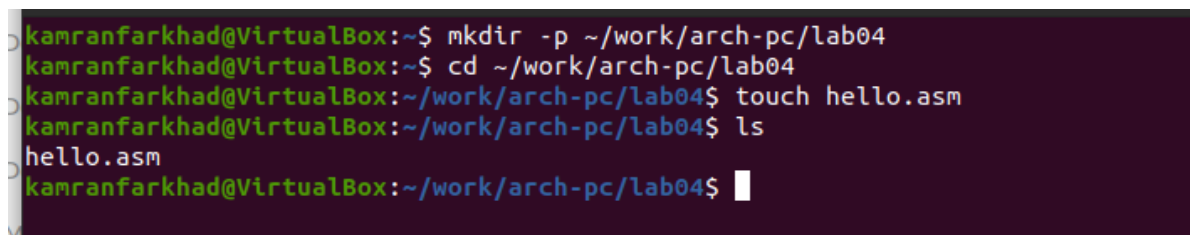
1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Программа Hello world!

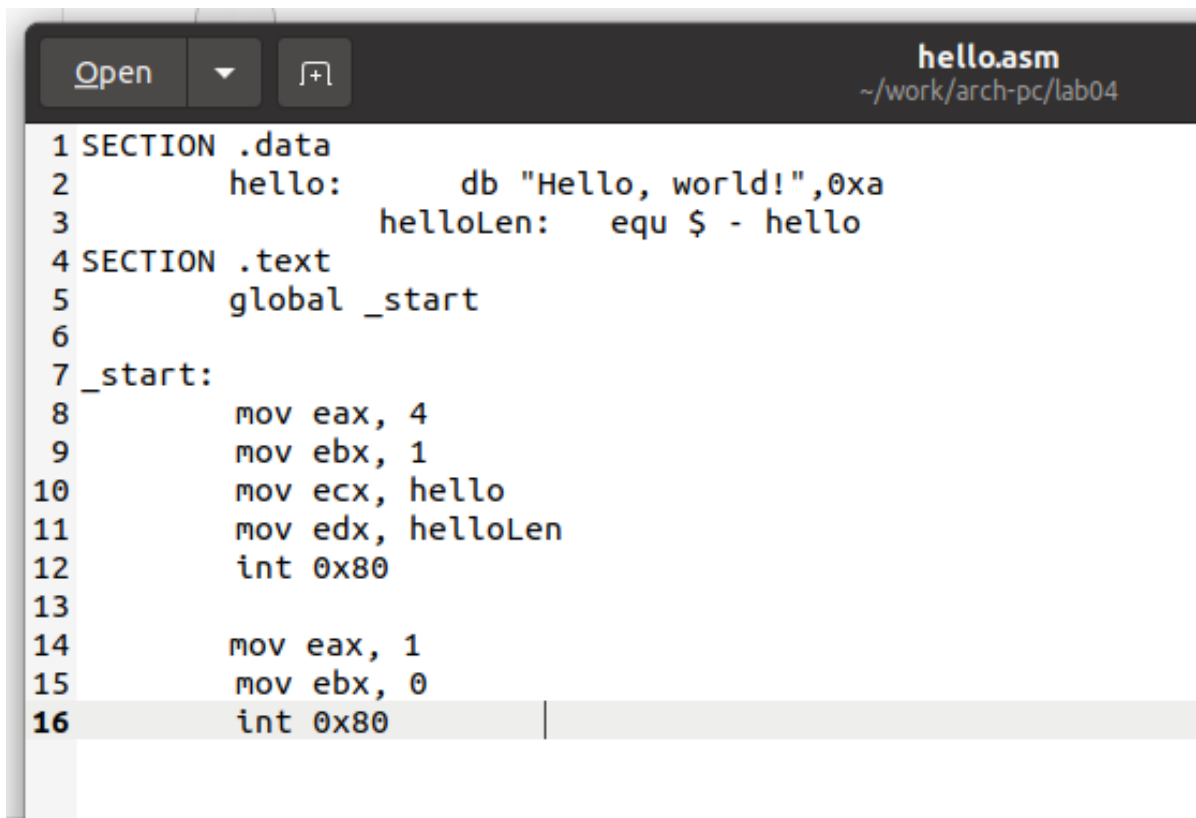
Для начала создаю новый каталог lab04 с помощью команды mkdir, затем перехожу в него, используя команду cd. После этого создаю файл hello.asm. На рис. 2.1 показан процесс создания каталога и файла.



```
kamranfarkhad@VirtualBox:~$ mkdir -p ~/work/arch-pc/lab04
kamranfarkhad@VirtualBox:~$ cd ~/work/arch-pc/lab04
kamranfarkhad@VirtualBox:~/work/arch-pc/lab04$ touch hello.asm
kamranfarkhad@VirtualBox:~/work/arch-pc/lab04$ ls
hello.asm
kamranfarkhad@VirtualBox:~/work/arch-pc/lab04$
```

Рис. 2.1: Создание каталога и файла

Открываю файл hello.asm в текстовом редакторе и пишу код программы по заданию, как показано на рис. 2.2.

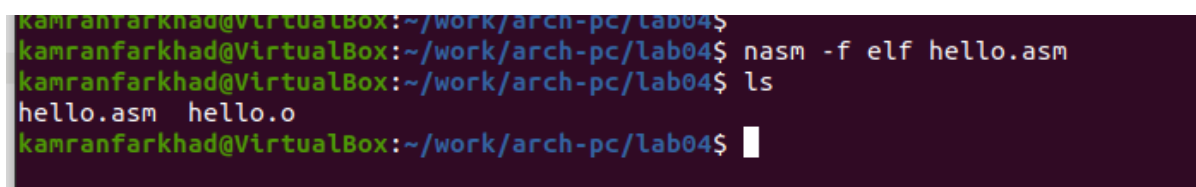


```
1 SECTION .data
2     hello:      db "Hello, world!",0xa
3     helloLen:   equ $ - hello
4 SECTION .text
5     global _start
6
7 _start:
8     mov eax, 4
9     mov ebx, 1
10    mov ecx, hello
11    mov edx, helloLen
12    int 0x80
13
14    mov eax, 1
15    mov ebx, 0
16    int 0x80
```

Рис. 2.2: Программа hello.asm

2.2 Трансляция кода с помощью NASM

Для того чтобы скомпилировать файл, использую транслятор NASM. С помощью команды `nasm` я создаю объектный файл `hello.o`, что показано на рис. 2.3.



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

Рис. 2.3: Трансляция hello.asm

Для более детального анализа программы применяю команду `nasm` с дополнительными опциями, которые позволяют создать листинг (`list.lst`),

объектный файл (obj.o), а также добавить отладочную информацию. Результат показан на рис. 2.4.

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

Рис. 2.4: Трансляция hello.asm с дополнительными опциями

2.3 Линковка с использованием LD

После успешной трансляции выполняю линковку объектного файла hello.o, используя компоновщик ld. Это позволяет создать исполняемый файл, как показано на рис. 2.5.

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

Рис. 2.5: Линковка программы

Затем повторяю этот процесс для объектного файла obj.o, в результате чего получаю исполняемый файл с именем main. Результат показан на рис. 2.6.

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

Рис. 2.6: Линковка программы

Запускаю оба полученных исполняемых файла, как видно на рис. 2.7.

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

Рис. 2.7: Запуск программ

2.4 Выполнение заданий для самостоятельной работы

Для выполнения индивидуального задания копирую исходную программу в новый файл. Вношу изменения в код, заменяя сообщение “Hello world” на своё имя, что продемонстрировано на рис. 2.8. После этого запускаю изменённую программу (рис. 2.9).



```
lab4.asm  
~/work/arch-pc/lab04  
1 SECTION .data  
2     hello:      db "Kamran Farkhad",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
```

Рис. 2.8: Код программы в файле lab4.asm

```
kamranfarkhad@VirtualBox:~/work/arch-pc/lab04$  
kamranfarkhad@VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm  
kamranfarkhad@VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4  
kamranfarkhad@VirtualBox:~/work/arch-pc/lab04$ ./lab4  
Kamran Farkhad  
kamranfarkhad@VirtualBox:~/work/arch-pc/lab04$
```

Рис. 2.9: Запуск программы lab4.asm

2.5 Выводы

В ходе выполнения лабораторной работы ознакомились с основными этапами работы с программами на ассемблере с использованием NASM.