

Отчёт по лабораторной работе 4

Архитектура компьютера

Сувд Адисурэн

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Программа Hello world!	6
2.2	Транслятор NASM	7
2.3	Компоновщик LD	8
2.4	Запуск исполняемого файла	8
2.5	Задание для самостоятельной работы	9
3	Выводы	11

Список иллюстраций

2.1	Создан каталог для работы и файл для программы	6
2.2	Программа в файле hello.asm	7
2.3	Трансляция программы	8
2.4	Компоновка программы	8
2.5	Запуск программы	9
2.6	Программа в файле lab4.asm	9
2.7	Проверка 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!

Я создала каталог `lab04` с помощью команды `mkdir`, затем перешла в него, используя команду `cd`, и создала файл `hello.asm`, в который будет написана программа. Чтобы убедиться, что файл был успешно создан, я выполнила команду `ls`.

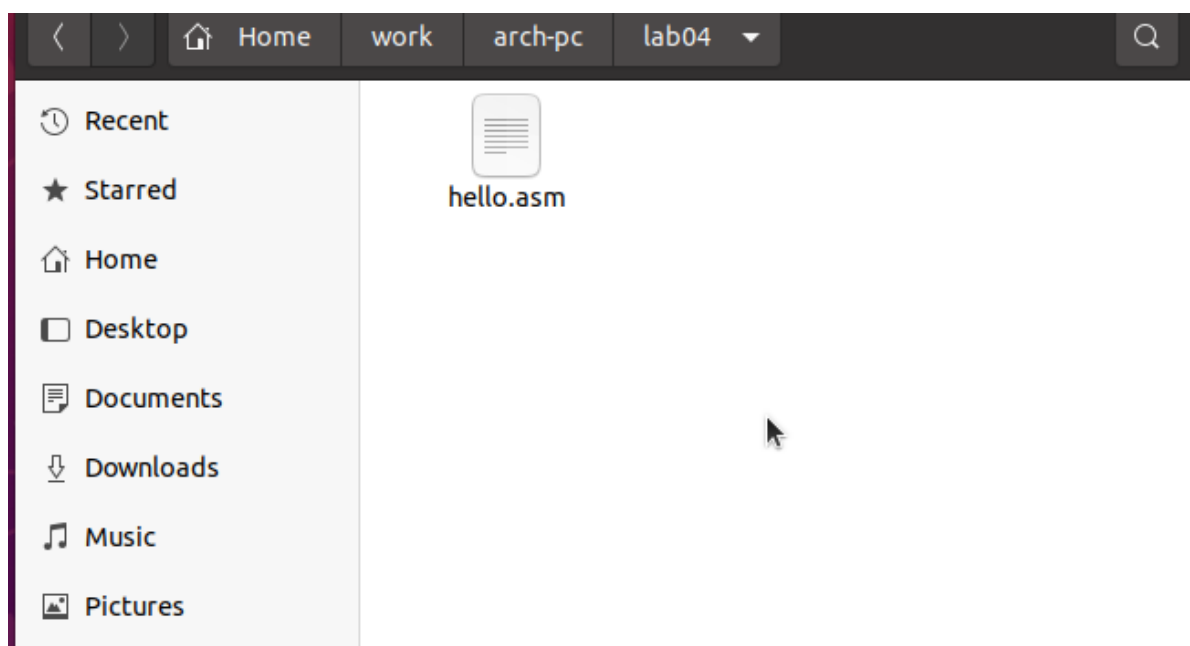
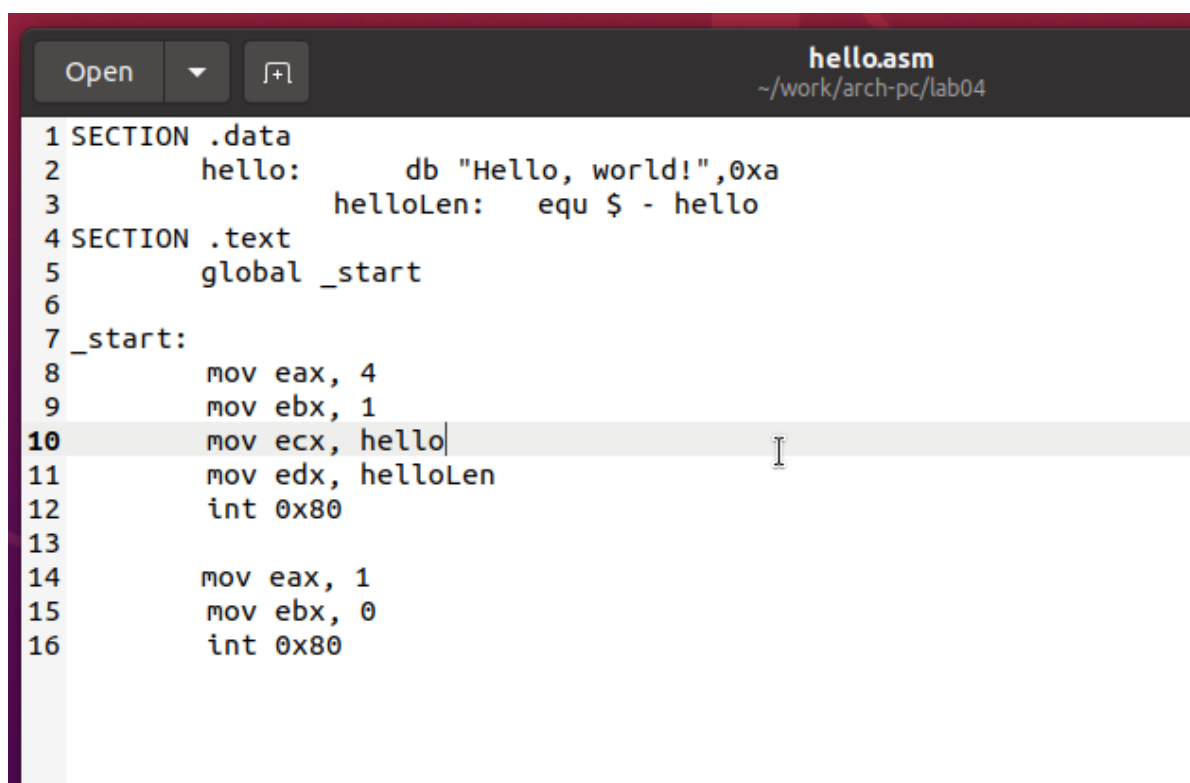


Рис. 2.1: Создан каталог для работы и файл для программы

После этого я написала программу на языке ассемблера по заданию.



```
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 в файле hello.asm

2.2 Транслятор NASM

Транслятор **NASM** превращает текст программы в объектный код. Если в тексте программы нет ошибок, транслятор преобразует её из файла `hello.asm` в объектный файл `hello.o`.

Я скомпилировала файл командой `nasm` и получила объектный файл `hello.o`.

Затем я повторно скомпилировала файл `hello.asm`, но уже с дополнительными опциями:

- с опцией `-l` был получен файл листинга `list.lst`;
- с опцией `-f` получен объектный файл `obj.o`;
- с опцией `-g` в программу добавлена отладочная информация.

```

suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$ nasm -f elf hello.asm
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$ ls
hello.asm hello.o
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst h
ello.asm
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$ ls
hello.asm hello.o list.lst obj.o
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$

```

Рис. 2.3: Трансляция программы

2.3 Компоновщик LD

Чтобы получить исполняемую программу, необходимо передать объектный файл на обработку компоновщику **LD**.

Я выполнила команду `ld` и скомпоновала объектный файл `hello.o`, получив исполняемый файл `hello`.

После этого я повторила команду `ld`, но уже для файла `obj.o`, что позволило мне получить исполняемый файл `main`.

```

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

```

Рис. 2.4: Компоновка программы

2.4 Запуск исполняемого файла

Запустила оба исполняемых файла, чтобы проверить их работу.


```

suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$ ./hello
Hello, world!
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$ ./main
Hello, world!
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$

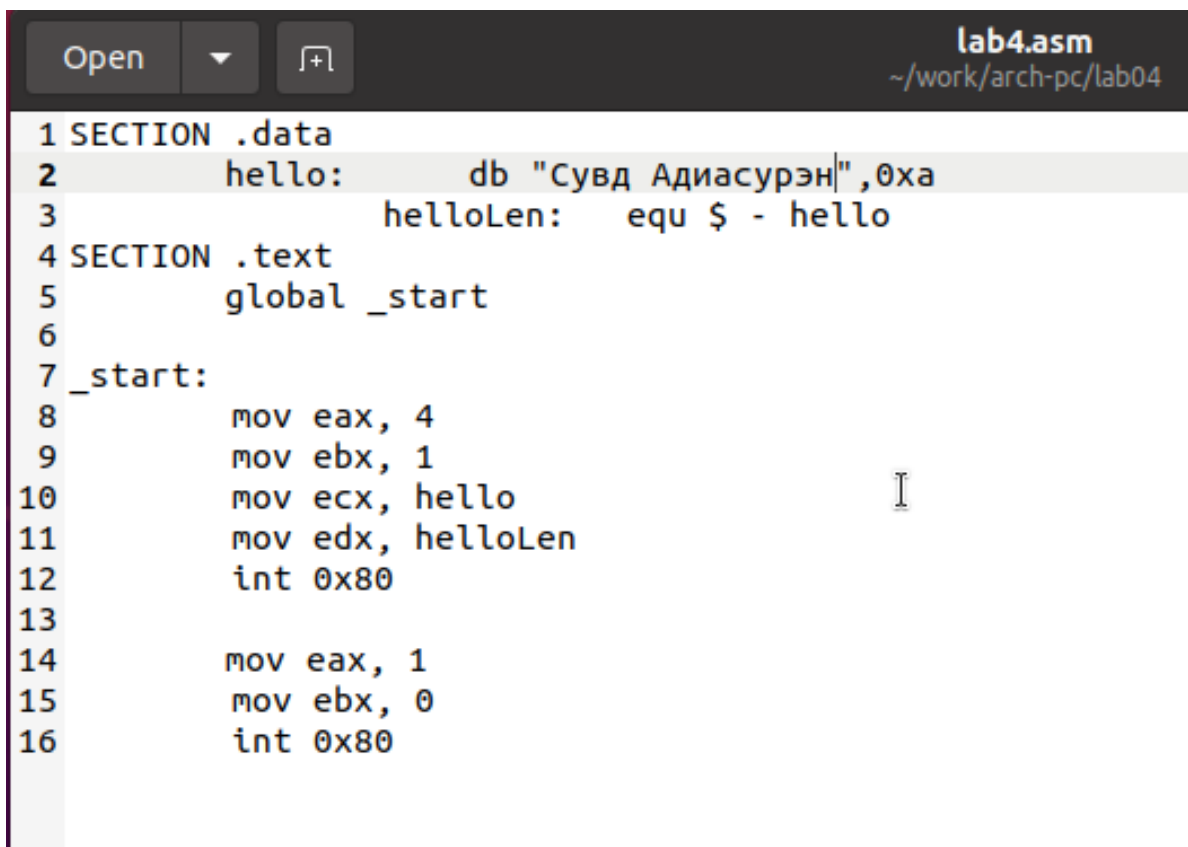
```

Рис. 2.5: Запуск программы

2.5 Задание для самостоятельной работы

Для выполнения самостоятельного задания я скопировала файл `hello.asm` в новый файл `lab4.asm` командой `cp`.

В файле `lab4.asm` я изменила текст сообщения с “Hello world” на своё имя.



```

Open  ▾  [+l  lab4.asm
~/work/arch-pc/lab04
1 SECTION .data
2     hello:      db "Сувд Адиясурэн",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. 2.6: Программа в файле `lab4.asm`

Затем запустила изменённую программу и проверила результат.

```
suvdadiasuren@ubuntu:~/work/arch-pc/lab04$  
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$ nasm -f elf lab4.asm  
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4  
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$ ./lab4  
Сувд Адасурэн  
suvdadiasuren@Ubuntu:~/work/arch-pc/lab04$
```

Рис. 2.7: Проверка программы lab4.asm

3 Выводы

Таким образом, я успешно выполнила лабораторную работу, используя **NASM** и **LD** для трансляции, компоновки и запуска программ на языке ассемблера.