

Отчёт по лабораторной работе 4

Архитектура компьютера

Малютина Софья Александровна

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	10
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

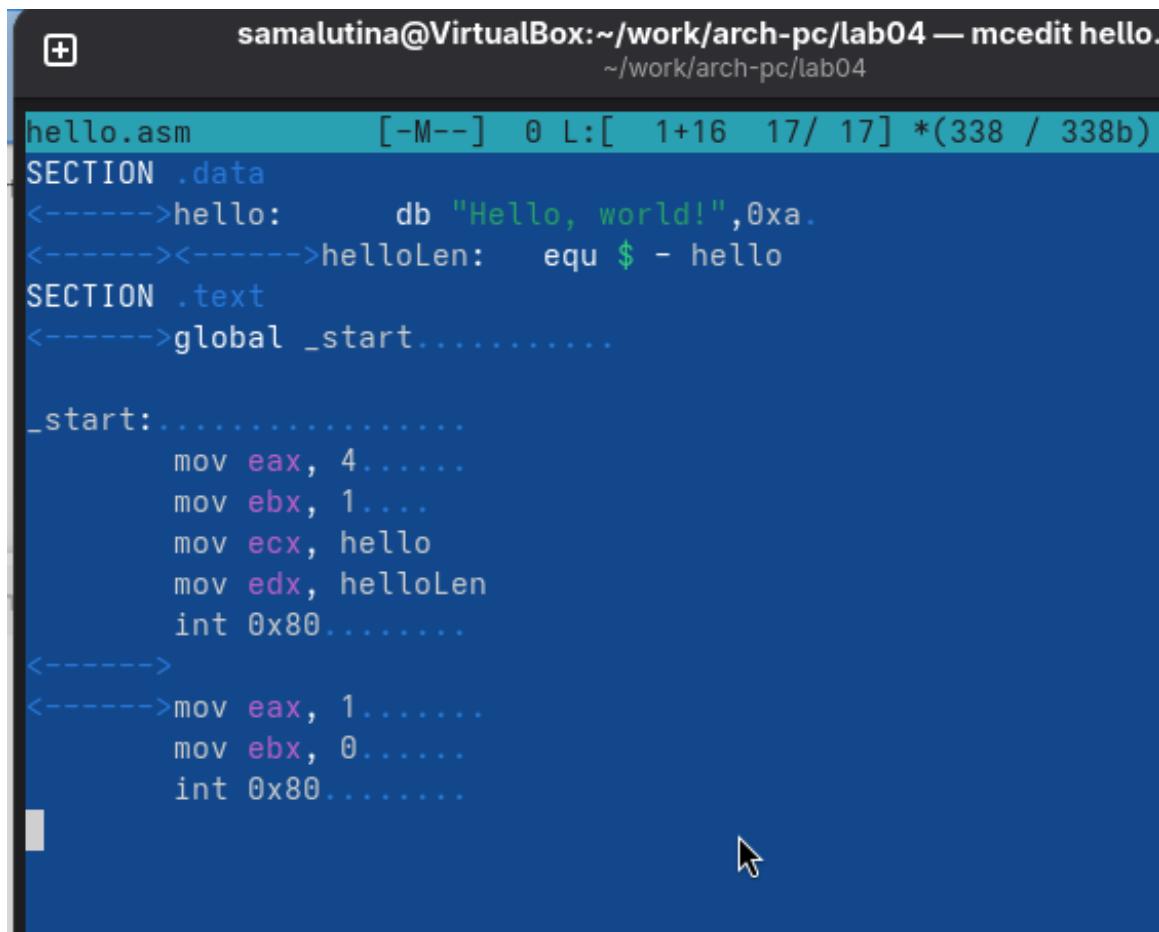
2.1 Программа Hello world!

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`.

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

Рисунок 2.1: Создан каталог для работы и файл для программы

После этого я написала программу на языке ассемблера по заданию.



```
samalutina@VirtualBox:~/work/arch-pc/lab04 — mcedit hello.  
~/work/arch-pc/lab04  
  
hello.asm [-M--] 0 L:[ 1+16 17/ 17] *(338 / 338b)  
SECTION .data  
<----->hello:      db "Hello, world!",0xa.  
<-----><----->helloLen:    equ $ - hello  
SECTION .text  
<----->global _start.....  
  
_start:.....  
        mov eax, 4.....  
        mov ebx, 1....  
        mov ecx, hello  
        mov edx, helloLen  
        int 0x80.....  
<----->  
<----->mov eax, 1.....  
        mov ebx, 0.....  
        int 0x80.....
```

Рисунок 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.

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

Рисунок 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`.

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

Рисунок 2.4: Компоновка программы

2.4 Запуск исполняемого файла

Запустила оба исполняемых файла, чтобы проверить их работу.

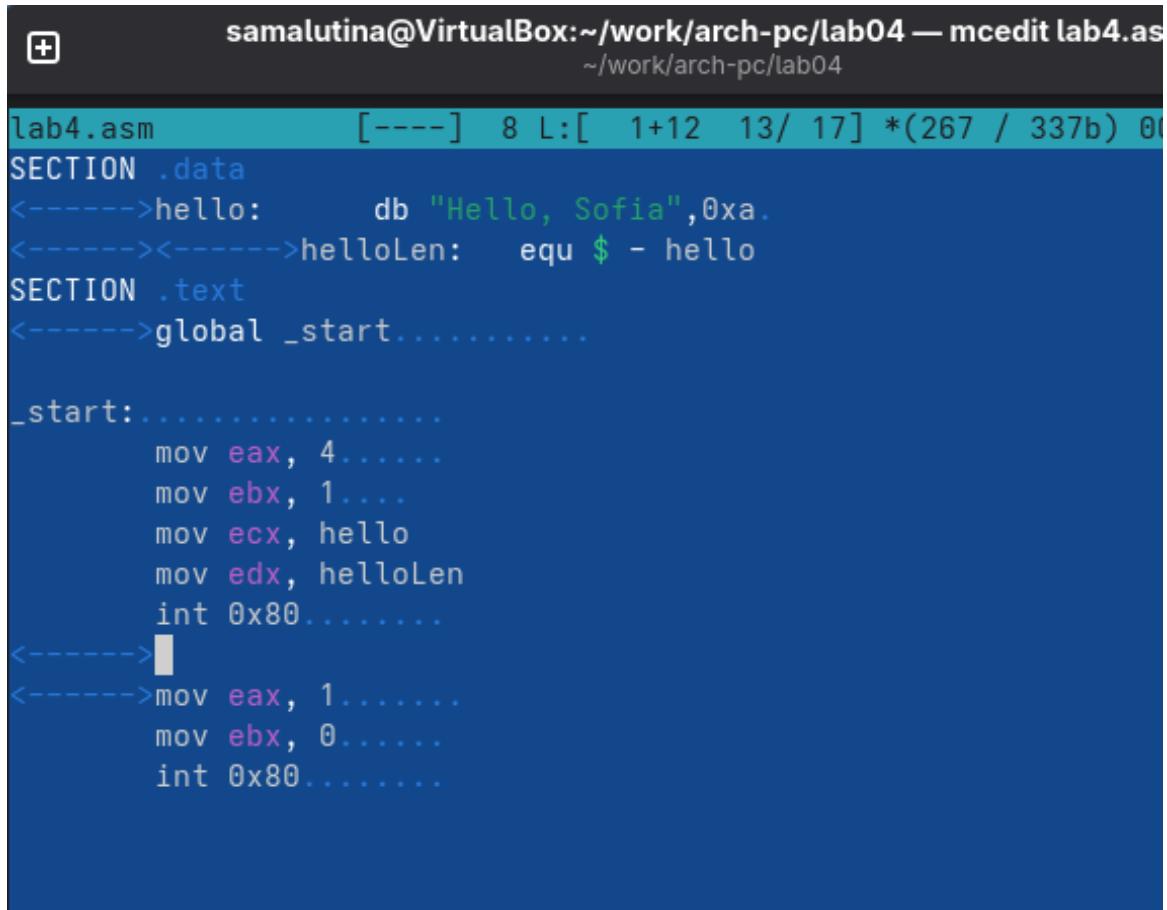
```
samatutina@VirtualBox:~/work/arch-pc/lab04$  
samatutina@VirtualBox:~/work/arch-pc/lab04$ ./hello  
Hello, world!  
samatutina@VirtualBox:~/work/arch-pc/lab04$ ./main  
Hello, world!  
samatutina@VirtualBox:~/work/arch-pc/lab04$ █
```

Рисунок 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`.

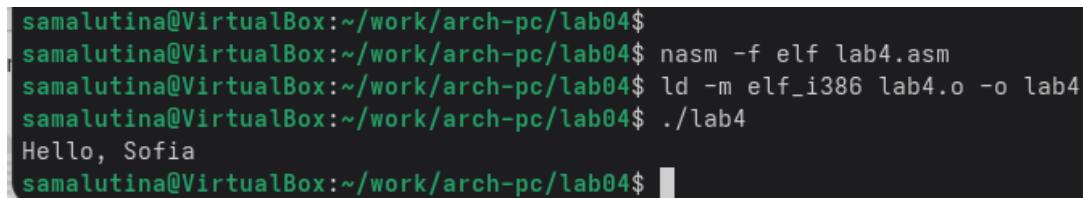
В файле `lab4.asm` я изменила текст сообщения с «Hello world» на своё имя.



```
samalutina@VirtualBox:~/work/arch-pc/lab04 — mcedit lab4.asm
samalutina@VirtualBox:~/work/arch-pc/lab04 — mcedit lab4.asm
lab4.asm      [----]  8 L:[ 1+12 13/ 17] *(267 / 337b) 00
SECTION .data
<----->hello:    db "Hello, Sofia",0xa.
<-----><----->helloLen:   equ $ - hello
SECTION .text
<----->global _start.....  
  
_start:.....  
    mov eax, 4.....  
    mov ebx, 1.....  
    mov ecx, hello  
    mov edx, helloLen  
    int 0x80.....  
<----->█  
<----->mov eax, 1.....  
    mov ebx, 0.....  
    int 0x80.....
```

Рисунок 2.6: Программа в файле lab4.asm

Затем запустила изменённую программу и проверила результат.



```
samalutina@VirtualBox:~/work/arch-pc/lab04$  
samalutina@VirtualBox:~/work/arch-pc/lab04$ nasm -f elf lab4.asm  
samalutina@VirtualBox:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4  
samalutina@VirtualBox:~/work/arch-pc/lab04$ ./lab4  
Hello, Sofia  
samalutina@VirtualBox:~/work/arch-pc/lab04$ █
```

Рисунок 2.7: Проверка программы lab4.asm

3 Выводы

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