

Отчёт по лабораторной работе 4

Архитектура компьютеров и операционных систем

Игнатова Анастасия НБИбд-01-23

Содержание

1	Цель работы	5
2	Задания	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	13

Список иллюстраций

4.1	Файл для программы	9
4.2	Программа hello.asm	10
4.3	Компиляция и запуск программы	11
4.4	Программа в файле lab4.asm	11
4.5	Сборка и проверка программы lab4.asm	12

Список таблиц

1 Цель работы

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

2 Задания

1. Изучить основы языка Ассемблера
2. Изучить и рассмотреть на практике процесс сборки программы
3. Выполнить задание по программе
4. Подготовить отчет и загрузить на GitHub

3 Теоретическое введение

Язык ассемблера (assembly language, сокращённо asm) — машинно-ориентированный язык низкого уровня. Можно считать, что он больше любых других языков приближен к архитектуре ЭВМ и её аппаратным возможностям, что позволяет получить к ним более полный доступ, нежели в языках высокого уровня, таких как C/C++, Perl, Python и пр. Заметим, что получить полный доступ к ресурсам компьютера в современных архитектурах нельзя, самым низким уровнем работы прикладной программы является обращение напрямую к ядру операционной системы. Именно на этом уровне и работают программы, написанные на ассемблере. Но в отличие от языков высокого уровня ассемблерная программа содержит только тот код, который ввёл программист. Таким образом язык ассемблера — это язык, с помощью которого понятным для человека образом пишутся команды для процессора.

Следует отметить, что процессор понимает не команды ассемблера, а последовательности из нулей и единиц — машинные коды. До появления языков ассемблера программистам приходилось писать программы, используя только лишь машинные коды, которые были крайне сложны для запоминания, так как представляли собой числа, записанные в двоичной или шестнадцатеричной системе счисления. Преобразование или трансляция команд с языка ассемблера в исполняемый машинный код осуществляется специальной программой транслятором — Ассемблер

Программы, написанные на языке ассемблера, не уступают в качестве и скорости программам, написанным на машинном языке, так как транслятор

просто переводит мнемонические обозначения команд в последовательности бит (нулей и единиц).

В нашем курсе будет использоваться ассемблер NASM (Netwide Assembler). NASM — это открытый проект ассемблера, версии которого доступны под различные операционные системы и который позволяет получать объектные файлы для этих систем. В NASM используется Intel-синтаксис и поддерживаются инструкции x86-64. Типичный формат записи команд NASM имеет вид:

[метка:] мнемокод [операнд {, операнд}] [; комментарий]

Здесь мнемокод — непосредственно мнемоника инструкции процессору, которая является обязательной частью команды. Операндами могут быть числа, данные, адреса регистров или адреса оперативной памяти. Метка — это идентификатор, с которым ассемблер ассоциирует некоторое число, чаще всего адрес в памяти. Т.о. метка перед командой связана с адресом данной команды.

4 Выполнение лабораторной работы

Создаю каталог lab04, перехожу в него, создаю файл hello.asm.

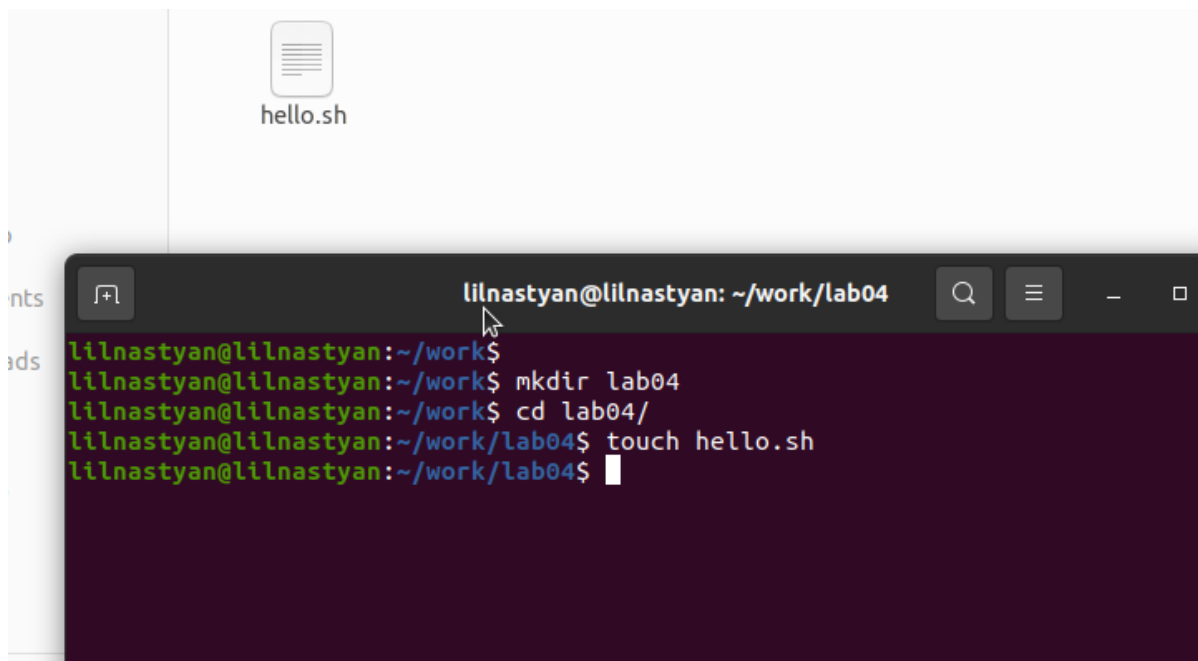


Рис. 4.1: Файл для программы

Написала код программы 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. 4.2: Программа hello.asm

Транслировала файл командой `nasm`. Получился объектный файл `hello.o`

Транслировала файл командой `nasm` с дополнительными опциями.Получился файл листинга и объектный файл.

Выполнила компоновку и получила исполняемый файл.

Еще раз выполнила компоновку для второго объектного файла и получила новый исполняемый файл.

Запустила исполняемые файлы.

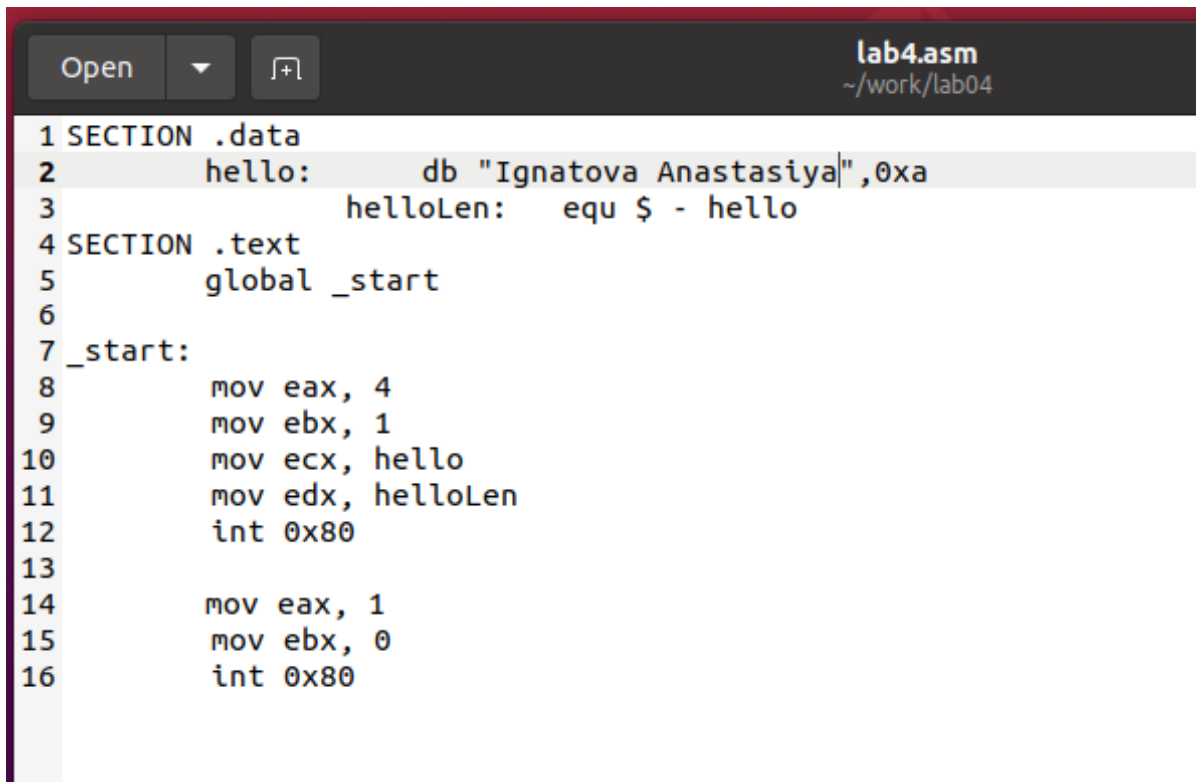
```

lilnastyan@lilnastyan:~/work/lab04$ nasm -f elf hello.asm
lilnastyan@lilnastyan:~/work/lab04$
lilnastyan@lilnastyan:~/work/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
lilnastyan@lilnastyan:~/work/lab04$ ls
hello.asm hello.o list.lst obj.o
lilnastyan@lilnastyan:~/work/lab04$ ld -m elf_i386 hello.o -o hello
lilnastyan@lilnastyan:~/work/lab04$ ld -m elf_i386 obj.o -o main
lilnastyan@lilnastyan:~/work/lab04$ ls
hello hello.asm hello.o list.lst main obj.o
lilnastyan@lilnastyan:~/work/lab04$ ./hello
Hello, world!
lilnastyan@lilnastyan:~/work/lab04$ ./main
Hello, world!
lilnastyan@lilnastyan:~/work/lab04$

```

Рис. 4.3: Компиляция и запуск программы

Изменила сообщение Hello world на свое имя и запустил файл еще раз.



```

1 SECTION .data
2     hello:      db "Ignatova Anastasiya",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. 4.4: Программа в файле lab4.asm

```
lilnastyan@lilnastyan:~/work/lab04$ cp hello.asm lab4.asm
lilnastyan@lilnastyan:~/work/lab04$ nasm -f elf lab4.asm
lilnastyan@lilnastyan:~/work/lab04$ ld -m elf_i386 lab4.o -o lab4
lilnastyan@lilnastyan:~/work/lab04$ ./lab4
Ignatova Anastasiya
lilnastyan@lilnastyan:~/work/lab04$ █
```

Рис. 4.5: Сборка и проверка программы lab4.asm

5 Выводы

Освоили процесс компиляции и сборки программ, написанных на ассемблере `nasm`.