

Отчёта по лабораторной работе 5

Создание и процесс обработки программ на языке ассемблера NASM

Оширова Юлия Николаевна НКАбд-02-22

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
5	Выводы	13
	Список литературы	14

Список иллюстраций

4.1	Файл hello.asm	9
4.2	Работа программы hello	10
4.3	Файл lab05.asm	11
4.4	Работа программы lab05	12

Список таблиц

1 Цель работы

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

2 Задание

1. Изучите программу HelloWorld и скомпилируйте ее.
2. С помощью любого текстового редактора внесите изменения в текст программы так, чтобы вместо Hello world! на экран выводилась строка с вашими фамилией и именем.
3. Скомпилируйте новую программу и проверьте ее работу.
4. Загрузите файлы на GitHub.

3 Теоретическое введение

NASM (англ. Netwide Assembler) – это 80x86 ассемблер, разработанный исходя из принципов переносимости и модульности. Он поддерживает широкий диапазон форматов объектных файлов, включая форматы Linux.out и ELF, NetBSD/FreeBSD, COFF, Microsoft 16-bit OBJ и Win32. Он способен также создавать простые бинарные файлы. Синтакс NASM максимально упрощен для понимания и похож на синтакс Intel, но слегка посложнее. Он поддерживает инструкции Pentium, P6 и MMX, а также имеет макро-расширения.

NASM был создан Саймоном Тэтхемом совместно с Юлианом Холлом и в настоящее время развивается небольшой командой разработчиков на SourceForge.net. Первоначально он был выпущен согласно его собственной лицензии, но позже эта лицензия была заменена на GNU LGPL после множества проблем, вызванных выбором лицензии. Начиная с версии 2.07 лицензия заменена на «упрощённую BSD» (BSD из 2 пунктов).

NASM может работать на платформах, отличных от x86, таких как SPARC и PowerPC, однако код он генерирует только для x86 и x86-64.

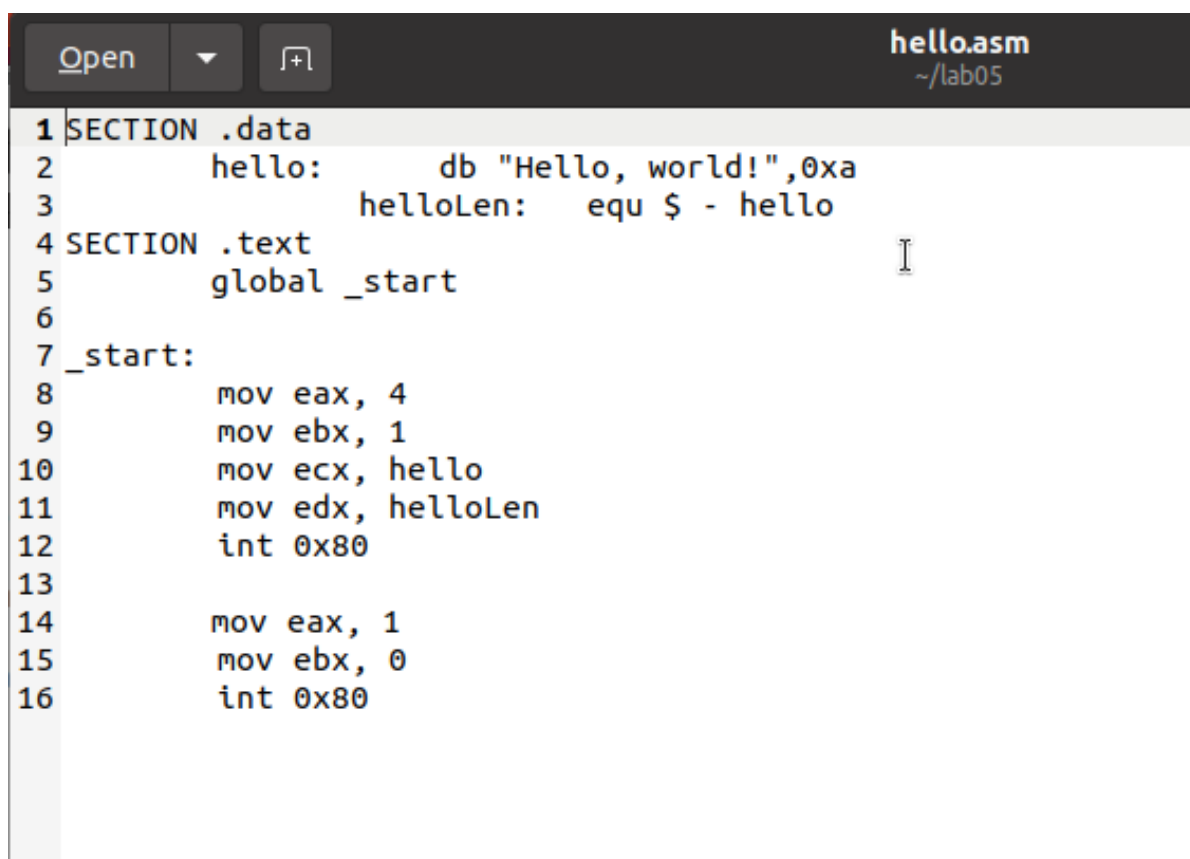
NASM успешно конкурирует со стандартным в Linux- и многих других UNIX-системах ассемблером gas. Считается, что качество документации у NASM выше, чем у gas. Кроме того, ассемблер gas по умолчанию использует AT&T-синтаксис, ориентированный на процессоры не от Intel, в то время как NASM использует вариант традиционного для x86-ассемблеров Intel-синтаксиса; Intel-синтаксис используется всеми ассемблерами для под DOS/Windows, например, MASM, TASM, fasm.

В NASM используется Intel-синтаксис записи инструкций. Предложение языка ассемблера NASM (строка программы) может состоять из следующих элементов: Метка, Инструкция, Операнды, Комментарий.

Операнды отделяются между собой запятой. Перед строкой и после инструкции можно использовать любое количество пробельных символов. Комментарий начинается с точки с запятой, а концом комментария считается конец строки. В качестве инструкции может использоваться команда или псевдокоманда (директива компилятора). Если строка очень длинная, то её можно перенести на следующую, используя обратный слеш (\), подобно тому, как это делается в языке Си.

4 Выполнение лабораторной работы

1. Создали каталог lab05 командой `mkdir`, перешли в него с помощью команды `cd`, скачали с ТУИС файл `hello.asm` и положили в папку. (рис. 4.1)
2. Открыли файл и изучили текст программы (рис. 4.1)

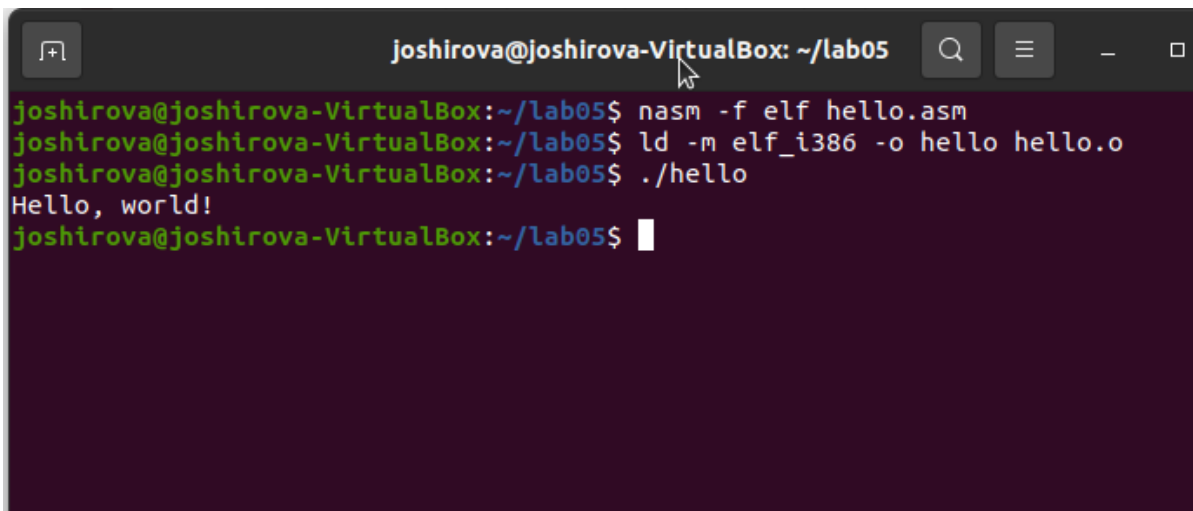


```
1 SECTION .data
2     hello:      db "Hello, world!",0xa
3     helloLen:   equ $ - hello
4 SECTION .text
5     global _start
6
7 _start:
8     mov eax, 4
9     mov ebx, 1
10    mov ecx, hello
11    mov edx, helloLen
12    int 0x80
13
14    mov eax, 1
15    mov ebx, 0
16    int 0x80
```

Рис. 4.1: Файл `hello.asm`

2. Транслировали файл командой `nasm`

3. Выполнили линковку командой `ld` и получили исполняемый файл и запустили его (рис. 4.2)



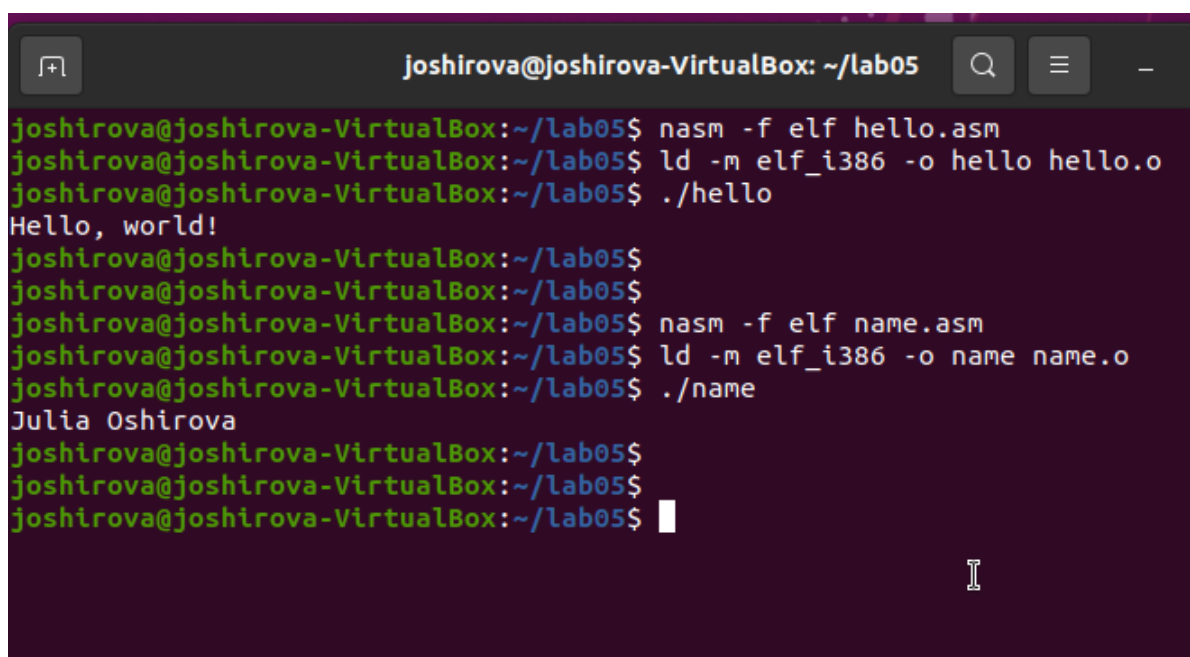
```
joshirova@joshirova-VirtualBox: ~/lab05
joshirova@joshirova-VirtualBox:~/lab05$ nasm -f elf hello.asm
joshirova@joshirova-VirtualBox:~/lab05$ ld -m elf_i386 -o hello hello.o
joshirova@joshirova-VirtualBox:~/lab05$ ./hello
Hello, world!
joshirova@joshirova-VirtualBox:~/lab05$
```

Рис. 4.2: Работа программы hello

4. Изменили сообщение Hello world на свое имя и запустили файл еще раз (рис. 4.3, 4.4)

```
1 SECTION .data
2     hello:      db "Julia Oshirova",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. 4.3: Файл lab05.asm



```
joshirova@joshirova-VirtualBox: ~/lab05
joshirova@joshirova-VirtualBox:~/lab05$ nasm -f elf hello.asm
joshirova@joshirova-VirtualBox:~/lab05$ ld -m elf_i386 -o hello hello.o
joshirova@joshirova-VirtualBox:~/lab05$ ./hello
Hello, world!
joshirova@joshirova-VirtualBox:~/lab05$
joshirova@joshirova-VirtualBox:~/lab05$
joshirova@joshirova-VirtualBox:~/lab05$ nasm -f elf name.asm
joshirova@joshirova-VirtualBox:~/lab05$ ld -m elf_i386 -o name name.o
joshirova@joshirova-VirtualBox:~/lab05$ ./name
Julia Oshirova
joshirova@joshirova-VirtualBox:~/lab05$
joshirova@joshirova-VirtualBox:~/lab05$
joshirova@joshirova-VirtualBox:~/lab05$
```

Рис. 4.4: Работа программы lab05

5 Выводы

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

Список литературы

1. Расширенный ассемблер: NASM
2. MASM, TASM, FASM, NASM под Windows и Linux