

Отчёт по лабораторной работе 4

Архитектура компьютера

Цыкунова Екатерина Михайловна НКАбд-05-24

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Программа Hello world!	9
4.2	Транслятор NASM	10
4.3	Компоновщик LD	11
4.4	Выполнение заданий для самостоятельной работы.	12
5	Выводы	14
	Список литературы	15

Список иллюстраций

4.1	Создание каталога и файла	9
4.2	Программа hello.asm	10
4.3	Трансляция hello.asm	10
4.4	Трансляция hello.asm с дополнительными опциями	11
4.5	Линковка программы	11
4.6	Линковка программы	11
4.7	Запуск программ	12
4.8	Код программы в файле lab4.asm	13
4.9	Запуск программы lab4.asm	13

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.

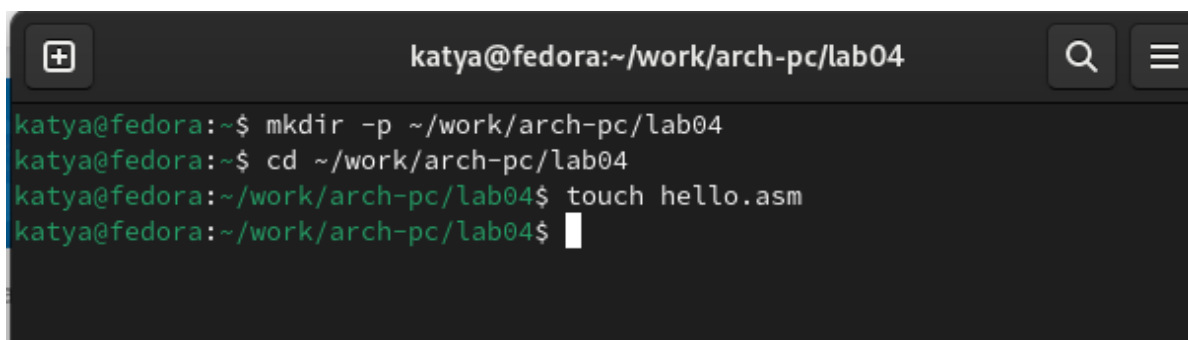
В NASM используется Intel-синтаксис записи инструкций. Предложение языка ассемблера NASM (строка программы) может состоять из следующих элементов: Метка, Инструкция, Операнды, Комментарий.

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

4.1 Программа Hello world!

Создаю каталог lab04 командой `mkdir`, перехожу в него с помощью команды `cd`, создаю файл `hello.asm`. (рис. 4.1)

A screenshot of a terminal window with a dark background. The title bar at the top shows a plus icon on the left, the text 'katya@fedora:~/work/arch-pc/lab04' in the center, and a magnifying glass icon and a hamburger menu icon on the right. The terminal content shows four lines of text: 'katya@fedora:~\$ mkdir -p ~/work/arch-pc/lab04', 'katya@fedora:~\$ cd ~/work/arch-pc/lab04', 'katya@fedora:~/work/arch-pc/lab04\$ touch hello.asm', and 'katya@fedora:~/work/arch-pc/lab04\$' followed by a white cursor. The text is in a green monospaced font.

```
katya@fedora:~$ mkdir -p ~/work/arch-pc/lab04
katya@fedora:~$ cd ~/work/arch-pc/lab04
katya@fedora:~/work/arch-pc/lab04$ touch hello.asm
katya@fedora:~/work/arch-pc/lab04$
```

Рис. 4.1: Создание каталога и файла

Открыла файл и написала код программы по заданию.(рис. 4.2)



```
SECTION .data
hello:      db "Hello, world!",0xa
hellolen:   equ $ - hello

SECTION .text
global _start

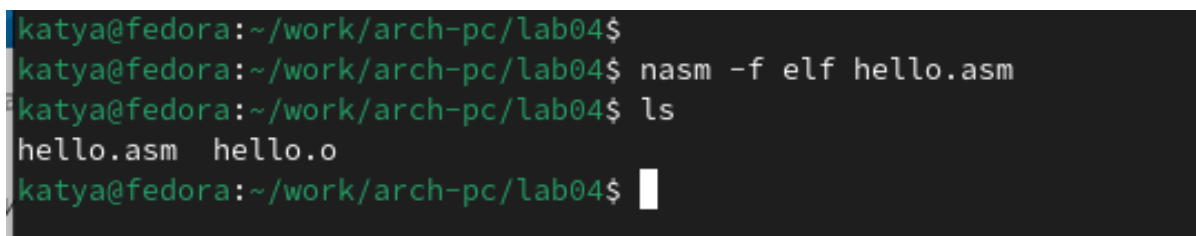
_start:
    mov eax, 4
    mov ebx, 1
    mov ecx, hello
    mov edx, hellolen
    int 0x80

    mov eax, 1
    mov ebx, 0
    int 0x80
```

Рис. 4.2: Программа hello.asm

4.2 Транслятор NASM

Транслирую файл командой `nasm`. Получился объектный файл `hello.o` (рис. 4.3)



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

Рис. 4.3: Трансляция hello.asm

Транслирую файл командой `nasm` с дополнительными опциями. (рис. 4.4) Получился файл листинга `list.lst`, объектный файл `obj.o`, в программу добавилась

отладочная информация.

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

Рис. 4.4: Трансляция hello.asm с дополнительными опциями

4.3 Компоновщик LD

Выполняю линковку командой ld и получил исполняемый файл. (рис. 4.5)

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

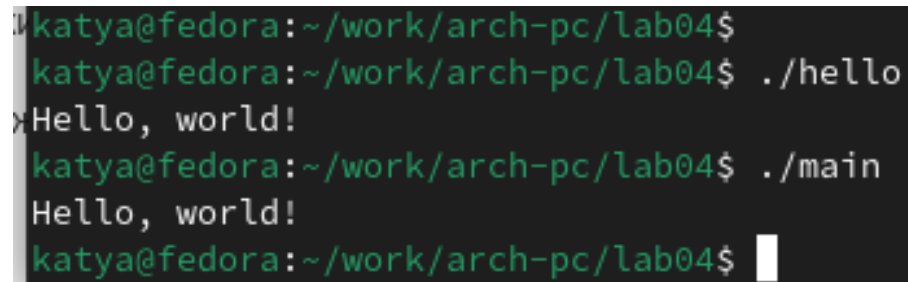
Рис. 4.5: Линковка программы

Еще раз выполняю линковку для объектного файла obj.o и получаю исполняемый файл main. (рис. 4.6)

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

Рис. 4.6: Линковка программы

Запускаю исполняемые файлы. (рис. 4.7)

A terminal window with a dark background and green text. The prompt is 'katya@fedora:~/work/arch-pc/lab04\$'. The first command is './hello', which outputs 'Hello, world!'. The second command is './main', which also outputs 'Hello, world!'. The prompt is now ready for a new command.

```
katya@fedora:~/work/arch-pc/lab04$  
katya@fedora:~/work/arch-pc/lab04$ ./hello  
Hello, world!  
katya@fedora:~/work/arch-pc/lab04$ ./main  
Hello, world!  
katya@fedora:~/work/arch-pc/lab04$
```

Рис. 4.7: Запуск программ

4.4 Выполнение заданий для самостоятельной работы.

Копирую программу в новый файл.

Изменяю сообщение Hello world на свое имя (рис. 4.8) и запускаю новую программу. (рис. 4.9)



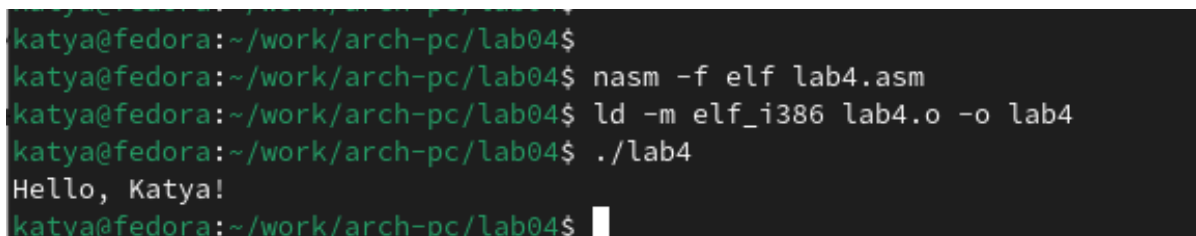
```
SECTION .data
    hello:      db "Hello, Katya!",0xa
               helloLen: equ $ - hello

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
```

Рис. 4.8: Код программы в файле lab4.asm



```
katya@fedora:~/work/arch-pc/lab04$
katya@fedora:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
katya@fedora:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
katya@fedora:~/work/arch-pc/lab04$ ./lab4
Hello, Katya!
katya@fedora:~/work/arch-pc/lab04$
```

Рис. 4.9: Запуск программы lab4.asm

5 Выводы

При выполнении данной лабораторной работы я освоила процесс компиляции и сборки программ, написанных на ассемблере `nasm`.

Список литературы

1. Архитектура ЭВМ
2. NASM Assembly Language Tutorials. — 2021. — URL: <https://asmtutor.com/>.