

Отчёт по лабораторной работе №5

Создание и процесс обработки программ на языке ассемблера NASM

Аскеров Александр Эдуардович

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Программа Hello world!	5
2.2	Транслятор NASM	6
2.3	Расширенный синтаксис командной строки NASM	7
2.4	Компоновщик LD	8
2.5	Запуск исполняемого файла	8
2.6	Задание для самостоятельной работы	9
3	Выводы	11

Список иллюстраций

2.1	Создание каталога для работы с программами	5
2.2	Переход в каталог lab05	5
2.3	Создание файла hello.asm	5
2.4	Ввод текста в файл hello.asm	6
2.5	Компиляция hello.asm и создание объектного файла	7
2.6	Компиляция исходного файла hello.asm в obj.o, а также создание файла листинга list.lst	7
2.7	Проверка того, что файлы созданы	8
2.8	Передача объектного файла на обработку компоновщику	8
2.9	Передача объектного файла на обработку компоновщику	8
2.10	Запуск исполняемого файла на исполнение	9
2.11	Переименование файла hello.asm в lab5.asm	9
2.12	Внесение изменений в файл lab5.asm	9
2.13	Трассировка, компоновка и запуск файла lab5.asm	10
2.14	Копирование файлов hello.asm и lab5.asm	10
2.15	Файлы, загруженные на GitHub	10

1 Цель работы

Освоить процедуры компиляции и сборки программ, написанных на ассемблере NASM.

2 Выполнение лабораторной работы

2.1 Программа Hello world!

Рассмотрим пример простой программы на языке ассемблера NASM. Традиционно первая программа выводит приветственное сообщение Hello world! на экран.

Создадим каталог для работы с программами на языке ассемблера NASM.

```
[aeaskerov@fedora ~]$ mkdir -p ~/work/arch-pc/lab05  
[aeaskerov@fedora ~]$
```

Рис. 2.1: Создание каталога для работы с программами

Перейдём в созданный каталог.

```
[aeaskerov@fedora ~]$ cd ~/work/arch-pc/lab05  
[aeaskerov@fedora lab05]$
```

Рис. 2.2: Переход в каталог lab05

Создадим текстовый файл с именем hello.asm.

```
[aeaskerov@fedora lab05]$ touch hello.asm  
[aeaskerov@fedora lab05]$
```

Рис. 2.3: Создание файла hello.asm

Откроем этот файл с помощью текстового редактора gedit и введём в него следующий текст.

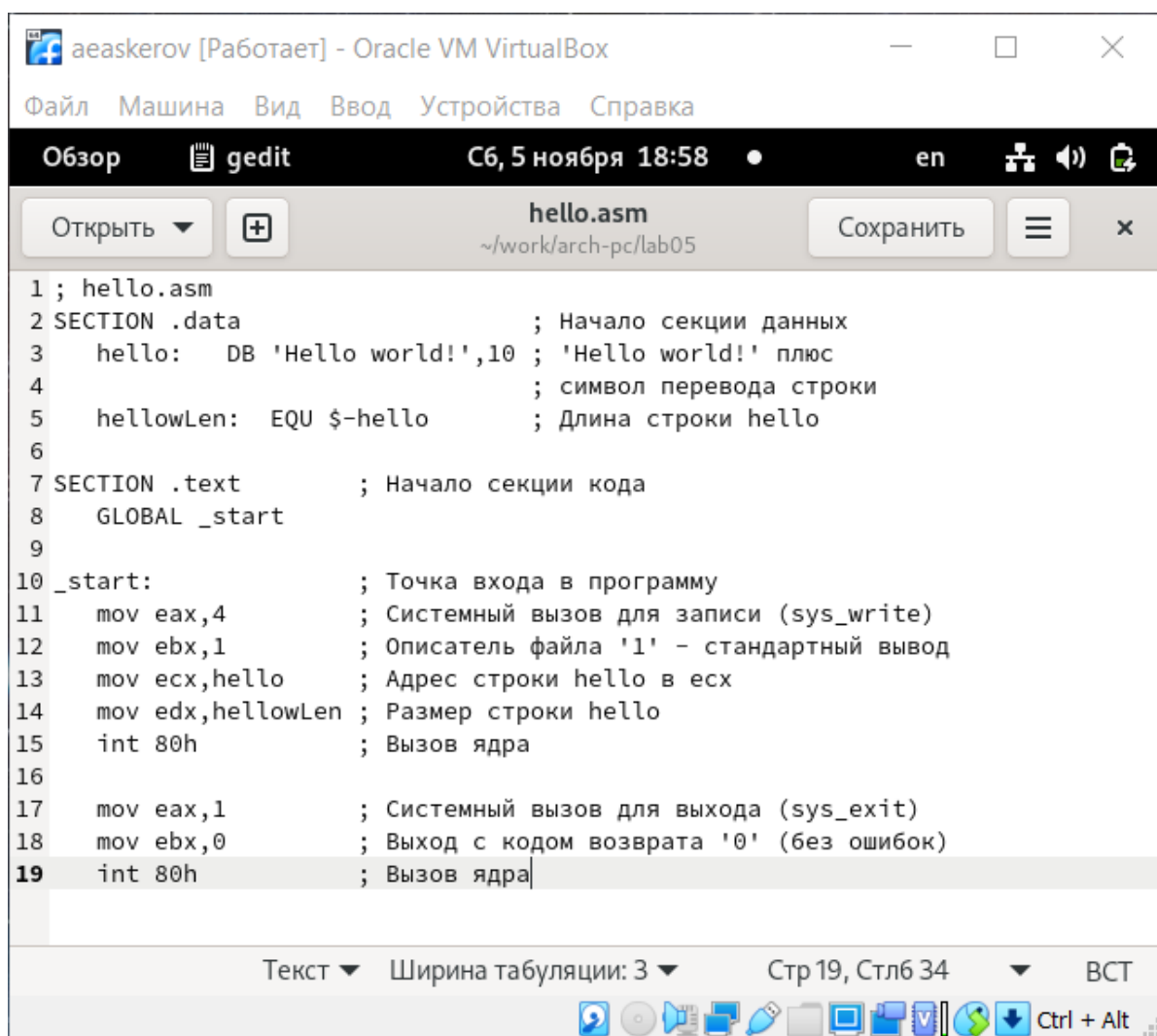


Рис. 2.4: Ввод текста в файл hello.asm

2.2 Транслятор NASM

NASM превращает текст программы в объектный код. Например, для компиляции приведённого выше текста программы «Hello World» необходимо написать: `nasm -f elf hello.asm`.

Если текст программы набран без ошибок, то транслятор преобразует текст программы из файла `hello.asm` в объектный код, который запишется в файл `hello.o`. Таким образом, имена всех файлов получаются из имени входного файла и рас-

ширения по умолчанию. При наличии ошибок объектный файл не создаётся, а после запуска транслятора появятся сообщения об ошибках или предупреждения.

С помощью команды `ls` проверим, что объектный файл был создан. Получившийся объектный файл имеет имя `hello.o`.

```
[aeaskerov@fedora lab05]$ nasm -f elf hello.asm
[aeaskerov@fedora lab05]$ ls
hello.asm  hello.o
[aeaskerov@fedora lab05]$
```

Рис. 2.5: Компиляция `hello.asm` и создание объектного файла

NASM не запускают без параметров. Ключ `-f` указывает транслятору, что требуется создать бинарные файлы в формате ELF. Следует отметить, что формат `elf64` позволяет создавать исполняемый код, работающий под 64-битными версиями Linux. Для 32-битных версий ОС указываем в качестве формата просто `elf`.

NASM всегда создаёт выходные файлы в текущем каталоге.

2.3 Расширенный синтаксис командной строки NASM

Скомпилируем исходный файл `hello.asm` в `obj.o` (опция `-o` позволяет задать имя объектного файла, в данном случае `obj.o`), при этом формат выходного файла будет `elf`, и в него будут включены символы для отладки (опция `-g`), кроме того, будет создан файл листинга `list.lst` (опция `-l`). Для этого выполним следующую команду.

```
[aeaskerov@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[aeaskerov@fedora lab05]$
```

Рис. 2.6: Компиляция исходного файла `hello.asm` в `obj.o`, а также создание файла листинга `list.lst`

С помощью команды `ls` проверим, что файлы были созданы.

```
[aeaskerov@fedora lab05]$ nasm -o obj.o -f elf -g -l list.lst hello.asm
[aeaskerov@fedora lab05]$ ls
hello.asm hello.o list.lst obj.o
[aeaskerov@fedora lab05]$
```

Рис. 2.7: Проверка того, что файлы созданы

2.4 Компоновщик LD

Чтобы получить исполняемую программу, объектный файл необходимо передать на обработку компоновщику. С помощью команды `ls` проверим, что исполняемый файл `hello` был создан.

```
[aeaskerov@fedora lab05]$ ld -m elf_i386 hello.o -o hello
[aeaskerov@fedora lab05]$ ls
hello hello.asm hello.o list.lst obj.o
[aeaskerov@fedora lab05]$
```

Рис. 2.8: Передача объектного файла на обработку компоновщику

Выполним следующую команду: `ld -m elf_i386 obj.o -o main`.

```
[aeaskerov@fedora lab05]$ ld -m elf_i386 obj.o -o main
[aeaskerov@fedora lab05]$ ls
hello hello.asm hello.o list.lst main obj.o
[aeaskerov@fedora lab05]$
```

Рис. 2.9: Передача объектного файла на обработку компоновщику

Исполняемый файл будет иметь имя `main`, а объектный файл, из которого собран этот исполняемый файл, будет иметь имя `obj.o`.

2.5 Запуск исполняемого файла

Запустить на выполнение созданный исполняемый файл, находящийся в текущем каталоге, можно, набрав в командной строке: `./hello`.


```
[aeaskerov@fedora lab05]$ ./hello
Hello world!
[aeaskerov@fedora lab05]$
```

Рис. 2.10: Запуск исполняемого файла на исполнение

2.6 Задание для самостоятельной работы

1. В каталоге `~/work/arch-pc/lab05` с помощью команды `cp` создадим копию файла `hello.asm` с именем `lab5.asm`.

```
[aeaskerov@fedora lab05]$ cp hello.asm lab5.asm
[aeaskerov@fedora lab05]$
```

Рис. 2.11: Переименование файла `hello.asm` в `lab5.asm`

2. С помощью любого текстового редактора внесём изменения в текст программы в файле `lab5.asm` так, чтобы вместо `Hello world!` на экран выводилась строка с фамилией и именем.

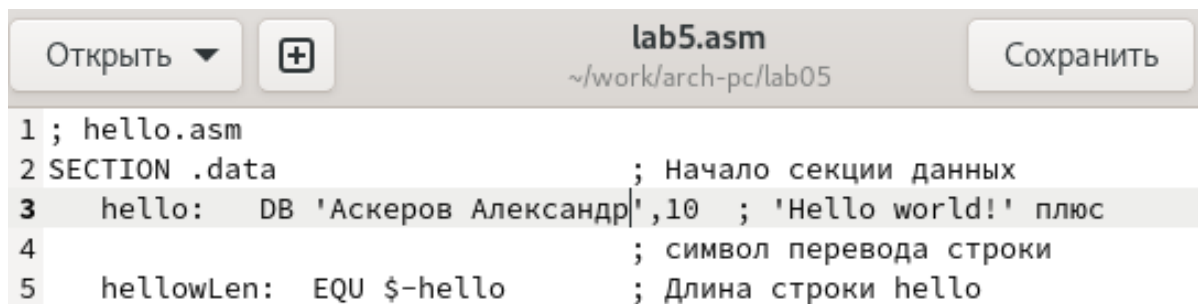


Рис. 2.12: Внесение изменений в файл `lab5.asm`

3. Оттранслируем полученный текст программы `lab5.asm` в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```
[aeaskerov@fedora ~]$ cd ~/work/arch-pc/lab05
[aeaskerov@fedora lab05]$
```

Рис. 2.13: Трассировка, компоновка и запуск файла lab5.asm

4. Скопируем файлы hello.asm и lab5.asm в локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/archpc/labs/lab05/. Загрузим файлы на Github.

```
[aeaskerov@fedora lab05]$ cp hello.asm ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab05/
[aeaskerov@fedora lab05]$ cp lab5.asm ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/labs/lab05/
[aeaskerov@fedora lab05]$
```

Рис. 2.14: Копирование файлов hello.asm и lab5.asm

master study_2022-2023_arh-pc / labs / lab05 /		Go to file	Add file ▾	...
aeaskerov feat(main): add files lab05		809ae4f	1 minute ago	History
..				
presentation	feat(main): add files lab04			8 days ago
report	feat(main): add files lab04			8 days ago
hello.asm	feat(main): add files lab05			1 minute ago
lab5.asm	feat(main): add files lab05			1 minute ago

Рис. 2.15: Файлы, загруженные на GitHub

3 Выводы

Освоены процедуры компиляции и сборки программ, написанных на ассемблере NASM.