

Отчёт по лабораторной работе 4

Архитектура компьютеров и операционные системы

Горелашвили Лия Михайловна НКАбд-02-23

Содержание

1	Цель работы	5
2	Задания	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	9
4.1	Программа Hello world!	9
4.2	Транслятор NASM	10
4.3	Расширенный синтаксис командной строки NASM	11
4.4	Компоновщик LD	11
4.5	Запуск исполняемого файла	12
4.6	Задание для самостоятельной работы	13
5	Выводы	15

Список иллюстраций

4.1	Создан каталог для работы и файл для программы	9
4.2	Редактирование файла hello.asm	10
4.3	Трансляция программы	11
4.4	Трансляция программы с дополнительными опциями	11
4.5	Компоновка программы	12
4.6	Компоновка программы	12
4.7	Запуск программы	12
4.8	Скопировала файл	13
4.9	Редактирование файла lab4.asm	13
4.10	Тестирование программы lab4.asm	14

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.

В процессе создания ассемблерной программы можно выделить четыре шага:

- Набор текста программы в текстовом редакторе и сохранение её в отдельном файле. Каждый файл имеет свой тип (или расширение), который определяет назначение файла. Файлы с исходным текстом программ на языке ассемблера имеют тип asm.
- Трансляция — преобразование с помощью транслятора, например nasm, текста программы в машинный код, называемый объектным. На данном этапе также может быть получен листинг программы, содержащий кроме текста программы различную дополнительную информацию, созданную транслятором. Тип объектного файла — o, файла листинга — lst.

- Компоновка или линковка — этап обработки объектного кода компоновщиком (ld), который принимает на вход объектные файлы и собирает по ним исполняемый файл. Исполняемый файл обычно не имеет расширения. Кроме того, можно получить файл карты загрузки программы в ОЗУ, имеющий расширение tar.
- Запуск программы.

4 Выполнение лабораторной работы

4.1 Программа Hello world!

Создала каталог с названием lab04 используя команду `mkdir`. Затем перешла в созданный каталог, воспользовавшись командой `cd`. Внутри каталога создала файл с именем `hello.asm`, где написала программу.

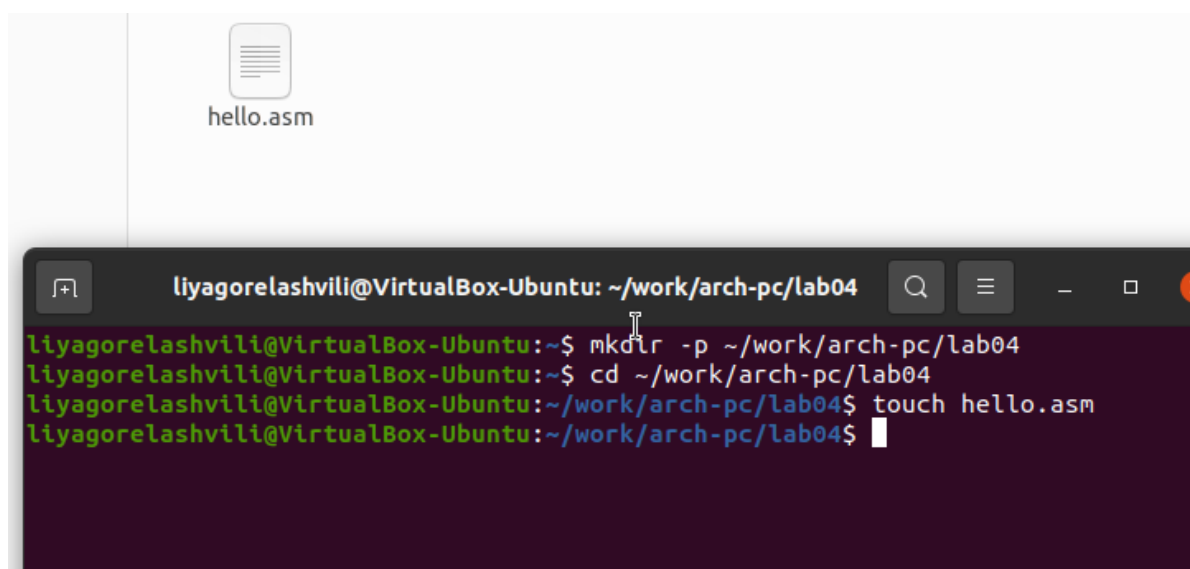
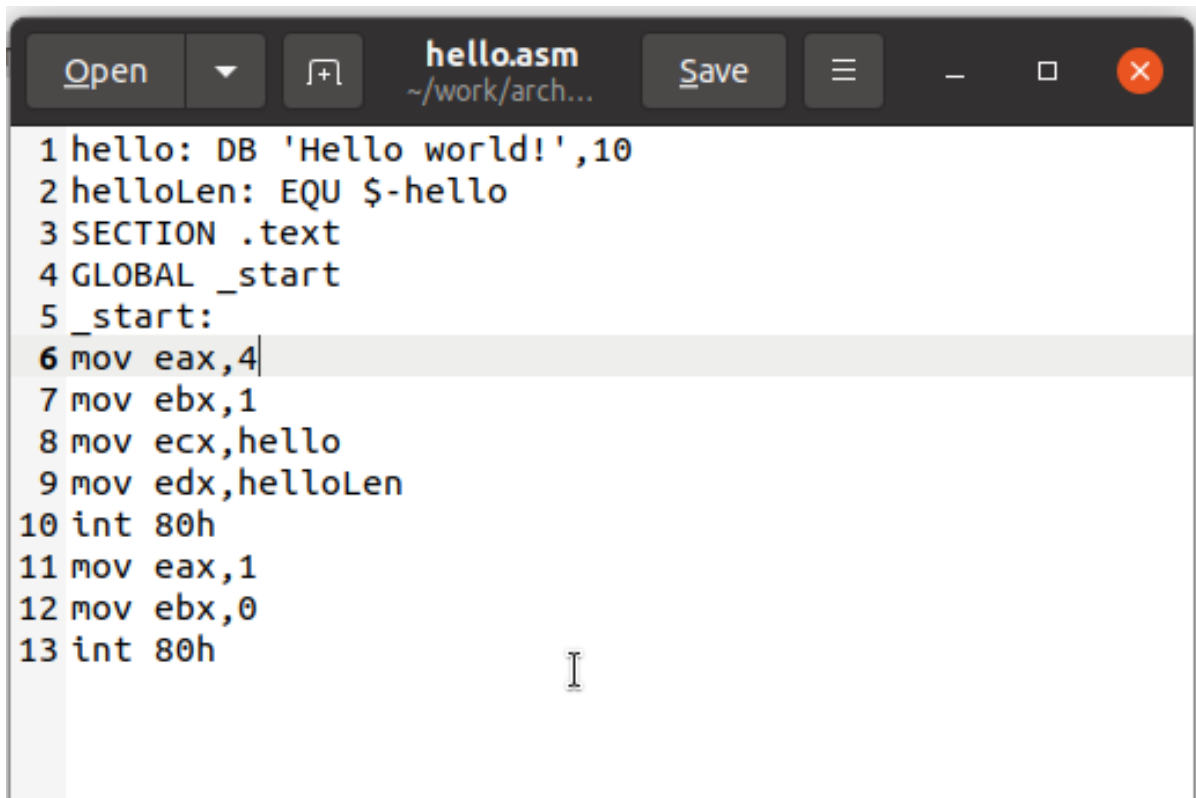


Рис. 4.1: Создан каталог для работы и файл для программы



```
1 hello: DB 'Hello world!',10
2 helloLen: EQU $-hello
3 SECTION .text
4 GLOBAL _start
5 _start:
6 mov eax,4
7 mov ebx,1
8 mov ecx,hello
9 mov edx,helloLen
10 int 80h
11 mov eax,1
12 mov ebx,0
13 int 80h
```

Рис. 4.2: Редактирование файла hello.asm

4.2 Транслятор NASM

Использовала транслятор NASM, чтобы преобразовать текст программы в объектный код. Если текст программы был написан без ошибок, то транслятор преобразовал текст программы из файла hello.asm в объектный код и сохраняет его в файле hello.o.

Транслировала файл с помощью команды `nasm` и получила объектный файл hello.o.

```
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ nasm -f elf hello.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$
```

Рис. 4.3: Трансляция программы

4.3 Расширенный синтаксис командной строки NASM

Полный вариант командной строки для трансляции с использованием NASM выглядит следующим образом

```
nasm [-@ косвенный_файл_настроек] [-o объектный_файл] [-f фор-
мат_объектного_файла] [-l листинг] [параметры...] [--] исходный_файл
```

Транслировала файл с помощью команды `nasm` и использовала дополнительные опции. С опцией `-l` получила файл листинга с именем `list.lst`, с опцией `-f` получила объектный файл с именем `obj.o`, а с опцией `-g` в программу добавилась отладочная информация.

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

Рис. 4.4: Трансляция программы с дополнительными опциями

4.4 Компоновщик LD

Для получения исполняемой программы, объектный файл нужно передать на обработку компоновщику.

Выполнила команду `ld` и получила исполняемый файл с именем `hello` из объектного файла `hello.o`.

```

liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ nasm -f elf hello.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ █

```

Рис. 4.5: Компоновка программы

Еще раз выполнена команда `ld` для объектного файла `obj.o` и получен исполняемый файл с именем `main`

```

liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ nasm -f elf hello.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ █

```

Рис. 4.6: Компоновка программы

4.5 Запуск исполняемого файла

Запустила исполняемые файлы.

```

liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ nasm -f elf hello.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello.asm  hello.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ nasm -o obj.o -f elf -g -l list.lst hello.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello.asm  hello.o  list.lst  obj.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ld -m elf_i386 hello.o -o hello
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  obj.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ld -m elf_i386 obj.o -o main
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls
hello  hello.asm  hello.o  list.lst  main  obj.o
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ./hello
Hello world!
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ./main
Hello world!
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ █

```

Рис. 4.7: Запуск программы

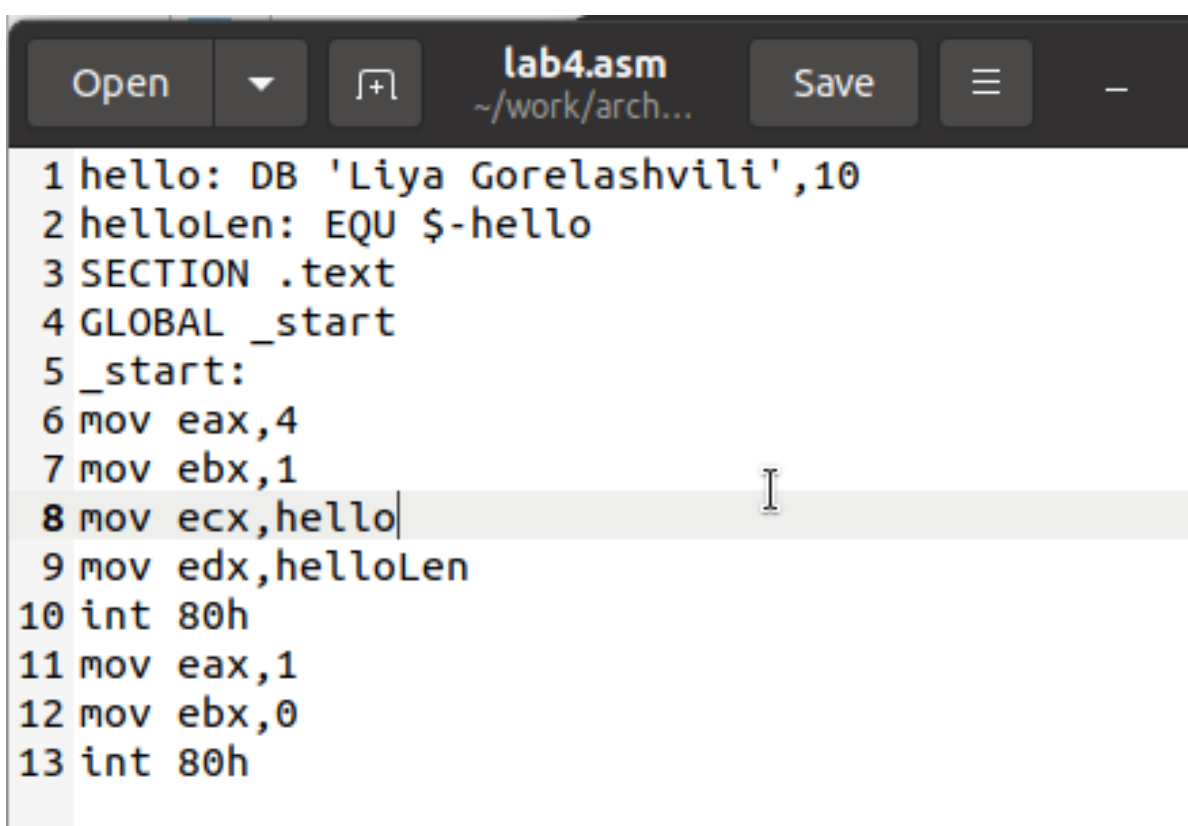
4.6 Задание для самостоятельной работы

Скопировала файл hello.asm в файл lab4.asm.

```
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$  
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ cp hello.asm lab4.asm  
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ls  
hello  hello.asm  hello.o  lab4.asm  list.lst  main  obj.o  
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$
```

Рис. 4.8: Скопировала файл

Изменила сообщение Hello world на свое имя.



```
Open  lab4.asm  Save  
~/work/arch...  
1 hello: DB 'Liya Gorelashvili',10  
2 helloLen: EQU $-hello  
3 SECTION .text  
4 GLOBAL _start  
5 _start:  
6 mov eax,4  
7 mov ebx,1  
8 mov ecx,hello  
9 mov edx,helloLen  
10 int 80h  
11 mov eax,1  
12 mov ebx,0  
13 int 80h
```

Рис. 4.9: Редактирование файла lab4.asm

Запустила программу и проверила.

```
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ nasm -f elf lab4.asm
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ld -m elf_i386 lab4.o -o lab4
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$ ./lab4
Liya Gorelashvili
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$
liyagorelashvili@VirtualBox-Ubuntu:~/work/arch-pc/lab04$
```

Рис. 4.10: Тестирование программы lab4.asm

5 Выводы

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