

Лабораторная работа No5. Создание и процесс обработки программ на языке ассемблера NASM

Дисциплина: Архитектура компьютера

Ли Евгения Олеговна

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	13
	Список литературы	14

Список иллюстраций

4.1	каталог	8
4.2	текстовый файл	8
4.3	текст	9
4.4	текст	9
4.5	Скомпилировала исходный файл hello.asm в obj.o, проверила, что файлы были созданы.	9
4.6	объектный файл передала на обработку компоновщику, проверила, что исполняемый файл hello был создан.	10
4.7	Задала имя создаваемого исполняемого файла.	10
4.8	Запустила на выполнение созданный исполняемый файл	10
4.9	копия	10
4.10	Изменила текст	10
4.11	Изменила текст	11
4.12	Оттранслировала	11
4.13	компоновка	11
4.14	компоновка	11
4.15	запуск	11
4.16	копирование	12
4.17	Загрузка	12

Список таблиц

1 Цель работы

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

2 Задание

Освоить процедуры компиляции и сборки программ, которые написаны на ассемблере NASM.

3 Теоретическое введение

Основные функциональные элементы любой электронно-вычислительной машины (ЭВМ):

центральный процессор, память и периферийные устройства.

Оперативное запоминающее устройство (ОЗУ) — это быстродействующее энергозависимое запоминающее устройство, которое напрямую взаимодействует с узлами процессора, предназначенное для хранения

программ и данных, с которыми процессор непосредственно работает в текущий момент. Состоит из одинаковых пронумерованных ячеек памяти. Номер ячейки памяти — это адрес хранящихся в ней данных.

4 Выполнение лабораторной работы

5.3.1. Программа Hello world!

Я создала каталог для работы с программами на языке ассемблера NASM (рис. 4.1)

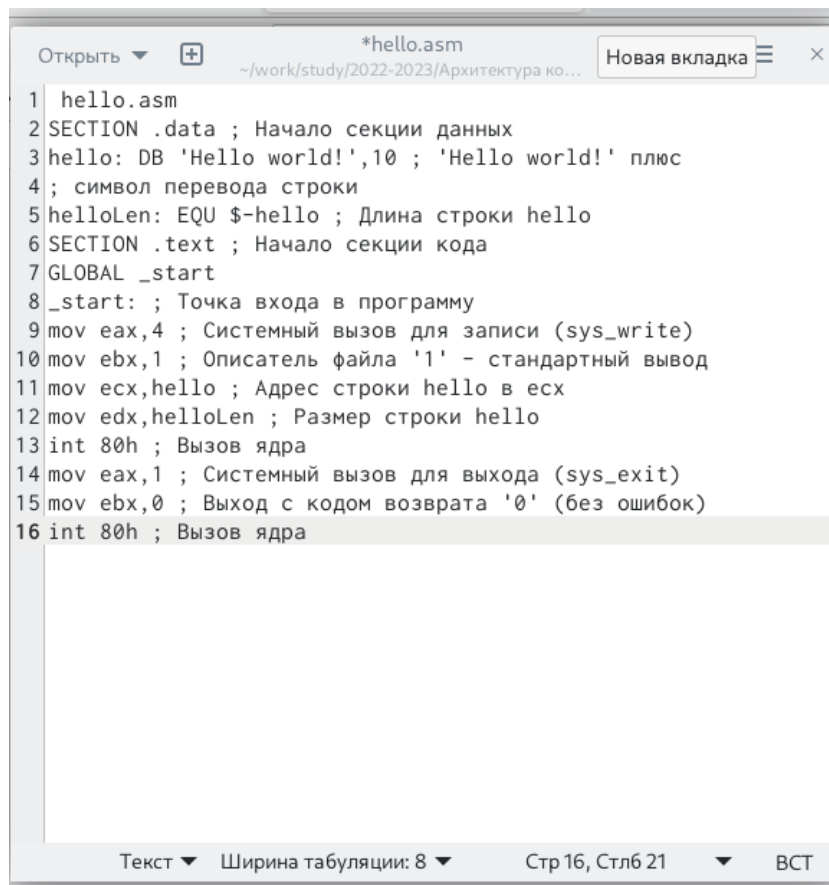
```
eoli@dk4n59 ~ $ mkdir ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05
```

Рис. 4.1: каталог

Перешла в созданный каталог и создала текстовый файл с именем hello.asm и открыла этот файл с помощью текстового редактора gedit и ввела в него требуемый текст: (рис. 4.2, 4.3)

```
eoli@dk4n59 ~ $ cd ~/work/study/2022-2023/"Архитектура компьютера"/arch-pc/lab05
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ touch hello.asm
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ gedit hello.asm
```

Рис. 4.2: текстовый файл

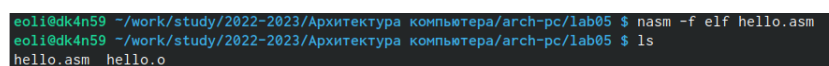


```
1 hello.asm
2 SECTION .data ; Начало секции данных
3 hello: DB 'Hello world!',10 ; 'Hello world!' плюс
4 ; символ перевода строки
5 helloLen: EQU $-hello ; Длина строки hello
6 SECTION .text ; Начало секции кода
7 GLOBAL _start
8 _start: ; Точка входа в программу
9 mov eax,4 ; Системный вызов для записи (sys_write)
10 mov ebx,1 ; Описатель файла '1' - стандартный вывод
11 mov ecx,hello ; Адрес строки hello в ecx
12 mov edx,helloLen ; Размер строки hello
13 int 80h ; Вызов ядра
14 mov eax,1 ; Системный вызов для выхода (sys_exit)
15 mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16 int 80h ; Вызов ядра
```

Рис. 4.3: текст

5.3.2. Транслятор NASM (рис. 4.4)

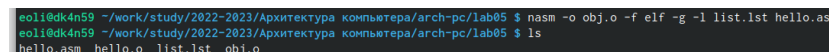
Я скомпилировала текст программы «Hello World» и с помощью команды ls проверила, что объектный файл был создан.



```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf hello.asm
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello.asm  hello.o
```

Рис. 4.4: текст

5.3.3. Расширенный синтаксис командной строки NASM (рис. 4.5)



```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello.asm  hello.o  list.lst  obj.o
```

Рис. 4.5: Скомпилировала исходный файл hello.asm в obj.o, проверила, что файлы были созданы.

5.4. Компоновщик LD(рис. 4.6, 4.7)

```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 hello.o -o hello
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello  hello.asm  hello.o  list.lst  obj.o
```

Рис. 4.6: объектный файл передала на обработку компоновщику, проверила, что исполняемый файл hello был создан.

```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 obj.o -o main
```

Рис. 4.7: Задала имя создаваемого исполняемого файла.

5.4.1. Запуск исполняемого файла (рис. 4.8)

```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./hello
Hello world!
```

Рис. 4.8: Запустила на выполнение созданный исполняемый файл

5.5. Задание для самостоятельной работы

1. С помощью команды `cp` создала копию файла `hello.asm` с именем `lab5.asm` (рис. 4.9)

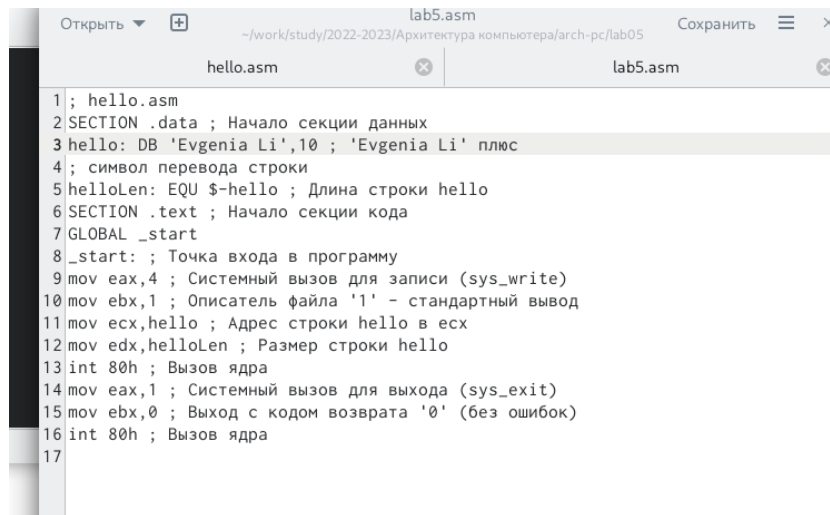
```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp hello.asm lab5.asm
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $
```

Рис. 4.9: копия

2. Внесла изменения в текст программы в файле `lab5.asm` так, чтобы вместо `Hello world!` на экран выводилось `Evgenia Li` (рис. 4.10, 4.11)

```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ gedit lab5.asm
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $
```

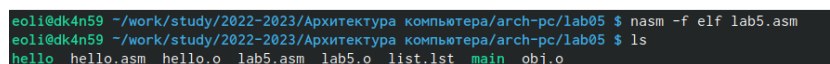
Рис. 4.10: Изменила текст



```
1; hello.asm
2SECTION .data ; Начало секции данных
3hello: DB 'Evgenia Li',10 ; 'Evgenia Li' плюс
4; символ перевода строки
5helloLen: EQU $-hello ; Длина строки hello
6SECTION .text ; Начало секции кода
7GLOBAL _start
8_start: ; Точка входа в программу
9mov eax,4 ; Системный вызов для записи (sys_write)
10mov ebx,1 ; Описатель файла '1' - стандартный вывод
11mov ecx,hello ; Адрес строки hello в ecx
12mov edx,helloLen ; Размер строки hello
13int 80h ; Вызов ядра
14mov eax,1 ; Системный вызов для выхода (sys_exit)
15mov ebx,0 ; Выход с кодом возврата '0' (без ошибок)
16int 80h ; Вызов ядра
17
```

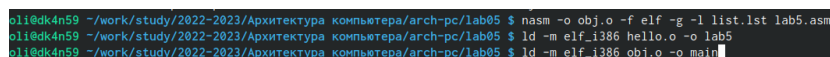
Рис. 4.11: Изменила текст

3. Оттранслировала полученный текст программы lab5.asm в объектный файл. Выполнила компоновку объектного файла и запустила получившийся исполняемый файл. (рис. 4.12, 4.13, 4.14, 4.15)



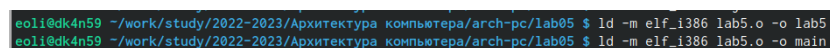
```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -f elf lab5.asm
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ls
hello  hello.asm  hello.o  lab5.asm  lab5.o  list.lst  main  obj.o
```

Рис. 4.12: Оттранслировала



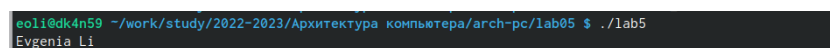
```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l list.lst lab5.asm
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 hello.o -o lab5
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 obj.o -o main
```

Рис. 4.13: компоновка



```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 lab5.o -o lab5
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ld -m elf_i386 lab5.o -o main
```

Рис. 4.14: компоновка



```
eoli@dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ ./lab5
Evgenia Li
```

Рис. 4.15: запуск

4. Скопировала файлы hello.asm и lab5.asm в локальный репозиторий в каталог ~/work/study/2022-2023/“Архитектура компьютера”/arch-pc/labs/lab05/. Загрузила файлы на Github (рис. [-fig. 4.16; -fig. 4.17)

```
eoli0dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp hello.asm ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
eoli0dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab05 $ cp lab5.asm ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05
```

Рис. 4.16: копирование

```
eoli0dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ git add .
eoli0dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ git commit -am 'lab5.asm hello.asm'
[master cf2eb8c] lab5.asm hello.asm
2 files changed, 34 insertions(+)
create mode 100644 labs/lab05/hello.asm
create mode 100644 labs/lab05/lab5.asm
eoli0dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $ git push
Перечисление объектов: 100% (18/18), готово.
Подсчет объектов: 100% (18/18), готово.
При сжатии изменений используется до 6 потоков
Сжатие объектов: 100% (15/15), готово.
Запись объектов: 100% (15/15), 3.10 КиБ | 3.10 МБ/с, готово.
Всего 15 (изменений 7), повторно использовано 0 (изменений 0), повторно использовано пакетов 0
remote: Resolving deltas: 100% (7/7), completed with 2 local objects.
To github.com:evgeniali/study_2022-2023_ar-h-pc.git
 f8739d9..cf2eb8c master -> master
eoli0dk4n59 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/labs/lab05 $
```

Рис. 4.17: Загрузка

5 Выводы

Я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.

Список литературы