

Отчёт по лабораторной работе №4

Создание и процесс обработки программ на языке ассемблера NASM

Дорохов Данила Антонович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Вывод	11

Список иллюстраций

3.1	Создание каталога	7
3.2	Переход в каталог	7
3.3	Создание текстового файла	7
3.4	Открытие файла	7
3.5	Ввод текста	8
3.6	Компиляция текста	8
3.7	Проверка, что объектный файл был создан	8
3.8	Создание файлов	8
3.9	Проверка, что файлы были созданы	8
3.10	Передача файла на компоновку	9
3.11	Проверка, что исполняемый файл hello был создан	9
3.12	Зададим имя создаваемого исполняемого файла	9
3.13	Запуск на выполнение созданный исполняемый файл	9
3.14	Создание копии файла с именем lab4.asm	9
3.15	Внесение изменения в текст программы	10
3.16	Оттранслирование, компоновка, запуск	10

Список таблиц

1 Цель работы

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

2 Задание

Здесь приводится описание задания в соответствии с рекомендациями методического пособия и выданным вариантом.

3 Выполнение лабораторной работы

1. Создайте каталог для работы с программами на языке ассемблера NASM:

```
dadorokhov@dk8n69 ~ $ mkdir ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab04
```

Рис. 3.1: Создание каталога

2. Перейдём в созданный каталог:

```
dadorokhov@dk8n69 ~ $ cd ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab04
```

Рис. 3.2: Переход в каталог

3. Создадим текстовый файл с именем hello.asm:

```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ touch hello.asm
```

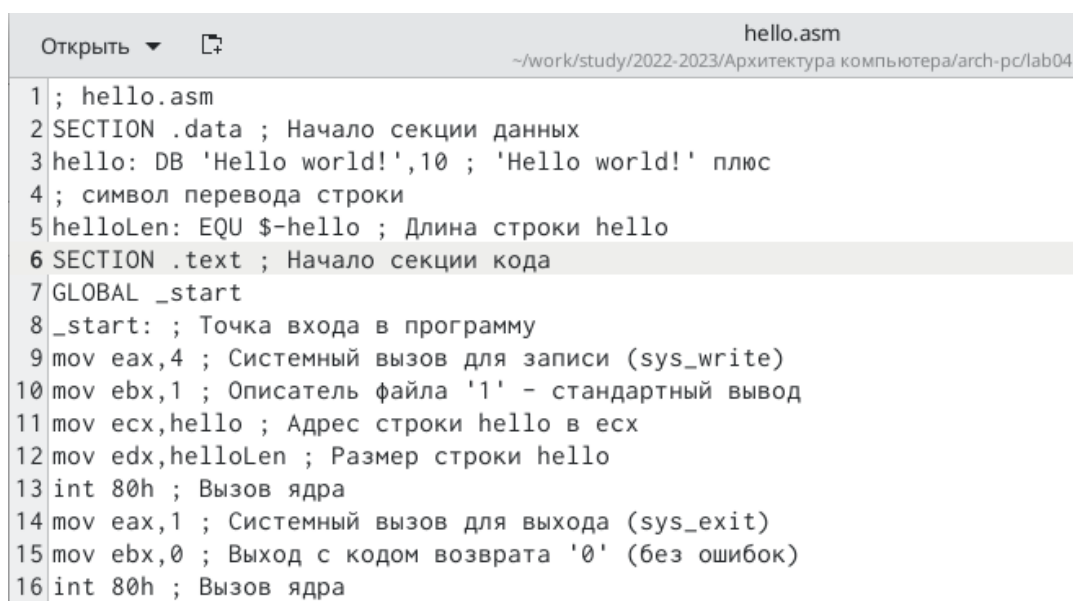
Рис. 3.3: Создание текстового файла

4. Откроем этот файл с помощью текстового редактора

```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ gedit hello.asm
```

Рис. 3.4: Открытие файла

5. Введём в него текст:



```
Открыть ▾  hello.asm
~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04
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. 3.5: Ввод текста

6. Скомпилируем данный текст



```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ nasm -f elf hello.asm
```

Рис. 3.6: Компиляция текста

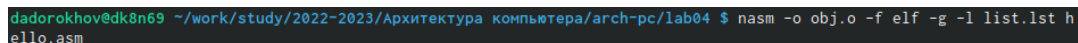
7. Проверим, что объектный файл был создан:



```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ls hello.asm hello.o
```

Рис. 3.7: Проверка, что объектный файл был создан

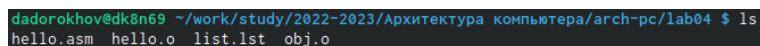
8. Скомпилируем исходный файл hello.asm в obj.o и создадим файл листинга list.lst



```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
```

Рис. 3.8: Создание файлов

9. Проверим, что файлы были созданы



```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ls
hello.asm hello.o list.lst obj.o
```

Рис. 3.9: Проверка, что файлы были созданы

10. Передадим объектный файл на обработку компоновщику

```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ld -m elf_i386 hello.o -o hello
```

Рис. 3.10: Передача файла на компоновку

11. Проверим, что исполняемый файл hello был создан

```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ls
hello hello.asm hello.o list.lst obj.o
```

Рис. 3.11: Проверка, что исполняемый файл hello был создан

12. Зададим имя создаваемого исполняемого файла

```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ld -m elf_i386 obj.o -o main
```

Рис. 3.12: Зададим имя создаваемого исполняемого файла

13. Запустим на выполнение созданный исполняемый файл, находящийся в текущем каталоге

```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ./hello
Hello world!
```

Рис. 3.13: Запуск на выполнение созданный исполняемый файл

14. Создадим копию файла hello.asm с именем lab4.asm

```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ cp hello.asm lab04.asm
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ gedit lab04.asm
```

Рис. 3.14: Создание копии файла с именем lab4.asm

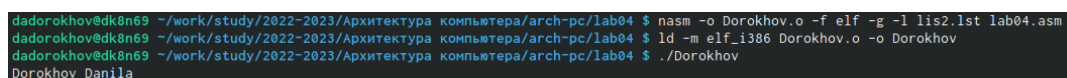
15. Внесём изменения в текст программы в файле lab5.asm



```
Открыть ▾  lab04.asm
~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04
1 ; hello.asm
2 SECTION .data
3 hello: DB 'Dorokhov Danila',10
4 helloLen: EQU $-hello
5 SECTION .text
6 GLOBAL _start
7 _start:
8 mov eax,4
9 mov ebx,1
10 mov ecx,hello
11 mov edx,helloLen
12 int 80h
13 mov eax,1
14 mov ebx,0
15 int 80h
```

Рис. 3.15: Внесение изменения в текст программы

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



```
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ nasm -o Dorokhov.o -f elf -g -l lis2.lst lab04.asm
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ld -m elf_i386 Dorokhov.o -o Dorokhov
dadorokhov@dk8n69 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ./Dorokhov
Dorokhov Danila
```

Рис. 3.16: Оттранслирование, компоновка, запуск

17. Скопируем файлы hello.asm и lab5.asm в локальный репозиторий и загрузим файлы на Github

4 Вывод

За время работы я научился написанию программы на ассемблере NASM.