

Отчёт по лабораторной работе №4

Создание и процесс обработки программ на языке ассемблера NASM

Луковкина Мария Дмитриевна

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	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	9
3.9	Проверка, что файлы были созданы.	9
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	10
3.15	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:

```
mdlukovkina@dk3n53 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Создание каталога

2. Перейдём в созданный каталог:

```
mdlukovkina@dk3n55 ~ $ cd ~/work/arch-pc/lab04
mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ touch
```

Рис. 3.2: Переход в каталог

3. Создадим текстовый файл с именем hello.asm:

```
mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ touch hello.asm
```

Рис. 3.3: Создание текстового файла

4. Откроем этот файл с помощью текстового редактора

```
mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ gedit hello.asm
```

Рис. 3.4: Открытие файла

5. Введём в него текст:

```

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

```

Рис. 3.5: Ввод текста

6. Скомпилируем данный текст



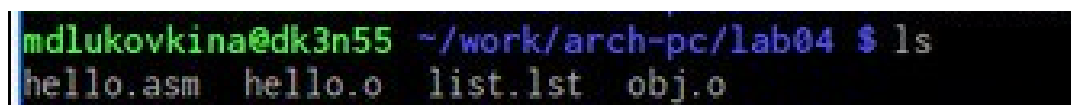
```

mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ nasm -f elf hello.asm

```

Рис. 3.6: Компиляция текста

7. Проверим, что объектный файл был создан:



```

mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ ls
hello.asm hello.o list.lst obj.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


```
mdlukovkina@dk3n55 ~/work/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.

```
mdlukovkina@dk3n55 ~/work/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.

```
mdlukovkina@dk3n55 ~/work/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.

```
mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ ls
hello hello.asm hello.o list.lst main obj.o
```

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

12. Зададим имя создаваемого исполняемого файла.

```
mdlukovkina@dk3n55 ~/work/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.

```
mdlukovkina@dk3n55 ~/work/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

```
mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ nasm -f elf lab04.asm
```

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

15. Оттранслируем полученный текст программы lab4.asm в объектный файл. Выполним компоновку объектного файла и запустим получившийся исполняемый файл.

```
mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ nasm -o obj.o -f elf -g -l list.list lab04.asm
mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 obj.o -o lab04
mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ ld -m elf_i386 lab04.o -o lab04
mdlukovkina@dk3n55 ~/work/arch-pc/lab04 $ ./lab04
Maria Lukovkina
```

Рис. 3.15: Оттранслирование, компоновка, запуск

16. Скопируем файлы hello.asm и lab4.asm в локальный репозиторий и загрузим файлы на Github.

4 Выводы

В ходе выполнения работы, я освоила процедуры компиляции и сборки программ, написанных на ассемблере NASM.