

Отчёт по лабораторной работе№4

Создание и процесс обработки программ на языке ассемблера NASM

Журавлев Даниил Павлович

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	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:

```
dpzhuravlev@dk3n55 ~ $ mkdir -p ~/work/arch-pc/lab04
```

Рис. 3.1: Создание каталога

2. Перейдём в созданный каталог:

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

Рис. 3.2: Переход в каталог

3. Создадим текстовый файл с именем hello.asm:

```
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ touch hello.asm
```

Рис. 3.3: Создание текстового файла

4. Откроем этот файл с помощью текстового редактора

```
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ gedit hello.asm
```

Рис. 3.4: Открытие файла

5. Введём в него текст:

```

1 |SECTION .data
2 |    hello:      db "Hello, world!",0xa
3 |              helloLen: equ $ - hello
4 |SECTION .text
5 |    global _start
6 |
7 |_start:
8 |    mov eax, 4
9 |    mov ebx, 1
10 |    mov ecx, hello
11 |    mov edx, helloLen
12 |    int 0x80
13 |
14 |    mov eax, 1
15 |    mov ebx, 0
16 |    int 0x80
17 |

```

Рис. 3.5: Ввод текста

6. Скомпилируем данный текст

```

dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ nasm -f elf hello.asm

```

Рис. 3.6: Компиляция текста

7. Проверим, что объектный файл был создан:

```

dpzhuravlev@dk3n55 ~/work/arch-pc/lab4 $ 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

```

dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ nasm -o obj.o -f elf -g -l list.lst hello.asm

```

Рис. 3.8: Создание файлов

9. Проверим, что файлы были созданы.

```

dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ ls
hello  hello.asm  hello.o  list.lst  main  obj.o  presentation  report

```

Рис. 3.9: Проверка, что файлы были созданы.

10. Передадим объектный файл на обработку компоновщику.

```
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ ld -m elf_i386 hello.o -o hello
```

Рис. 3.10: Передача файла на компоновку

11. Проверим, что исполняемый файл hello был создан.

```
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ ls
hello hello.asm hello.o list.lst main obj.o presentation report
```

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

12. Зададим имя создаваемого исполняемого файла.

```
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/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.

```
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ ./hello
Hello, world!
```

Рис. 3.13: Запуск на выполнение созданный исполняемый файл

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

```
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ cp hello.asm lab04.asm
```

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

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

```

1 |SECTION .data
2 |    hello:      db "Zhuravlev Daniil",0xa
3 |    helloLen:   equ $ - hello
4 |SECTION .text
5 |    global _start
6 |
7 |_start:
8 |    mov eax, 4
9 |    mov ebx, 1
10 |    mov ecx, hello
11 |    mov edx, helloLen
12 |    int 0x80
13 |
14 |    mov eax, 1
15 |    mov ebx, 0
16 |    int 0x80
17 |

```

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.

```

dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ gedit lab04.asm
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ nasm -f elf lab04.asm
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ nasm -o obj.o -f elf -g -l list.lst lab04.asm
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ ld -m elf_i386 obj.o -o lab04
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ ld -m elf_i386 lab04.o -o lab04
dpzhuravlev@dk3n63 ~/work/study/2024-2025/Архитектура компьютера/study_2024-2025_arch-pc/labs/lab04 $ ./lab04
Zhuravlev Daniil

```

Рис. 3.16: Оттранслирование, компоновка, запуск

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

4 Выводы

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