

# **Отчёт по лабораторной работе №5**

**Компьютерные науки и технология программирования**

Сячинова Ксения Ивановна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
<b>3</b>	<b>Задания для самостоятельной работы</b>	<b>10</b>
<b>4</b>	<b>Выводы</b>	<b>12</b>

# Список иллюстраций

2.1	Создание и переход в каталог . . . . .	6
2.2	Создание текстового файла . . . . .	6
2.3	Текст программы . . . . .	6
2.4	Преобразование текста . . . . .	7
2.5	Компиляция файлы . . . . .	7
2.6	Ввод команды . . . . .	7
2.7	Вывод команды . . . . .	7
2.8	Ввод и вывод команды . . . . .	8
2.9	Передача файла на компоновку . . . . .	8
2.10	Выполнение команды . . . . .	8
2.11	Формат командной строки . . . . .	8
2.12	Ввод команды . . . . .	9
2.13	Выполнение команды . . . . .	9
2.14	Запуск программы . . . . .	9
3.1	Копирование и переименование файла . . . . .	10
3.2	Переход в тестовый файл . . . . .	10
3.3	Изменение программы . . . . .	11
3.4	Итог программы . . . . .	11

## Список таблиц

# 1 Цель работы

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

## 2 Выполнение лабораторной работы

1. Создаём каталог для работы с программами на языке ассемблера NASM и переходим в него.

```
kisyachinova1@dk6n58 ~ $ cd work
kisyachinova1@dk6n58 ~/work $ mkdir arch-pc
kisyachinova1@dk6n58 ~/work $ cd arch-pc
kisyachinova1@dk6n58 ~/work/arch-pc $ mkdir lab05
kisyachinova1@dk6n58 ~/work/arch-pc $ cd lab05
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $
```

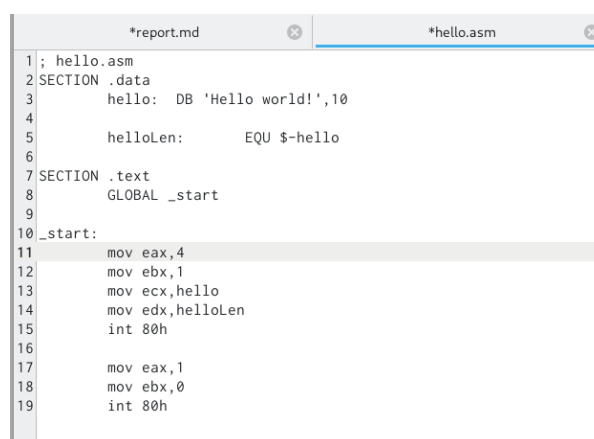
Рис. 2.1: Создание и переход в каталог

Затем создаём текстовый файл с 'hello.asm' и открываем его с помощью текстового редактора 'gedit'

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ touch hello.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ gedit hello.asm
```

Рис. 2.2: Создание текстового файла

Вводим следующий текст в текстовом документе.



```
*report.md *hello.asm
1; hello.asm
2SECTION .data
3    hello: DB 'Hello world!',10
4
5    helloLen: EQU $-hello
6
7SECTION .text
8    GLOBAL _start
9
10 _start:
11    mov eax,4
12    mov ebx,1
13    mov ecx,hello
14    mov edx,helloLen
15    int 80h
16
17    mov eax,1
18    mov ebx,0
19    int 80h
```

Рис. 2.3: Текст программы

2. После этого превращаем текст программы в объектный код. Для этого используем команду 'nasm -f elf hello.asm'. Объектный файл создан, т.к. текст написан без ошибок. Транслятор преобразовал объектный код, который был записан в файл 'hello.o'. Проверяем это с помощью команды 'ls'.

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ nasm -f elf hello.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ ls
hello.asm  hello.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $
```

Рис. 2.4: Преобразование текста

3. После этого выполним следующую команду: 'nasm -o obj.o -f elf -g -l list.lst hello.asm'. Данная команда скомпилирует исходный файл в obj.o. Также, этой же командой создаётся файл 'list.lst'. Команда 'ls' помогает нам проверить создание файлов.

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ nasm -o obj.o -f elf -g -l list.lst hello.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ ls
hello.asm  hello.o  list.lst  obj.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $
```

Рис. 2.5: Компиляция файлы

Для более подробной информации может использовать команду 'man nasm'.

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ man nasm
```

Рис. 2.6: Ввод команды

```
NASM(1)                                The Netwide Assembler Project                                NASM(1)
NAME
    nasm - the Netwide Assembler, a portable 80x86 assembler

SYNOPSIS
    nasm [-@ response file] [-f format] [-o outfile] [-l listfile] [options...] filename

DESCRIPTION
    The nasm command assembles the file filename and directs output to the file outfile if specified. If outfile is not
    specified, nasm will derive a default output file name from the name of its input file, usually by appending '.o'
    or '.obj', or by removing all extensions for a raw binary file. Failing that, the output file name will be
    'nasm.out'.

OPTIONS
    -@ filename
        Causes nasm to process options from filename as if they were included on the command line.

    -a
        Causes nasm to assemble the given input file without first applying the macro preprocessor.
```

Рис. 2.7: Вывод команды

Для получения списка форматов объектного файла используем 'nasm -hf'.

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ nasm -h
Usage: nasm [-@ response_file] [options...] [--] filename
       nasm -v (or --v)

Options (values in brackets indicate defaults):
  -h                show this text and exit (also --help)
  -v (or --v)       print the NASM version number and exit
  -@ file           response file; one command line option per line
  -o outfile        write output to outfile
  --keep-all       output files will not be removed even if an error happens
  -xformat          specify error reporting format (gnu or vc)
  -s               redirect error messages to stdout
  -Zfile           redirect error messages to file
```

Рис. 2.8: Ввод и вывод команды

4. Затем необходимо объектный файл передать на обработку компоновщику. Это можно делать с помощью команды 'ld -m elf\_i386 hello.o -o hello'. Проверяем с помощью команды 'ls'.

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ ld -m elf_i386 hello.o -o hello
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $ ls
hello hello.asm hello.o list.lst obj.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab05 $
```

Рис. 2.9: Передача файла на компоновку

Затем выполняем команду 'ld -m elf\_i386 obj.o -o main'. Получаем исполняемый файл, который имеет название 'hello'. Объектный файл из которого собран этот исполняемый файл называется hello.o.

```
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 obj.o -o main
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ ls
hello hello.asm hello.o list.lst main obj.o
```

Рис. 2.10: Выполнение команды

Команда 'ld -help' позволяет нам увидеть формат командной строки LD. Для получения более подробной информации можно использовать команду 'man ld'.

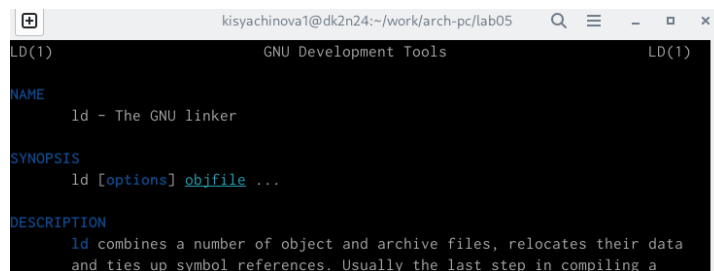
```
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ ld --help
Использование ld [параметры] файл...
Параметры:
  -a КЛЮЧЕВОЕ СЛОВО                                Управление общей библиотекой для совместимости с H
P/UX
  -A АРХИТЕКТУРА, --architecture АРХИТЕКТУРА      Задать архитектуру
  -b ЦЕЛЬ, --format ЦЕЛЬ                            Задать цель для следующих входных файлов
  -c ФАЙЛ, --mri-script ФАЙЛ                       Прочитать сценарий компоновщика в формате MRI
```

Рис. 2.11: Формат командной строки



```
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ man ld
```

Рис. 2.12: Ввод команды



```
LD(1) GNU Development Tools LD(1)
NAME
ld - The GNU linker
SYNOPSIS
ld [options] objfile ...
DESCRIPTION
ld combines a number of object and archive files, relocates their data
and ties up symbol references. Usually the last step in compiling a
```

Рис. 2.13: Выполнение команды

5. Для запуска программы вводим в командной строке './hello'. Программа работает корректно.

```
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ ./hello
Hello world!
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $
```

Рис. 2.14: Запуск программы

### 3 Задания для самостоятельной работы

1. В каталоге, созданном для лабораторной работы №5, создаём копию файла hello.asm с именем lab5.asm. Для этого используем команду 'cp'.

```
kisyachinova1@dk2n24 ~ $ cp work/arch-pc/lab05/hello.asm work/arch-pc/lab05/lab5.asm
kisyachinova1@dk2n24 ~ $ cd work
kisyachinova1@dk2n24 ~/work $ cd arch-pc
kisyachinova1@dk2n24 ~/work/arch-pc $ cd lab05
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ ls
hello  hello.asm  hello.o  lab5.asm  list.lst  main  obj.o
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $
```

Рис. 3.1: Копирование и переименование файла

2. Затем изменяем программу так, чтобы вместо "Hello world!" выводилась фамилия и имя.

```
hello  hello.asm  hello.o  lab5.asm  list.lst  main  obj.o
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ gedit lab5.asm
```

Рис. 3.2: Переход в тестовый файл

```

1 SECTION .data
2     hello:    db "Kseniya Syachinova",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

```

Рис. 3.3: Изменение программы

3. Выполняем необходимые действия: трансляция в объектный файл, компоновка объектного файла. После этого запускаем нашу программу.

```

kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ nasm -f elf lab5.asm
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ nasm -o obj1.o -f elf -g -l list1.lst lab5.asm
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 lab5.o -o lab5
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ ld -m elf_i386 obj1.o -o main
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ ./lab5
Kseniya Syachinova
kisyachinova1@dk2n24 ~/work/arch-pc/lab05 $ 

```

Рис. 3.4: Итог программы

## 4 Выводы

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