

# **Отчёт по лабораторной работе№4**

**Создание и процесс обработки программ на языке ассемблера NASM**

Степан Михайлович Токаев

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>4</b>	<b>Выводы</b>	<b>11</b>

## 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:

```
smtokaev@dk3n40 ~ $ mkdir ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab04
```

Рис. 3.1: Создание каталога

2. Перейдём в созданный каталог:

```
smtokaev@dk3n40 ~ $ cd ~/work/study/2022-2023/Архитектура\ компьютера/arch-pc/lab04
```

Рис. 3.2: Переход в каталог

3. Создадим текстовый файл с именем hello.asm:

```
smtokaev@dk3n40 ~/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

```
smtokaev@dk3n40 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ gedit hello.asm
```

Рис. 3.4: Открытие файла

5. Введём в него текст:

```

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

```

smtokaev@dk3n40 ~/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:

```

smtokaev@dk3n40 ~/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

```

smtokaev@dk3n40 ~/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.

```

smtokaev@dk3n40 ~/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.

```
smtokaev@dk3n40 ~/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.

```
smtokaev@dk3n40 ~/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.

```
smtokaev@dk3n40 ~/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.

```
smtokaev@dk3n40 ~/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

```
smtokaev@dk3n40 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ cp hello.asm lab4.asm
```

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

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

```

; hello.asm
SECTION .data ; Начало секции данных
hello: DB 'Tokaev Stepan',10
helloLen: EQU $-hello
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,1
mov ecx,hello
mov edx,helloLen
int 80h
mov eax,1
mov ebx,0
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.

```

smtokaev@dk3n40 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ gedit lab4.asm
smtokaev@dk3n40 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ nasm -o Tokaev.o -f elf -g -l list2.lst lab4.asm
smtokaev@dk3n40 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ld -m elf_i386 Tokaev.o -o Tokaev
smtokaev@dk3n40 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ ./Tokaev
Tokaev Stepan
smtokaev@dk3n40 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab04 $ █

```

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.