

# **Отчёт по лабораторной работе №8**

**Простейший вариант**

Янушкевич Михаил денисович

# 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>Задание для самостоятельной работы</b>	<b>18</b>
<b>4</b>	<b>Выводы</b>	<b>21</b>
	<b>Список литературы</b>	<b>22</b>

## Список иллюстраций

2.1	Создание файла . . . . .	6
2.2	Ввод программы . . . . .	7
2.3	Создание исполняемого файла, проверка программы . . . . .	8
2.4	Изменение текста программы . . . . .	9
2.5	Создание исполняемого файла, проверка программы . . . . .	9
2.6	Изменение текста программы . . . . .	10
2.7	Создание исполняемого файла, проверка программы . . . . .	11
2.8	Ввод программы . . . . .	12
2.9	Создание исполняемого файла, проверка программы . . . . .	13
2.10	Ввод программы . . . . .	14
2.11	Создание исполняемого файла, проверка программы . . . . .	15
2.12	Изменение программы . . . . .	16
2.13	Создание исполняемого файла, проверка программы . . . . .	17
3.1	Написание программы . . . . .	19
3.2	Создание исполняемого файла, проверка программы . . . . .	20

## Список таблиц

# 1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

## 2 Выполнение лабораторной работы

1. Создать каталог ЛБ №8. В нём создать файл lab8-1.asm.(рис. 2.1)

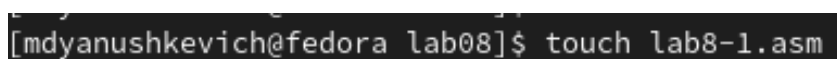
A terminal window showing the command `touch lab8-1.asm` being executed in the directory `lab08` on a Fedora system. The prompt is `[mdyanushkevich@fedora lab08]$`.

Рис. 2.1: Создание файла

С помощью команды `mkdir` создаём каталог каталог для программ лабораторной работы #8. Переходим в него и с помощью команды `touch` создаём файл `lab8-1.asm`.

2. В файл `lab8-1.asm` ввести код программы из листинга 8.1.рис.(2.2)

```

%include 'in_out.asm'

SECTION .data
    msg1 db 'Введите N: ',0h

SECTION .bss
    N: resb 10

SECTION .text
    global _start

_start:
    mov eax,msg1
    call sprint

    mov ecx, N
    mov edx, 10
    call sread

    mov eax,N
    call atoi
    mov [N],eax

    mov ecx,[N]
label:
    mov [N],ecx
    mov eax,[N]
    call iprintLF
    pop ecx
    loop label

    call quit

```

Рис. 2.2: Ввод программы

С помощью текстового редактора открываем файл lab8-1.asm, далее в него вводим текст программы из листинга 8.1.

### 3. Создать исполняемый файл и проверить его работу.(рис. 2.3)

```
[mdyanushkevich@fedora lab08]$ nasm -f elf lab8-1.asm
[mdyanushkevich@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[mdyanushkevich@fedora lab08]$ ./lab8-1
Введите N: 11
11
10
9
8
7
6
5
4
3
2
1
```

Рис. 2.3: Создание исполняемого файла, проверка программы

В командную строку вводим команды, чтобы создать исполняемый файл. Далее запускаем его. Результатом работы программы стала последовательность чисел от 11 до 1.

### 4. Изменить текст программы.(рис. 2.4)



```

label:
    sub ecx, 1
    mov [N], ecx
    mov eax, [N]
    call iprintLF
    loop label

    call quit

```

Рис. 2.4: Изменение текста программы

В файле lab8-1.asm изменяем текст программы, добавив изменение значения регистра ecx в цикле.

5. Создать исполняемый файл и проверить его работу.(рис. 2.5)

```

[mdyanushkevich@fedora lab08]$ nasm -f elf lab8-1.asm
[mdyanushkevich@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[mdyanushkevich@fedora lab08]$ ./lab8-1
Введите N: 14
13
11
9
7
5
3
1

```

Рис. 2.5: Создание исполняемого файла, проверка программы

В командную строку вводим необходимые команды, чтобы создать исполняемый файл. Далее запускаем программу. Результатом работы программы стала

последовательность нечетных чисел от 14 до 1. Число проходов цикла не соответствует введенному числу.

6. Изменить текст программы.(рис. 2.6)

```
label:
    push ecx
    sub ecx,1
    mov [N],ecx
    mov eax,[N]
    call iprintLF
    pop ecx

loop label
```

Рис. 2.6: Изменение текста программы

В файле lab8-1.asm изменяем текст программы, добавив команды push и pop.

7. Создать исполняемый файл и проверить его работу.(рис. 2.7)

```

[mdyanushkevich@fedora lab08]$ gcc -c lab8-1.asm
^?[mdyanushkevich@fedora lab08]nasm -f elf lab8-1.asm.o
[mdyanushkevich@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[mdyanushkevich@fedora lab08]$ ./lab8-1
Введите N: 11
10
9
8
7
6
5
4
3
2
1
0

```

Рис. 2.7: Создание исполняемого файла, проверка программы

В командную строку вводим необходимые команды, чтобы создать исполняемый файл. Далее запускаем программу. Результатом работы программы стала последовательность чисел от 11 до 0. Число проходов цикла соответствует введенному числу.

8. Создать файл lab8-2.asm и ввести в него текст программы из листинга 8.2.(рис. 2.8)

```
%include 'in_out.asm'

SECTION .text
global _start

_start:
pop ecx

pop edx

sub ecx, 1

next:
cmp ecx, 0
jz _end

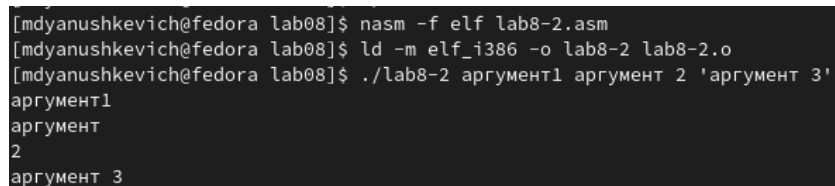
pop eax
call sprintf
loop next

_end:
call quit
```

Рис. 2.8: Ввод программы

Создаём файл lab8-2.asm и вводим в него текст программы листинга 8.2.

9. Создать исполняемый файл и проверить его работу.(рис. 2.9)



```
[mdyanushkevich@fedora lab08]$ nasm -f elf lab8-2.asm
[mdyanushkevich@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[mdyanushkevich@fedora lab08]$ ./lab8-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
```

Рис. 2.9: Создание исполняемого файла, проверка программы

В командную строку вводим необходимые команды, чтобы создать исполняемый файл. Далее запускаем программу. Результатом работы программы стали 4 аргумента.

10. Создать файл lab8-3.asm и ввести в него текст программы и листинга 8.3.(рис. 2.10)

```

#include 'in_out.asm'

SECTION .data
msg db "Результат: ",0

SECTION .text
global _start

_start:

pop ecx

pop edx

sub ecx,1

mov esi, 0

next:
cmp ecx,0h
jz _end

pop eax
call atoi
add esi,eax

loop next

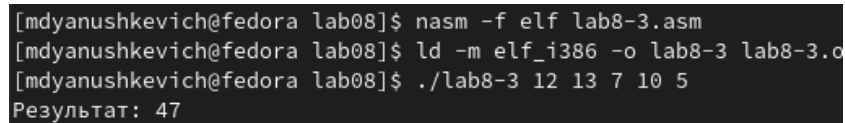
_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit

```

Рис. 2.10: Ввод программы

Создаём файл lab8-3.asm и в нем вводим текст программы листинга 8.3.

11. Создать исполняемый файл и проверить его работу.(рис. 2.11)



```
[mdyanushkevich@fedora lab08]$ nasm -f elf lab8-3.asm
[mdyanushkevich@fedora lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[mdyanushkevich@fedora lab08]$ ./lab8-3 12 13 7 10 5
Результат: 47
```

Рис. 2.11: Создание исполняемого файла, проверка программы

В командную строку вводим необходимые команды, чтобы создать исполняемый файл. Далее запускаем программу. Результатом работы программы стала сумма введенных аргументов: число 47.

12. Изменить текст программы для вычисления произведения аргументов.(рис. 2.12)

```

SECTION .data
msg db "Результат: ",0

SECTION .text
global _start

_start:

pop ecx

pop edx

sub ecx,1

mov esi,1

next:
cmp ecx,0h
jz _end

pop eax
call atoi
mul esi
mov esi, eax

loop next

_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit

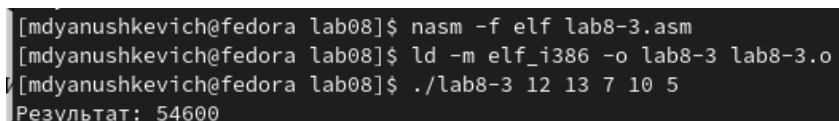
```

Рис. 2.12: Изменение программы



В файле lab8-3.asm изменяем текст программы для вычисления произведения аргументов.

13. Создать исполняемый файл и проверить его работу.(рис. 2.13)



```
[mdyanushkevich@fedora lab08]$ nasm -f elf lab8-3.asm
[mdyanushkevich@fedora lab08]$ ld -m elf_i386 -o lab8-3 lab8-3.o
[mdyanushkevich@fedora lab08]$ ./lab8-3 12 13 7 10 5
Результат: 54600
```

Рис. 2.13: Создание исполняемого файла, проверка программы

В командную строку вводим необходимые команды, чтобы создать исполняемый файл. Далее запускаем программу. Результатом работы программы стало произведение введенных аргументов: число 54600.

### 3 Задание для самостоятельной работы

1. Напишите программу, которая находит сумму значений функции  $f(x)$  для  $x = x_1, x_2 \dots x_n$ . (рис. 3.1)

```

#include 'in_out.asm'

SECTION .data
fx db "f(x)=2x+15", 0h
msg db 10,13 'Результат: ',0h

SECTION .text
global _start

_start:

pop ecx
pop edx
sub ecx,1
mov esi,0

next:
cmp ecx,0h
jz _end
pop eax
call atoi
mov ebx,2
mul ebx
add eax, 15
add esi, eax
|
loop next

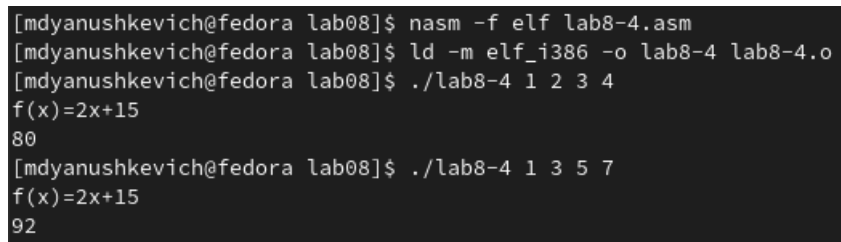
_end:
mov eax, fx
call sprint
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit

```

Рис. 3.1: Написание программы

Создаём файл lab8-4.asm, в нём пишем необходимую программу в соответствии с моим вариантом функции(мой вариант-1). Функция вида  $f(x)=2x+15$ .

2. Создать исполняемый файл и проверить его работу.(рис. 3.2)



```
[mdyanushkevich@fedora lab08]$ nasm -f elf lab8-4.asm
[mdyanushkevich@fedora lab08]$ ld -m elf_i386 -o lab8-4 lab8-4.o
[mdyanushkevich@fedora lab08]$ ./lab8-4 1 2 3 4
f(x)=2x+15
80
[mdyanushkevich@fedora lab08]$ ./lab8-4 1 3 5 7
f(x)=2x+15
92
```

Рис. 3.2: Создание исполняемого файла, проверка программы

В командную строку вводим необходимые команды, чтобы создать исполняемый файл. Далее запускаем программу. Результатом работы программы для  $x = 1\ 2\ 3\ 4$  стало число 80, а для  $x = 1\ 3\ 5\ 7$  стало число 92. Проверив полученные значения можно понять, что программа написана верно.

## 4 Выводы

Благодаря этой лабораторной работе я приобрел навыки по использованию циклов, а также научился работать с аргументами, введенными в командную строку.

## **Список литературы**