

# **ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №9**

**дисциплина: Архитектура компьютера**

**Шурыгин Илья Максимович**

# 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>Задание для самостоятельной работы:</b>	<b>11</b>
<b>5</b>	<b>Выводы</b>	<b>13</b>

## Список иллюстраций

3.1	Создали каталог lab09 в Midnight Commander и файл lab9-1.asm .	7
3.2	Вывод программы lab9-1.asm . . . . .	7
3.3	Некорректная работа программы . . . . .	8
3.4	Вывод программы lab9-1.asm . . . . .	8
3.5	Вывод программы lab9-1.asm . . . . .	9
3.6	Вывод результата суммы . . . . .	9
3.7	Вывод результата произведения . . . . .	9
3.8	Код программы, вычисляющей произведение аргументов . . . .	10
4.1	Результат работы программы . . . . .	11
4.2	Код программы . . . . .	12

## Список таблиц

# 1 Цель работы

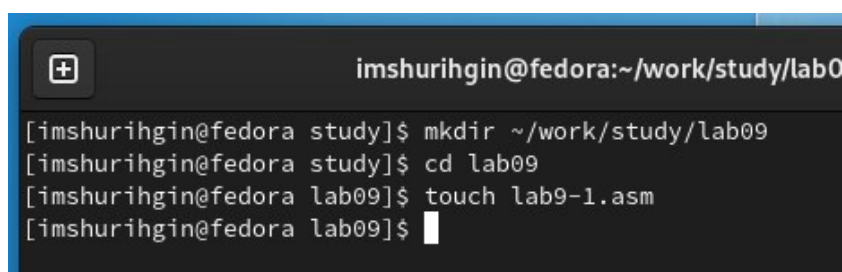
Приобрести навык написания программ с использованием циклов и обработкой аргументов командной строки.

## 2 Задание

Необходимо написать программы, которые вычисляют значение функции, зависящее от вводных данных.

### 3 Выполнение лабораторной работы

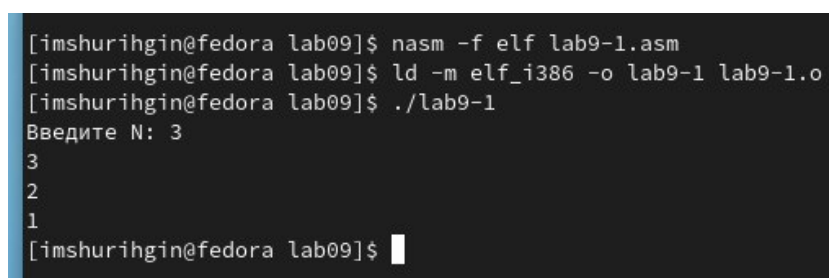
1. Откроем Midnight Commander и перейдем в каталог ~/work/study. Создадим каталог для программ лабораторной работы N9, перейдем в него и создадим файл lab9-1.asm.(рис. 3.1)



```
imshurihgin@fedora:~/work/study/lab0
[imshurihgin@fedora study]$ mkdir ~/work/study/lab09
[imshurihgin@fedora study]$ cd lab09
[imshurihgin@fedora lab09]$ touch lab9-1.asm
[imshurihgin@fedora lab09]$
```

Рис. 3.1: Создали каталог lab09 в Midnight Commander и файл lab9-1.asm

2. Запишем в файл lab9-1.asm текст программы из листинга 9.1, она будет выводить: 3, 2, 1. Изменим текст программы. Теперь программа работает некорректно, она выводит: 2, 0, а дальше уходит в бесконечный цикл.(рис. 3.2)(рис. 3.3)



```
[imshurihgin@fedora lab09]$ nasm -f elf lab9-1.asm
[imshurihgin@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[imshurihgin@fedora lab09]$ ./lab9-1
Введите N: 3
3
2
1
[imshurihgin@fedora lab09]$
```

Рис. 3.2: Вывод программы lab9-1.asm

```
imshurihgin@fedora:~/work/study/lab09 — /usr/bin/mc -P /var/.[imshurihgin@fedora lab09]$ nasm -f elf lab9-1.asm[imshurihgin@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o[imshurihgin@fedora lab09]$ ./lab9-1Введите N: 32042949672944294967292429496729042949672884294967286429496728442949672824294967280429496727842949672764294967274429496727242949672704294967268
```

Рис. 3.3: Некорректная работа программы

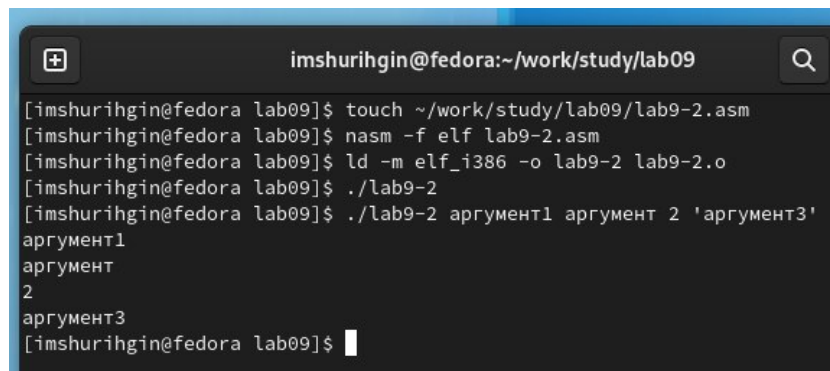
3. Изменим текст программы lab9-1.asm так, чтобы вывод программы был следующим: 2, 1, 0. В этом случае число проходов цикла соответствует значению N введенному с клавиатуры.(рис. 3.4)

```
imshurihgin@fedora:~/work/study/lab09[imshurihgin@fedora lab09]$ nasm -f elf lab9-1.asm[imshurihgin@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o[imshurihgin@fedora lab09]$ ./lab9-1Введите N: 3210[imshurihgin@fedora lab09]$
```

Рис. 3.4: Вывод программы lab9-1.asm

4. Создадим файл lab9-2.asm и запишем в него код из листинга 9.2(Программа, которая выводит на экран аргументы командной строки) Она выводит 4 аргумента.(рис. 3.5)

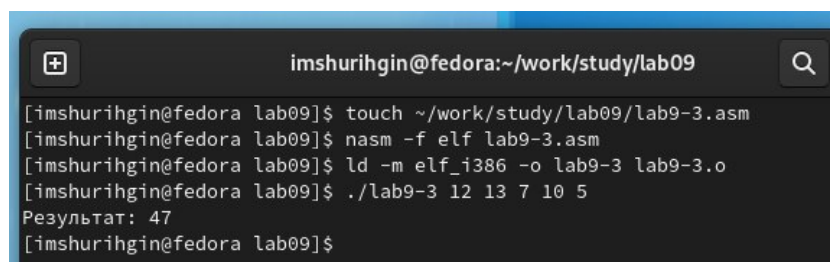




```
imshurihgin@fedora:~/work/study/lab09
[imshurihgin@fedora lab09]$ touch ~/work/study/lab09/lab9-2.asm
[imshurihgin@fedora lab09]$ nasm -f elf lab9-2.asm
[imshurihgin@fedora lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[imshurihgin@fedora lab09]$ ./lab9-2
[imshurihgin@fedora lab09]$ ./lab9-2 аргумент1 аргумент 2 'аргумент3'
аргумент1
аргумент
2
аргумент3
[imshurihgin@fedora lab09]$
```

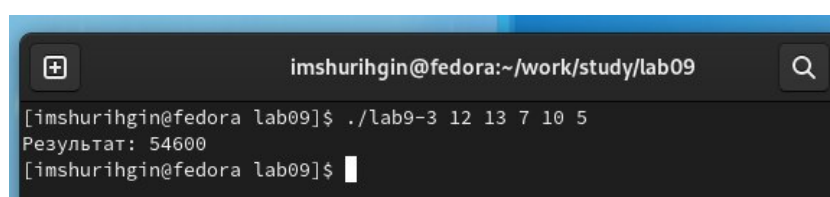
Рис. 3.5: Вывод программы lab9-1.asm

5. Создадим файл lab9-3.asm и введем в него текст программы из листинга 9.3(Программа, которая вычисляет суммы аргументов командной строки). Изменим текст программы из листинга 9.3 для вычисления произведения аргументов командной строки.(рис. 3.6)(рис. 3.7)



```
imshurihgin@fedora:~/work/study/lab09
[imshurihgin@fedora lab09]$ touch ~/work/study/lab09/lab9-3.asm
[imshurihgin@fedora lab09]$ nasm -f elf lab9-3.asm
[imshurihgin@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[imshurihgin@fedora lab09]$ ./lab9-3 12 13 7 10 5
Результат: 47
[imshurihgin@fedora lab09]$
```

Рис. 3.6: Вывод результата суммы



```
imshurihgin@fedora:~/work/study/lab09
[imshurihgin@fedora lab09]$ ./lab9-3 12 13 7 10 5
Результат: 54600
[imshurihgin@fedora lab09]$
```

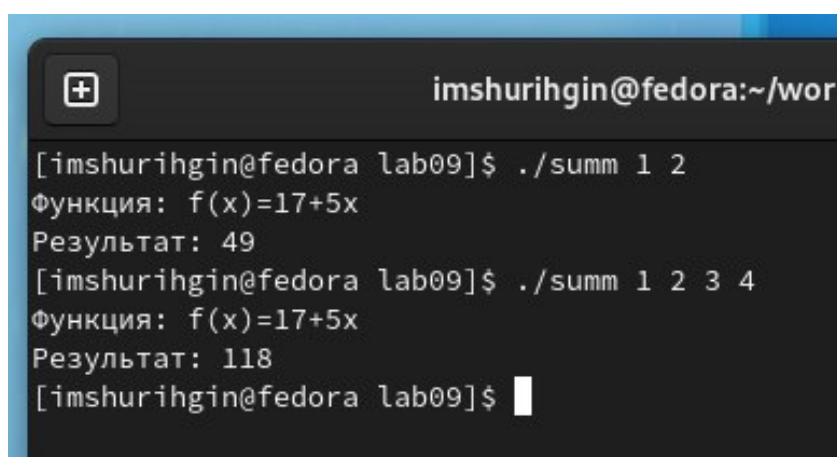
Рис. 3.7: Вывод результата произведения

```
next:
cmp ecx,0h ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку `_end`)
pop eax ; иначе извлекаем следующий аргумент из стека
call atoi ; преобразуем символ в число
mul esi
mov esi,eax
loop next ; переход к обработке следующего аргумента
_end:
```

Рис. 3.8: Код программы, вычисляющей произведение аргументов

## 4 Задание для самостоятельной работы:

1. Напишем программу, которая находит сумму значений функции  $f(x)$  для  $x = x_1, x_2, \dots, x_n$ , т.е. программа должна выводить значение  $f(x_1) + f(x_2) + \dots + f(x_n)$ .  
Данная функция:  $17+5x$ . (рис. 4.1)(рис. 4.2)



```
imshurihgin@fedora:~/worl
[imshurihgin@fedora lab09]$ ./summ 1 2
Функция: f(x)=17+5x
Результат: 49
[imshurihgin@fedora lab09]$ ./summ 1 2 3 4
Функция: f(x)=17+5x
Результат: 118
[imshurihgin@fedora lab09]$
```

Рис. 4.1: Результат работы программы

```
mc [imshurihgin@fedora]:~/w
summ.asm [----] 7 L: [ 1+ 2 3/ 30]
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
msg1 db "Функция: f(x)=17+5x",0
SECTION .text
global _start
_start:
pop ecx
pop edx
sub ecx,1
mov esi, 0
mov ebx,5
; ----- Вывод сообщения 'Введите function: '
mov eax,msg1
call sprintf
next:
cmp ecx,0h
jz _end
pop eax
call atoi
mul ebx
add eax, 17
add esi,eax
loop next
_end:
mov eax, msg
call sprintf
mov eax, esi
call iprintLF
call quit
```

Рис. 4.2: Код программы

## 5 Выводы

В данной лабораторной работе я научился писать простые программы на языке ассемблера NASM, а именно: создал программу которая вычисляет количество введенных аргументов и программу нахождения суммы результатов функций.