

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8**

*дисциплина: Архитектура компьютера*

Студент: Юсупова Амина Руслановна

Группа: НКАбд-06-25

**МОСКВА**

2025 г.

# Содержание

1	Цель работы .....	2
2	Задание .....	2
3	Теоретическое введение .....	2
4	Выполнение лабораторной работы.....	3
4.1	Реализация циклов в NASM.....	3
4.2	Обработка аргументов командной строки .....	6
4.3	Задание для самостоятельной работы.....	8
5	Выводы.....	11
6	Список литературы .....	11

## 1 Цель работы

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

## 2 Задание

1. Реализация циклом в NASM
2. Обработка аргументов командной строки
3. Самостоятельное написание программы по материалам

лабораторной работы

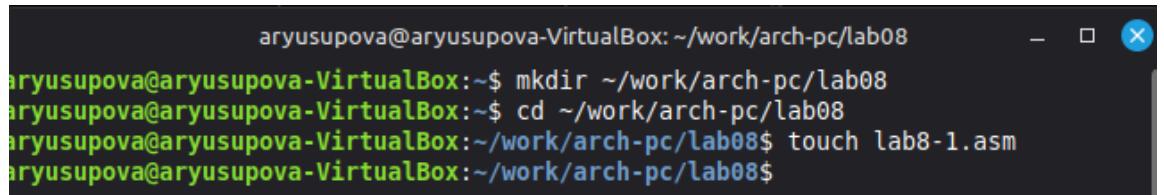
## 3 Теоретическое введение

Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нём выделяется память для локальных переменных и могут временно храниться значения регистров.

## 4 Выполнение лабораторной работы

### 4.1 Реализация циклов в NASM

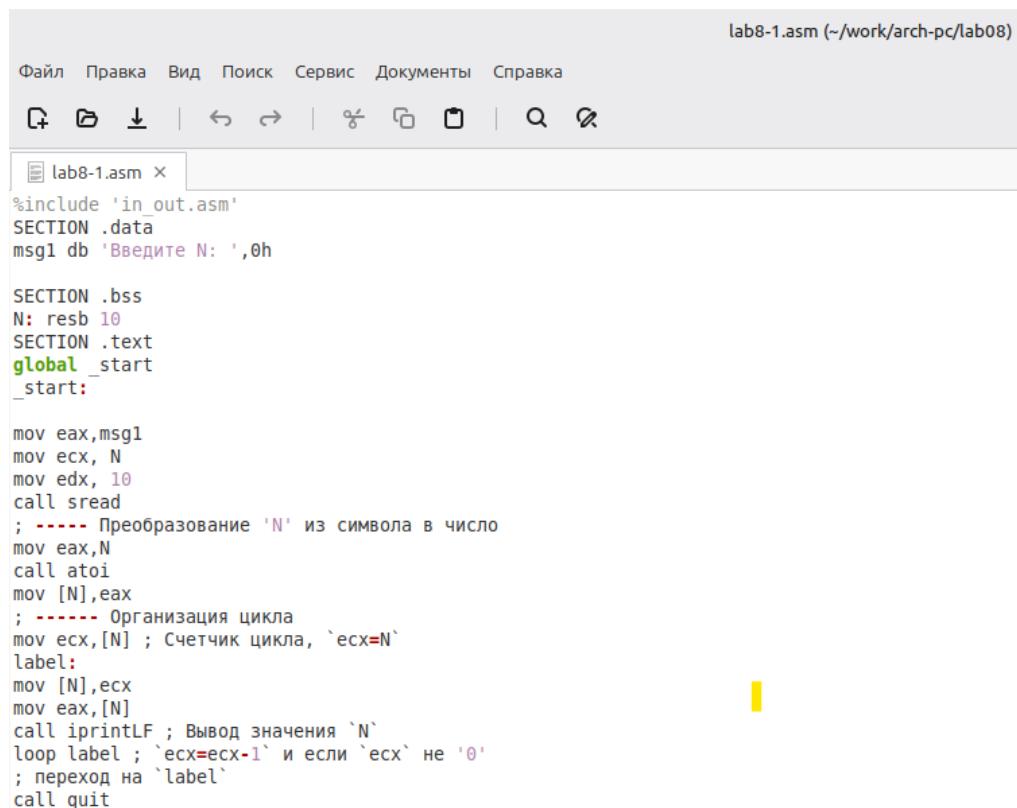
Создаю каталог для программ лабораторной работы № 8, перехожу в него и создаю файл lab8-1.asm (рис. 1).



```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08
aryusupova@aryusupova-VirtualBox:~$ mkdir ~/work/arch-pc/lab08
aryusupova@aryusupova-VirtualBox:~$ cd ~/work/arch-pc/lab08
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ touch lab8-1.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$
```

Рисунок 1: Создание каталога и файла в нём

Копирую текст программы из листинга 8.1 и вставляю в файл lab8-1.asm. (рис. 2).



```
lab8-1.asm (~/work/arch-pc/lab08)

Файл Правка Вид Поиск Сервис Документы Справка
lab8-1.asm x
%include 'in_out.asm'
SECTION .data
msg1 db 'Введите N: ',0h

SECTION .bss
N: resb 10
SECTION .text
global _start
_start:

    mov eax,msg1
    mov ecx, N
    mov edx, 10
    call sread
; ----- Преобразование 'N' из символа в число
    mov eax,N
    call atoi
    mov [N],eax
; ----- Организация цикла
    mov ecx,[N] ; Счетчик цикла, `ecx=N`
label:
    mov [N],ecx
    mov eax,[N]
    call iprintf ; Вывод значения `N`
    loop label ; `ecx=ecx-1` и если `ecx` не `0`
; переход на `label`
    call quit
```

Рисунок 2: Копирование программы из листинга 8.1

Запускаю программу lab8-1.asm. Она показывает работу циклов в NASM (рис. 3).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ mousepad lab8-1.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
```

### Рисунок 3: Запуск программы lab8-1.asm

Изменяю программу так, что в теле цикла я меняю значение регистра есх (рис. 4).

The screenshot shows a Microsoft Notepad window with the file name \*lab8-1.asm. The code is written in assembly language using AT&T syntax. It includes sections for data (.data), bss (.bss), and text (.text). The code starts at the \_start label, which initializes variable N to 10, reads a value from the user, converts it to an integer, and then prints it. A loop is used to repeat this process until the user enters quit.

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

call quit
```

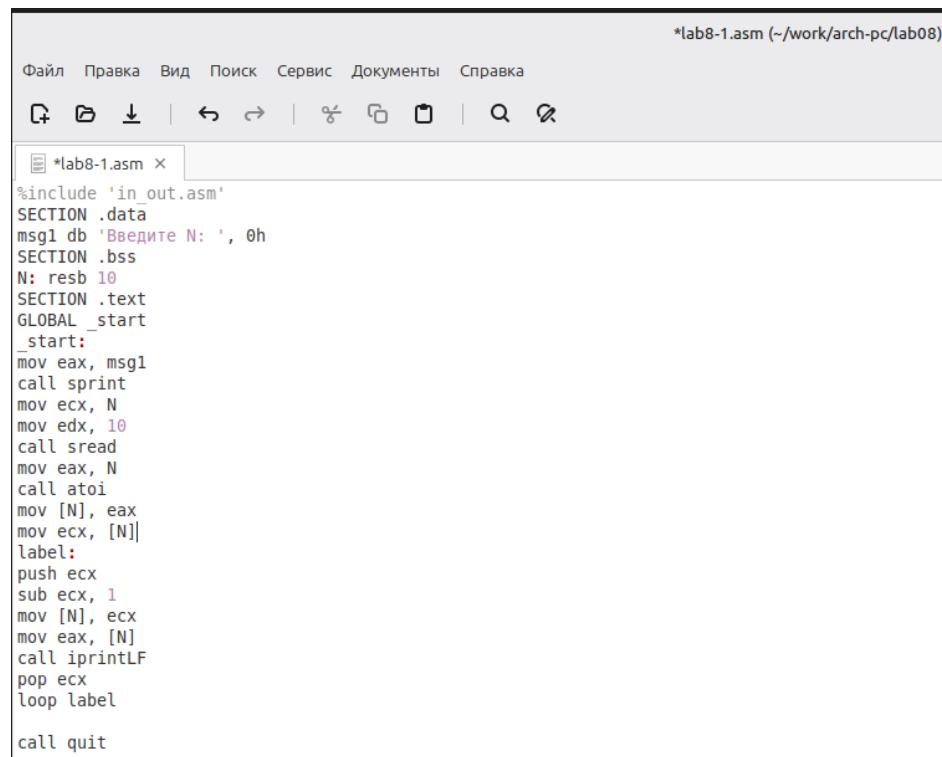
**Рисунок 4: Изменённая программа lab8-1.asm**

Далее компоную и запускаю изменённую программу lab8-1.asm (рис. 5).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
7
5
3
1
```

Рисунок 5: Запуск изменённой программы lab8-1.asm

Добавлю команды pop и push в программу lab8-1.asm (рис. 6).



The screenshot shows a Windows Notepad window titled "lab8-1.asm (~/work/arch-pc/lab08)". The menu bar includes "Файл", "Правка", "Вид", "Поиск", "Сервис", "Документы", and "Справка". Below the menu is a toolbar with icons for file operations. The main text area contains the following assembly code:

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

Рисунок 6: Изменённая программа lab8-1.asm (добавление pop и push)

Компоную и запускаю изменённую программу lab8-1.asm. Теперь количество итераций совпадает с введённым N, но произошло смещение вводимых чисел на -1 (рис. 7).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1.o lab8-1.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
```

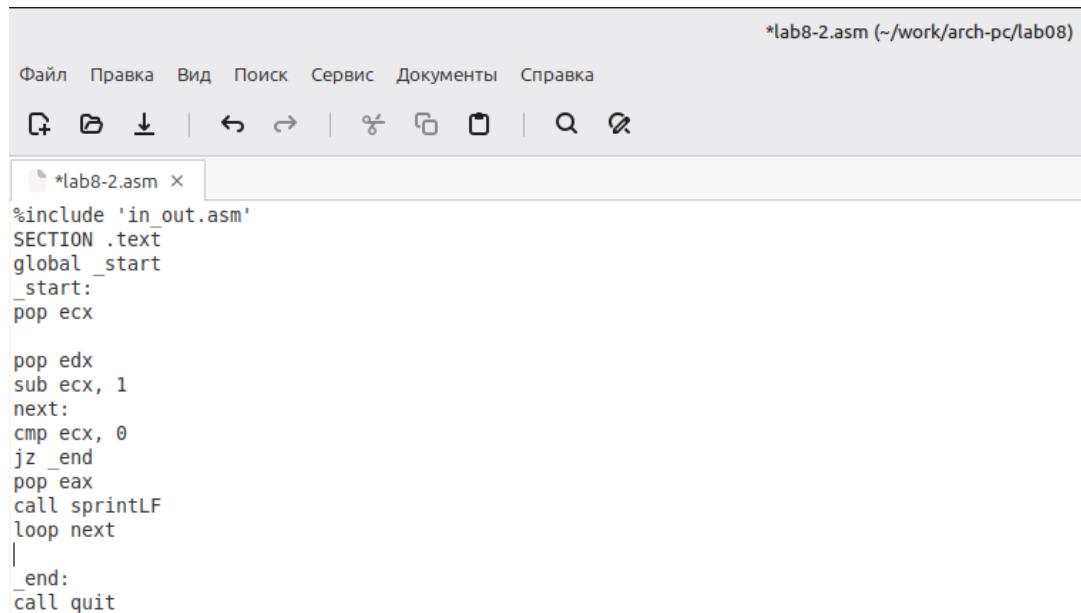
Рисунок 7: Запуск изменённой программы lab8-1.asm

#### 4.2 Обработка аргументов командной строки

Создаю новый файл lab8-2.asm для программы (рис. 8) и копирую в него код из листинга 8.2 (рис. 9).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ touch lab8-2.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ls
in_out.asm  lab8-1  lab8-1.asm  lab8-1.o  lab8-2.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ █
```

Рисунок 8: Создание файла lab8-2.asm



The screenshot shows a text editor window with the title bar "lab8-2.asm (~/work/arch-pc/lab08)". The menu bar includes "Файл", "Правка", "Вид", "Поиск", "Сервис", "Документы", and "Справка". Below the menu is a toolbar with icons for file operations. The main editor area contains the following assembly code:

```
%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 sprintLF
    loop next
|
_end:
    call quit
```

Рисунок 9: Программа lab8-2.asm

Далее компилирую и запускаю программу lab8-2.asm, указав аргументы. Программой было обработано то же количество аргументов, что и было введено (рис. 10).

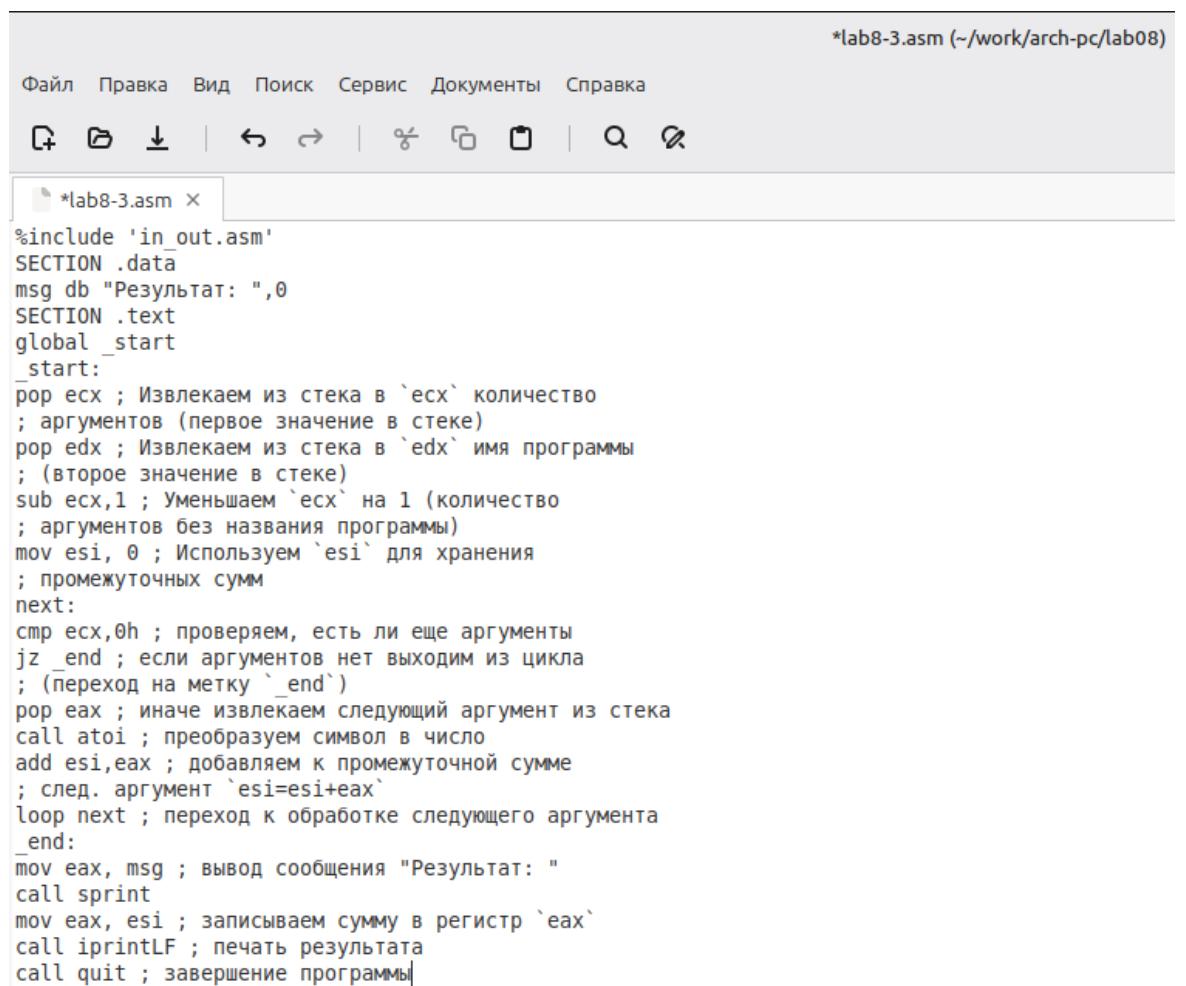
```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ./lab8-2 argument1 argument2 'argument3'
argument1
argument2
argument3
```

Рисунок 10: Запуск программы lab8-2.asm

Создаю новый файл lab8-3.asm (рис. 11) и копирую в него код из листинга 8.3 (рис. 12).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ touch lab8-3.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ls
in_out.asm  lab8-1  lab8-1.asm  lab8-1.o  lab8-2  lab8-2.asm  lab8-2.o  lab8-3.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$
```

Рисунок 11: Создание файла lab8-3.asm



```
*lab8-3.asm (~/work/arch-pc/lab08)

Файл Правка Вид Поиск Сервис Документы Справка
Q F D | ← → | % ⌂ ⌂ | Q ⌂

*lab8-3.asm ×
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx ; Извлекаем из стека в `ecx` количество
    ; аргументов (первое значение в стеке)
    pop edx ; Извлекаем из стека в `edx` имя программы
    ; (второе значение в стеке)
    sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
    ; аргументов без названия программы)
    mov esi, 0 ; Используем `esi` для хранения
    ; промежуточных сумм
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
    add esi,eax ; добавляем к промежуточной сумме
    ; след. аргумент `esi=esi+eax`
loop next ; переход к обработке следующего аргумента
_end:
    mov eax, msg ; вывод сообщения "Результат: "
    call sprint
    mov eax, esi ; записываем сумму в регистр `eax`
    call iprintLF ; печать результата
    call quit ; завершение программы
```

Рисунок 12: Копирование программы из листинга lab8-3.asm

Компоную программу и запускаю, указав в качестве аргументов некоторые числа, программа их складывает (рис. 13).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
```

**Рисунок 13:** Запуск программы lab8-3.asm

Изменяю поведение программы так, чтобы указанные аргументы она умножала, а не складывала (рис. 14).

The screenshot shows a Microsoft Notepad window with the following content:

```
*lab8-3.asm x
%include 'in_out.asm'

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
```

**Рисунок 14:** Изменение программы файла lab8-3.asm

Теперь запускаю изменённую программу lab8-3.asm. Она действительно умножает введённые числа (рис. 15).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 lll 1 6
Результат: 666
```

**Рисунок 15:** Запуск изменённой программы lab8-3.asm

### 4.3 Задание для самостоятельной работы

Создаю файл lab8-4.asm для выполнения самостоятельной работы (рис. 16).

```

aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ touch lab8-4.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ls
in_out.asm lab8-1.lab8-1.asm lab8-1.o lab8-2.lab8-2.asm lab8-2.o lab8-3.lab8-3.asm lab8-3.o lab8-4.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$
```

**Рисунок 16: Создание файла lab8-4.asm**

Пишу программу, которая будет находить сумму значений для функции  $f(x) = 10x - 4$ , которая совпадает с моим девятым вариантом (рис. 17).

```

*lab8-4.asm (~/work/arch-pc/lab08)

Файл Правка Вид Поиск Сервис Документы Справка

*lab8-4.asm ×

%include 'in_out.asm'

SECTION .data
msg_func db "Функция: f(x) = 10x - 4", 0
msg_result db "Результат: ", 0

SECTION .text
GLOBAL _start

_start:
    mov eax, msg_func
    call sprintLF

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

    next:
    cmp ecx, 0h
    jz _end
    pop eax
    call atoi

    mov ebx, 10
    mul ebx
    sub eax, 4

    add esi, eax
    loop next

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

```

**Рисунок 17: Написание программы для самостоятельной работы (lab8-4.asm)**

Код программы:

```
%include 'in_out.asm'

SECTION .data

msg_func db "Функция: f(x) = 10x - 4", 0
msg_result db "Результат: ", 0
```

```

SECTION .text
GLOBAL _start

_start:
    mov eax, msg_func
    call sprintLF
    pop ecx
    pop edx
    sub ecx, 1
    mov esi, 0
next:
    cmp ecx, 0h
    jz _end
    pop eax
    call atoi
    mov ebx, 10
    mul ebx
    sub eax, 4
    add esi, eax
    loop next

```

```

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

```

Компоную и запускаю программу lab8-4.asm, указав в качестве аргумента несколько чисел (рис. 18).

```

aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1 2 3 4
Функция: f(x) = 10x - 4
Результат: 84

```

Рисунок 18: Запуск программы lab8-4.asm

## 5 Выводы

В результате выполнения данной лабораторной работы я приобрела навыки написания программ с использованием циклов а также научилась обрабатывать аргументы командной строки.

## 6 Список литературы

1. <https://esystem.rudn.ru/mod/resource/view.php?id=1030556>
2. <https://esystem.rudn.ru/mod/resource/view.php?id=1030457>
3. Таненбаум Э. Архитектура компьютера. — 6-е изд. — СПб. : Питер, 2013.  
— 874 с. — (Классика Computer Science).