

# **Отчёта по лабораторной работе №8**

**Программирование цикла. Обработка аргументов командной строки.**

Семенов Сергей Алексеевич

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
3.1	Реализация циклов в NASM . . . . .	6
3.2	Обработка аргументов командной строки. . . . .	10
3.3	Задание для самостоятельной работы . . . . .	15
<b>4</b>	<b>Выводы</b>	<b>18</b>

## Список иллюстраций

3.1	Создаем каталог с помощью команды <code>mkdir</code> и файл с помощью команды <code>touch</code> . . . . .	6
3.2	Заполняем файл . . . . .	7
3.3	Запускаем файл и проверяем его работу . . . . .	8
3.4	Изменяем файл . . . . .	8
3.5	Запускаем файл и смотрим на его работу . . . . .	9
3.6	Редактируем файл . . . . .	9
3.7	Проверяем, сошелся ли наш вывод с данным в условии выводом .	10
3.8	Создаем файл командой <code>touch</code> . . . . .	10
3.9	Заполняем файл . . . . .	11
3.10	Смотрим на работу программ . . . . .	12
3.11	Создаем файл командой <code>touch</code> . . . . .	13
3.12	Заполняем файл . . . . .	14
3.13	Смотрим на работу программы . . . . .	14
3.14	Изменяем файл . . . . .	15
3.15	Проверяем работу файла(работает правильно) . . . . .	15
3.16	Создаем файл командой <code>touch</code> . . . . .	16
3.17	Пишем программу . . . . .	16
3.18	Смотрим на работу программы при $x_1=5$ $x_2=3$ $x_1=4$ (всё верно) . .	16
3.19	Смотрим на работу программы при $x_1=1$ $x_2=3$ $x_1=7$ (всё верно) . .	17

# 1 Цель работы

Изучить работу циклов и обработкой аргументов командной строки.

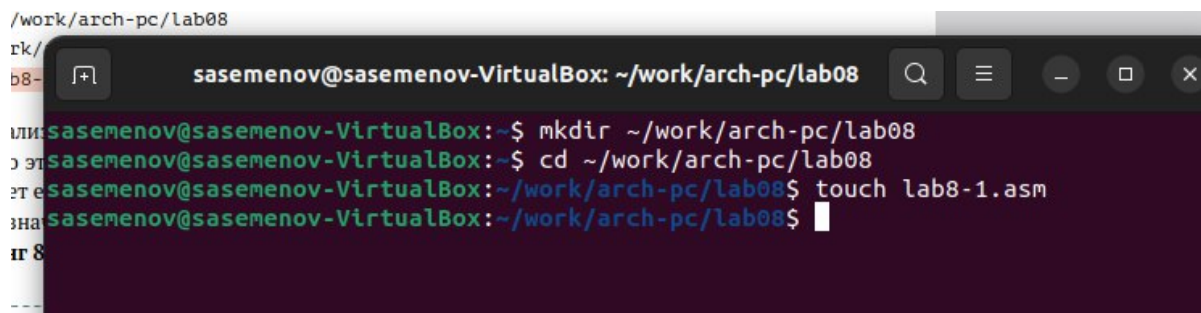
## 2 Задание

Написать программы с использованием циклов и обработкой аргументов командной строки.

## 3 Выполнение лабораторной работы

### 3.1 Реализация циклов в NASM

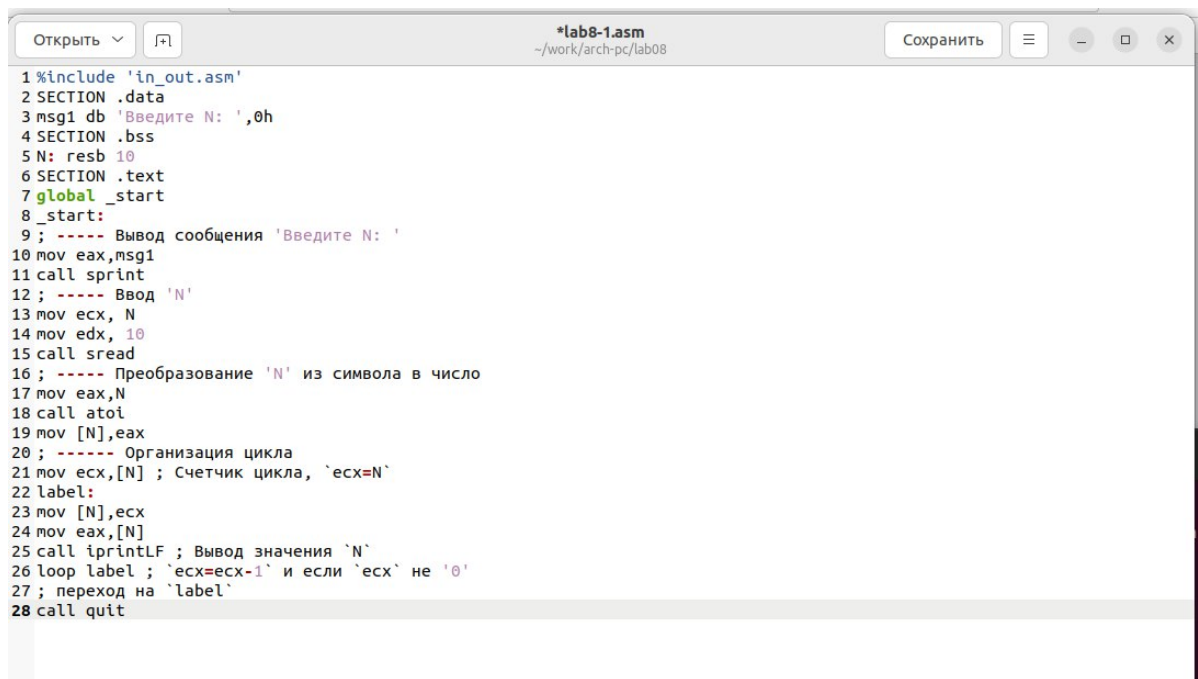
Создаем каталог для программ ЛБ8, и в нем создаем файл



```
/work/arch-pc/lab08
rk/
b8-
sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab08
или: sasemenov@sasemenov-VirtualBox:~$ mkdir ~/work/arch-pc/lab08
д эт: sasemenov@sasemenov-VirtualBox:~$ cd ~/work/arch-pc/lab08
эт е: sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ touch lab8-1.asm
зна: sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$
т 8
---
```

Рис. 3.1: Создаем каталог с помощью команды `mkdir` и файл с помощью команды `touch`

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 8.1



```
1 %include 'in_out.asm'
2 SECTION .data
3 msg1 db 'Введите N: ',0h
4 SECTION .bss
5 N: resb 10
6 SECTION .text
7 global _start
8 _start:
9 ; ----- Вывод сообщения 'Введите N: '
10 mov eax,msg1
11 call sprint
12 ; ----- Ввод 'N'
13 mov ecx, N
14 mov edx, 10
15 call sread
16 ; ----- Преобразование 'N' из символа в число
17 mov eax,N
18 call atoi
19 mov [N],eax
20 ; ----- Организация цикла
21 mov ecx,[N] ; Счетчик цикла, `ecx=N`
22 label:
23 mov [N],ecx
24 mov eax,[N]
25 call iprintLF ; Вывод значения `N`
26 loop label ; `ecx=ecx-1` и если `ecx` не `0`
27 ; переход на `label`
28 call quit
```

Рис. 3.2: Заполняем файл

Создаем исполняемый файл и запускаем его

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

Рис. 3.3: Запускаем файл и проверяем его работу

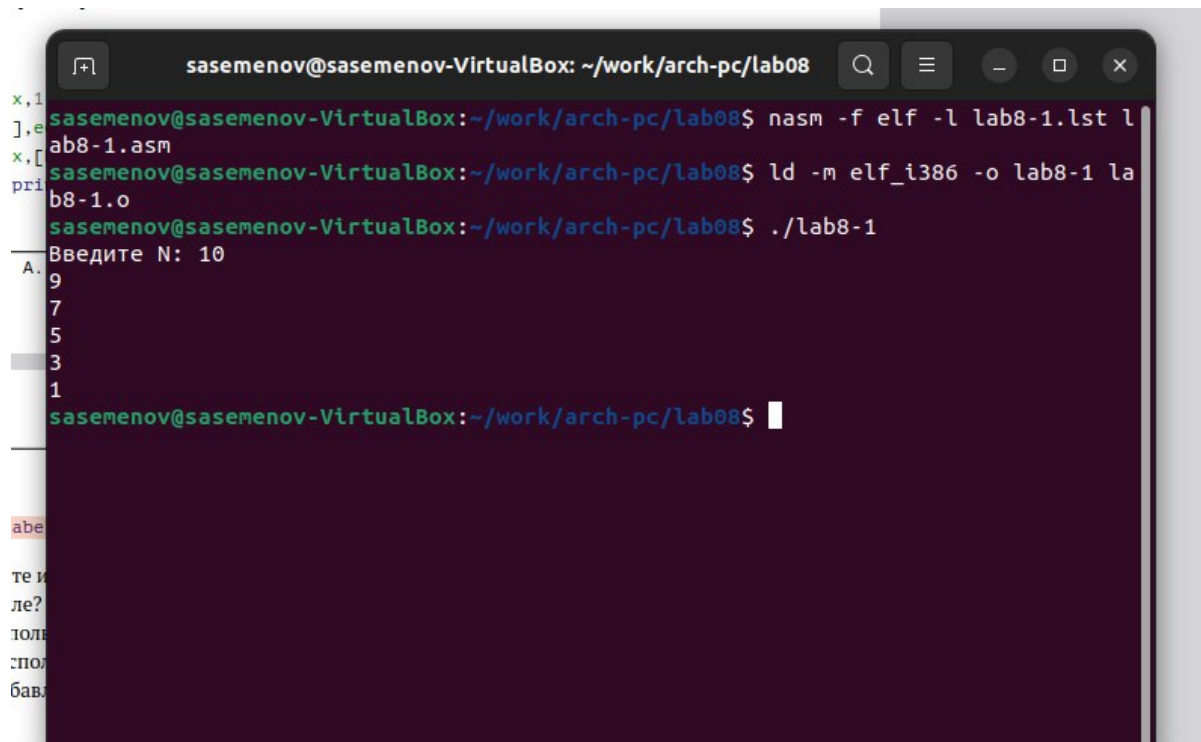
Снова открываем файл для редактирования и изменяем его, добавив изменение значения регистра в цикле

```
21 mov ecx,[N] ; счетчик цикла, ecx=
22 label:
23 sub ecx,1 ; `ecx=ecx-1`
24 mov [N],ecx
25 mov eax,[N]
26 call iprintLF
27 loop label
28 call quit
```

Рис. 3.4: Изменяем файл

Создаем исполняемый файл и запускаем его





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

Рис. 3.5: Запускаем файл и смотрим на его работу

Регистр `ecx` принимает значения 9,7,5,3,1(на вход подается число 10, в цикле `label` данный регистр уменьшается на 2 командой `sub` и `loop`).

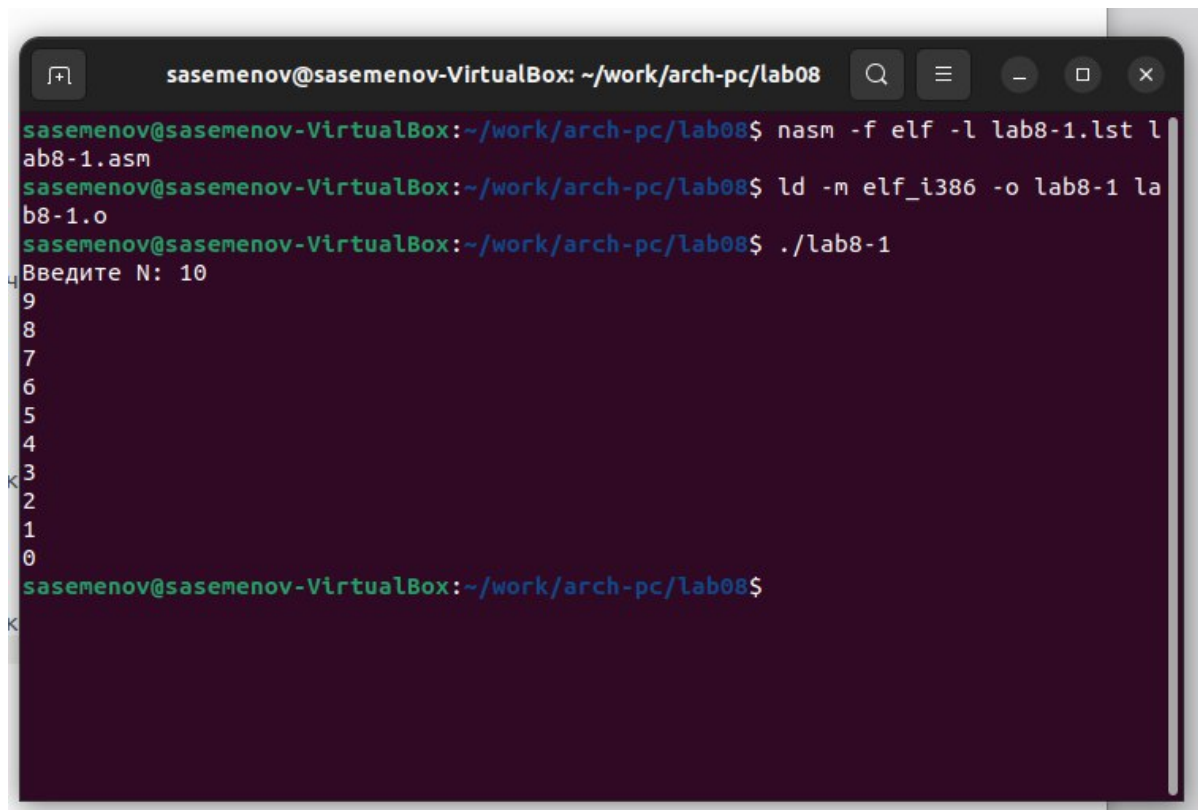
Число проходов цикла не соответствует числу `N`, так как уменьшается на 2.

Снова открываем файл для редактирования и изменяем его, чтобы все корректно работало

```
22 label:
23 push ecx ; добавление значения ecx в стек
24 sub ecx,1
25 mov [N],ecx
26 mov eax,[N]
27 call iprintLF
28 pop ecx ; извлечение значения ecx из стека
29 loop label
30 call quit
```

Рис. 3.6: Редактируем файл

Создаем исполняемый файл и запускаем его

A terminal window titled 'sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab08'. The user enters the command 'nasm -f elf -l lab8-1.lst lab8-1.asm', followed by 'ld -m elf\_i386 -o lab8-1 lab8-1.o', and then './lab8-1'. The program prompts 'Введите N: 10' and then prints a vertical list of numbers from 9 down to 0.

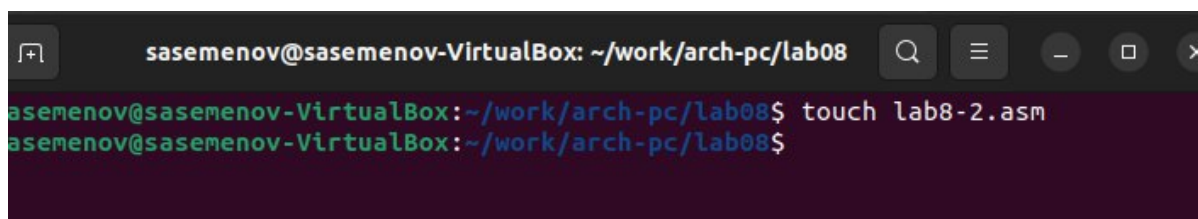
```
sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab08
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf -l lab8-1.lst lab8-1.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-1
Введите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

Рис. 3.7: Проверяем, сошелся ли наш вывод с данным в условии выводом

В данном случае число проходов цикла равна числу N.

## 3.2 Обработка аргументов командной строки.

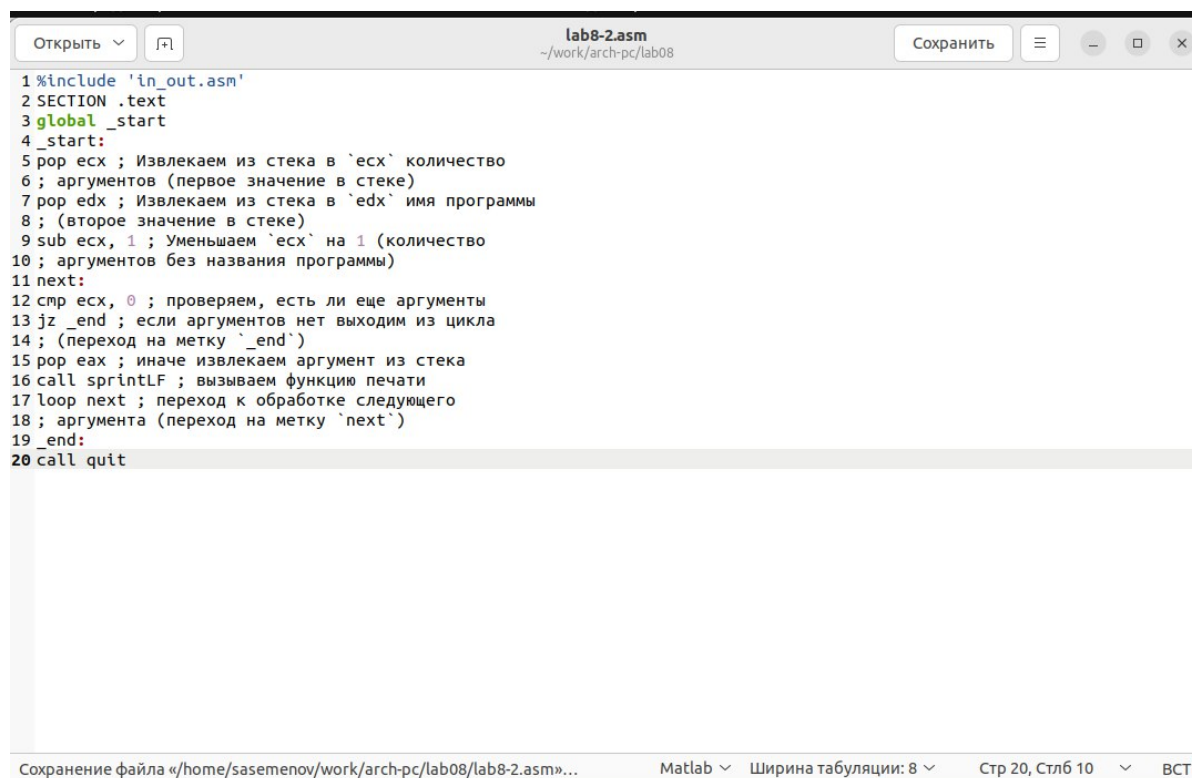
Создаем новый файл

A terminal window titled 'sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab08'. The user enters the command 'touch lab8-2.asm'.

```
sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab08
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ touch lab8-2.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$
```

Рис. 3.8: Создаем файл командой touch

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 8.2



```
1 %include 'in_out.asm'
2 SECTION .text
3 global _start
4 _start:
5 pop ecx ; Извлекаем из стека в `ecx` количество
6 ; аргументов (первое значение в стеке)
7 pop edx ; Извлекаем из стека в `edx` имя программы
8 ; (второе значение в стеке)
9 sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество
10 ; аргументов без названия программы)
11 next:
12 cmp ecx, 0 ; проверяем, есть ли еще аргументы
13 jz _end ; если аргументов нет выходим из цикла
14 ; (переход на метку `_end`)
15 pop eax ; иначе извлекаем аргумент из стека
16 call sprintf ; вызываем функцию печати
17 loop next ; переход к обработке следующего
18 ; аргумента (переход на метку `next`)
19 _end:
20 call quit
```

Сохранение файла «/home/sasemenov/work/arch-pc/lab08/lab8-2.asm»... Matlab Ширина табуляции: 8 Стр 20, Стлб 10 ВСТ

Рис. 3.9: Заполняем файл

Создаем исполняемый файл и проверяем его работу, указав аргументы

```
1z end ; если аргументов нет выходим из цикла
sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab08
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf -l lab8-2.lst lab8-2.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-2 1 2 '3'
1
2
3
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$
global start
```

Рис. 3.10: Смотрим на работу программ

Программой было обработано 3 аргумента.

Создаем новый файл lab8-3.asm

```
17     end      ; если аргументов нет выходим из цикла

sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab08
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ touch lab8-3.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$
```

Рис. 3.11: Создаем файл командой touch

Открываем файл и заполняем его в соответствии с листингом 8.3

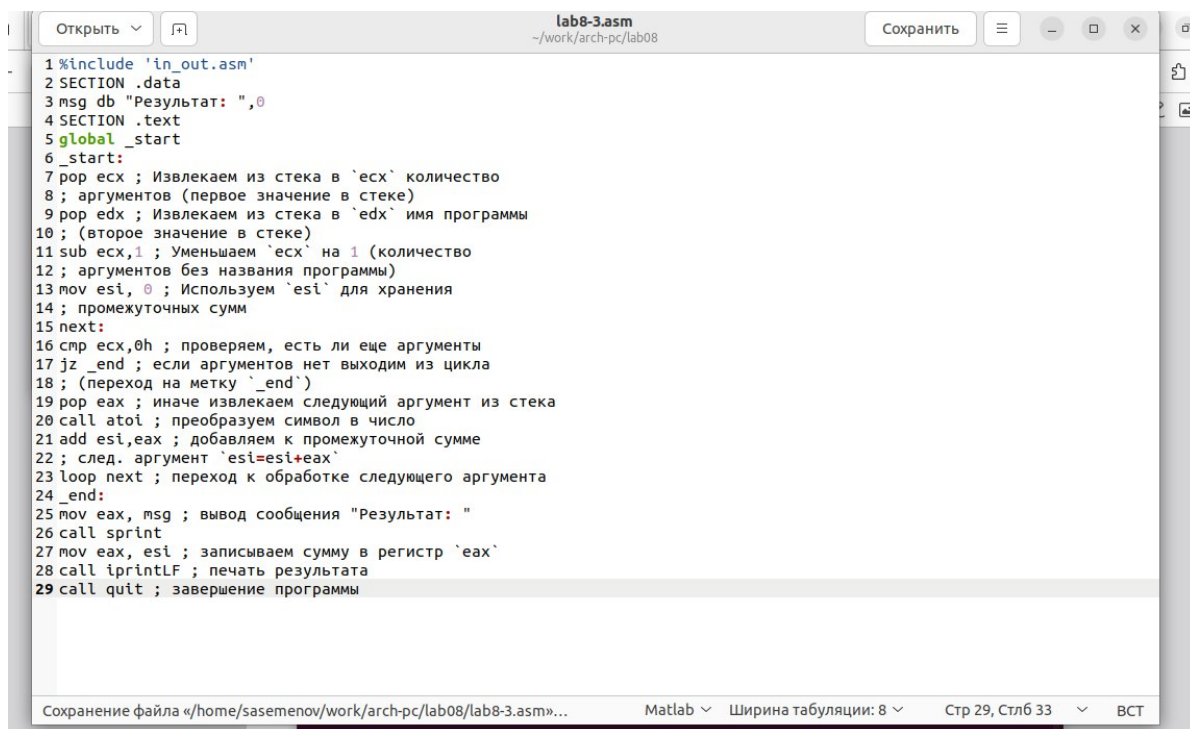


Рис. 3.12: Заполняем файл

Создаём исполняемый файл и запускаем его, указав аргументы

```

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

```

Рис. 3.13: Смотрим на работу программы

Снова открываем файл для редактирования и изменяем его, чтобы вычислялось произведение вводимых значений

```

15 next:
16 cmp ecx,0h
17 jz _end
18 pop eax
19 call atoi
20 mul esi
21 mov esi,eax
22 loop next
23 end

```

Рис. 3.14: Изменяем файл

Создаём исполняемый файл и запускаем его, указав аргументы

```

sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf -l lab8-3.lst lab8-3.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-3 5 3 4
Результат: 60
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$

```

Рис. 3.15: Проверяем работу файла(работает правильно)

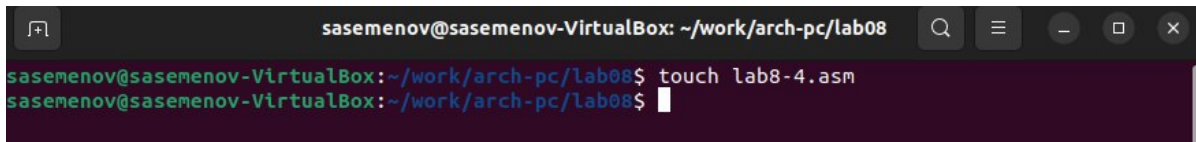
### 3.3 Задание для самостоятельной работы

#### ВАРИАНТ-3

1. Напишите программу, которая находит сумму значений функции  $f(x)$  для  $x = x_1, x_2, \dots, x_n$ , т.е. программа должна выводить значение  $f(x_1) + f(x_2) + \dots + f(x_n)$ . Значения  $x_i$  передаются как аргументы. Вид функции  $f(x)$  выбрать из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах  $x = x_1, x_2, \dots, x_n$ .

Создаем новый файл





```
sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab08
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ touch lab8-4.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$
```

Рис. 3.16: Создаем файл командой touch

Открываем его и пишем программу, которая выведет сумму значений, получившихся после решения выражения  $10x-5$

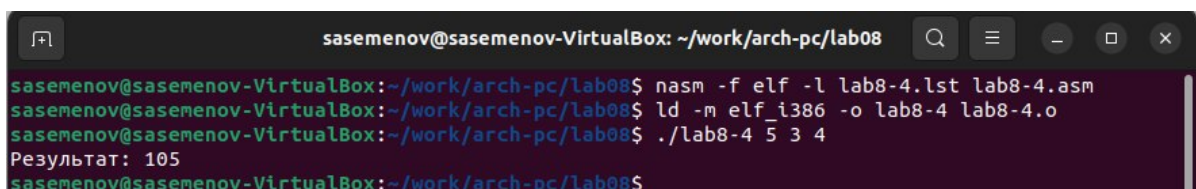


```
lab8-4.asm
~/work/arch-pc/lab08
Сохранить

1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 SECTION .bss
5 ans: RESB 80
6 SECTION .text
7 global _start
8 _start:
9 pop ecx
10 pop edx
11 sub ecx,1
12 mov esi,10
13 next:
14 cmp ecx,0h
15 jz _end
16 pop eax
17 call atoi
18 mul esi
19 sub eax,5
20 add [ans],eax
21 loop next
22 _end:
23 mov eax, msg
24 call sprint
25 mov eax,[ans]
26 call iprintLF
27 call quit
```

Рис. 3.17: Пишем программу

Транслируем файл и смотрим на работу программы



```
sasemenov@sasemenov-VirtualBox: ~/work/arch-pc/lab08
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf -l lab8-4.lst lab8-4.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 5 3 4
Результат: 105
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$
```

Рис. 3.18: Смотрим на работу программы при  $x_1=5$   $x_2=3$   $x_3=4$ (всё верно)

Транслируем файл и смотрим на работу программы



```
Результат: 95
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf -l lab8-4.lst lab8-4.asm
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1 3 7
Результат: 95
sasemenov@sasemenov-VirtualBox:~/work/arch-pc/lab08$
```

Рис. 3.19: Смотрим на работу программы при  $x1=1$   $x2=3$   $x3=7$  (всё верно)

## 4 Выводы

Мы научились решать программы с использованием циклов и обработкой аргументов командной строки.