

Шаблон отчёта по лабораторной работе №9

Мугари Абдеррахим , НКАбд-03-22

Содержание

1	Цель работы :	5
2	Выполнение лабораторной работы :	6
2.1	Реализация циклов в NASM :	6
2.2	Обработка аргументов командной строки :	9
2.3	Программа вычисления суммы аргументов командной строки : .	11
2.4	Выводы по результатам выполнения заданий :	15
3	Задание для самостоятельной работы :	16
3.1	Выводы по результатам выполнения заданий :	18
4	Выводы, согласованные с целью работы :	19

Список иллюстраций

2.1	Ресунок 1	6
2.2	Ресунок 2	7
2.3	Ресунок 3	8
2.4	Ресунок 4	8
2.5	Ресунок 5	9
2.6	Ресунок 6	10
2.7	Ресунок 7	11
2.8	Ресунок 8	12
2.9	Ресунок 9	13
2.10	Ресунок 10	14
2.11	Ресунок 11	15
3.1	Ресунок 12	17
3.2	Ресунок 13	18

Список таблиц

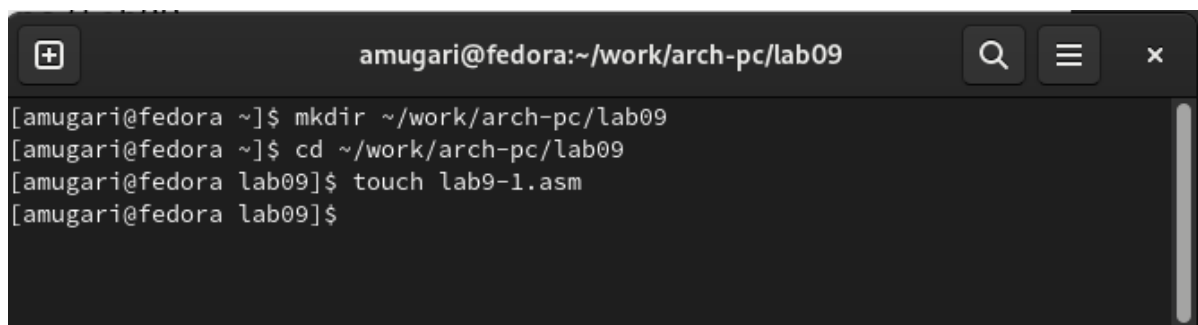
1 Цель работы :

В девятой лабораторной работе мы научимся писать программы с циклами и обработкой аргументов с помощью командной строки.

2 Выполнение лабораторной работы :

2.1 Реализация циклов в NASM :

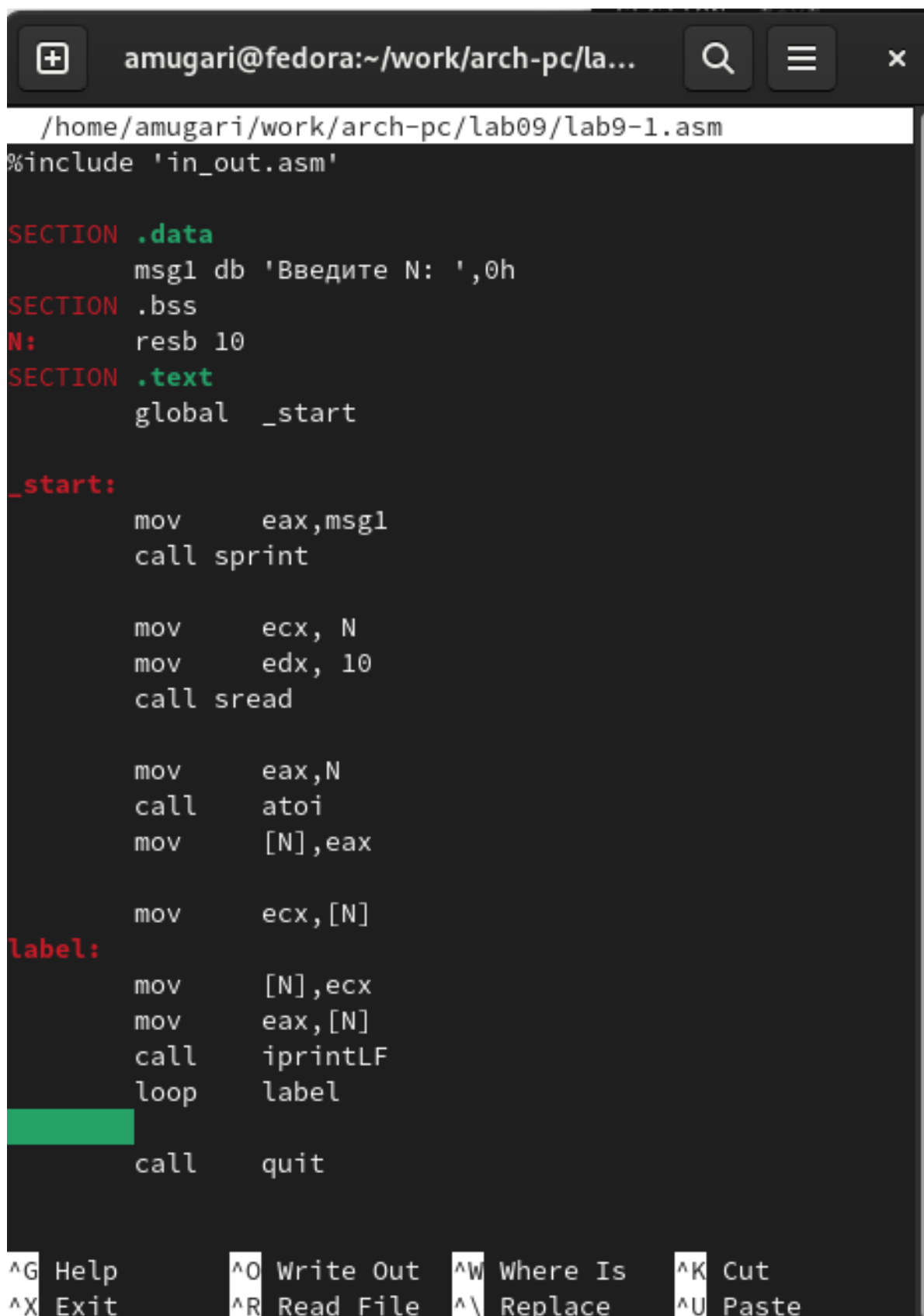
1. Здесь мы начали с создания каталога для программы лабораторной работы № 9, а затем переместились в девятый каталог лаборатории “~/work/arch-pc/lab09”, после чего мы создали файл “**lab9-1.asm**”. (рис. 2.1)



```
amugari@fedora:~/work/arch-pc/lab09
[amugari@fedora ~]$ mkdir ~/work/arch-pc/lab09
[amugari@fedora ~]$ cd ~/work/arch-pc/lab09
[amugari@fedora lab09]$ touch lab9-1.asm
[amugari@fedora lab09]$
```

Рис. 2.1: Ресунок 1

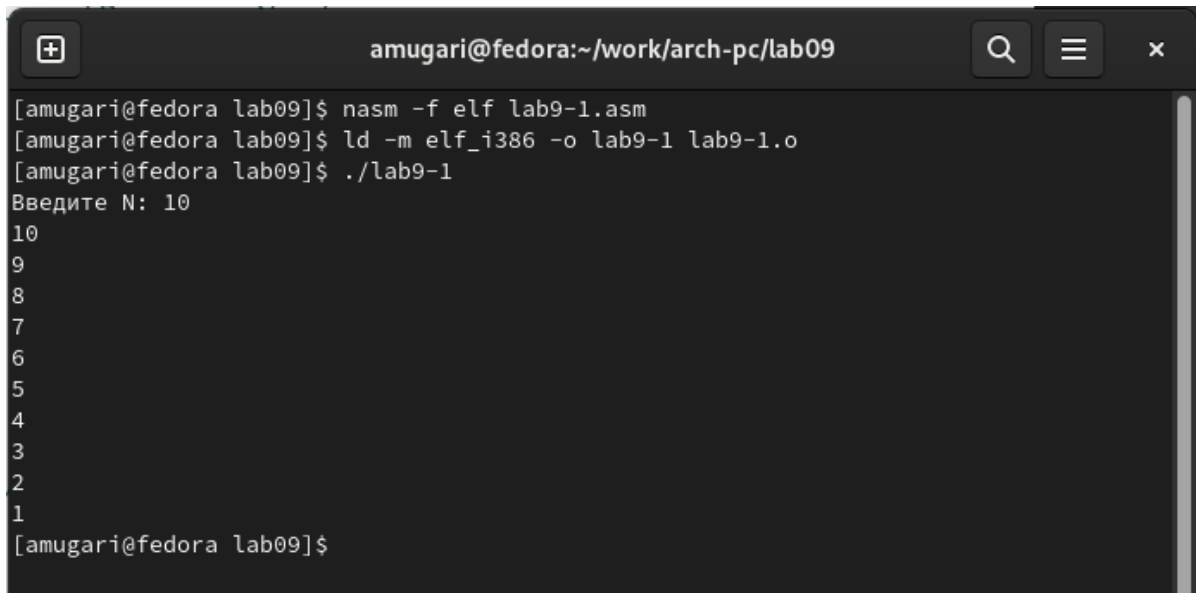
2. Затем мы заполнили код нашей программы в файле **lab9-1.asm**. (рис. 2.2)



```
amugari@fedora:~/work/arch-pc/la...  
/home/amugari/work/arch-pc/lab09/lab9-1.asm  
%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  
    loop    label  
    call    quit  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut  
^X Exit      ^R Read File  ^\ Replace    ^U Paste
```

Рис. 2.2: Ресунок 2

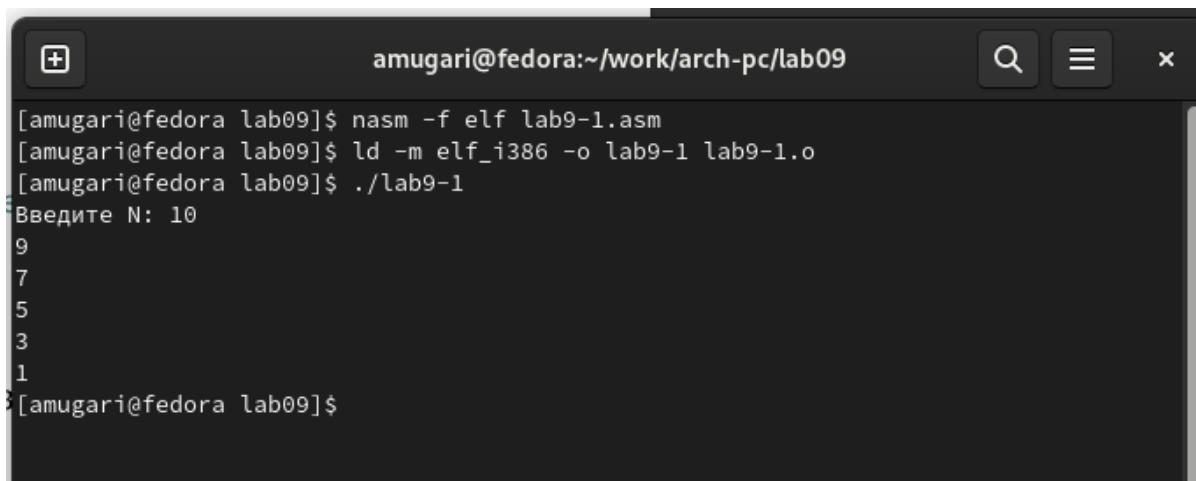
- После этого мы скомпилировали файл, создали исполняемый файл и проверили его работу.(рис. 2.3)



```
amugari@fedora:~/work/arch-pc/lab09
[amugari@fedora lab09]$ nasm -f elf lab9-1.asm
[amugari@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[amugari@fedora lab09]$ ./lab9-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
[amugari@fedora lab09]$
```

Рис. 2.3: Рисунок 3

- Мы внесли изменения в наш код, а затем создали исполняемый файл.(рис. 2.4)



```
amugari@fedora:~/work/arch-pc/lab09
[amugari@fedora lab09]$ nasm -f elf lab9-1.asm
[amugari@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[amugari@fedora lab09]$ ./lab9-1
Введите N: 10
9
7
5
3
1
[amugari@fedora lab09]$
```

Рис. 2.4: Рисунок 4

- Регистр **ecx** принимает пять значений, которые являются: **9,7,5,3,1**, мы

можем заметить, что количество циклов не соответствует числу, введенному пользователем.

- На этот раз мы использовали **стек**, и в конечном итоге количество циклов соответствует числу, которое было введено в начале.(рис. 2.5)



```
amugari@fedora:~/work/arch-pc/lab09
[amugari@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[amugari@fedora lab09]$ ./lab9-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
[amugari@fedora lab09]$
```

Рис. 2.5: Ресунок 5

2.2 Обработка аргументов командной строки :

1. На этом шаге мы создали файл **lab9-2.asm**, затем заполнили в нем наш код.(рис. 2.6)

```
amugari@fedora:~/work/arch-pc/...  
/home/amugari/work/arch-pc/lab09/lab9-2.asm  
%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
```

^G Help	^O Write Out	^W Where Is	^K Cut
^X Exit	^R Read File	^\ Replace	^U Paste

Рис. 2.6: Ресунок 6

2. После этого мы скомпилировали файл и создали исполняемый файл.(рис. 2.7)

A screenshot of a terminal window with a dark background. The title bar at the top shows the user 'amugari' on a 'fedora' machine, in the directory '~/work/arch-pc/lab09'. The terminal contains the following text:

```
[amugari@fedora lab09]$ nasm -f elf lab9-2.asm
[amugari@fedora lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[amugari@fedora lab09]$ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
[amugari@fedora lab09]$
```

Рис. 2.7: Ресунок 7

- и, как вы можете видеть, на этот раз при запуске программы мы добавили в команду три аргумента, и в этом случае были обработаны три аргумента.

2.3 Программа вычисления суммы аргументов командной строки :

1. Первым делом мы создали файл **lab9-3.asm**, затем заполнили кодом программы.(рис. 2.8)



```
GNU nano 6.0 /home/amugari/work/arch-pc/lab09/lab9-3.asm
%include 'in_out.asm'

SECTION .data
msg db "Результат: ",0

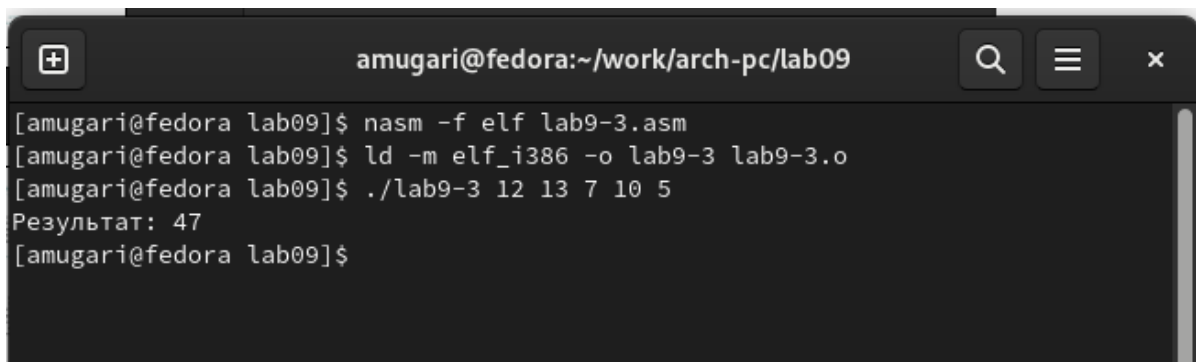
SECTION .text
global _start

_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

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Рис. 2.8: Ресунок 8

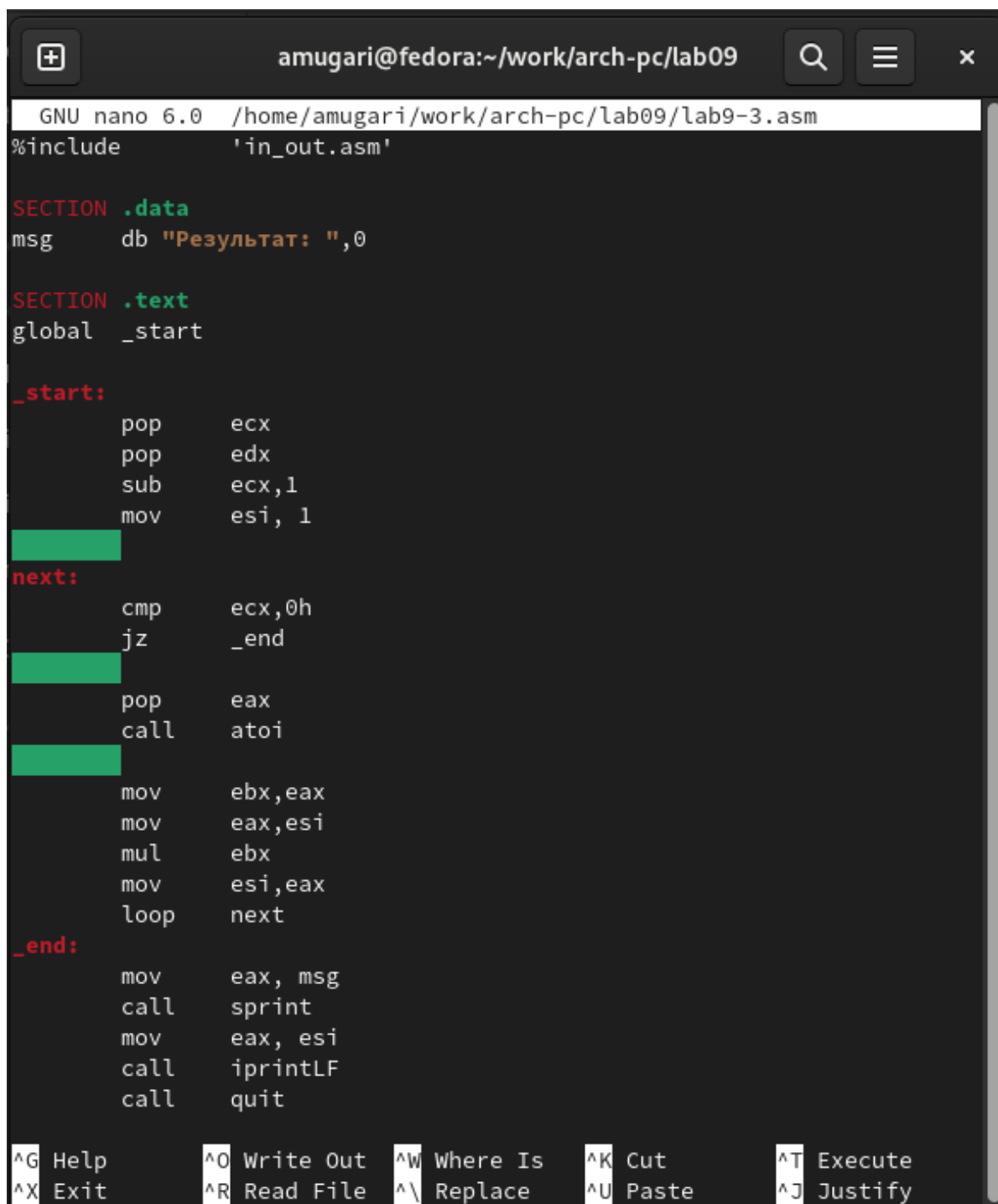
2. После этого мы скомпилировали файл, затем создали исполняемый файл, ввели нужное количество аргументов и запустили prgoram.(рис. 2.9)

A terminal window with a dark background. The title bar shows the user 'amugari@fedora' and the directory '~/work/arch-pc/lab09'. The terminal contains the following text:

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

Рис. 2.9: Ресунок 9

3. Затем мы изменили код, чтобы вычислить произведение аргументов командной строки.(рис. 2.10)



```
GNU nano 6.0 /home/amugari/work/arch-pc/lab09/lab9-3.asm
#include      'in_out.asm'

SECTION .data
msg          db "Результат: ",0

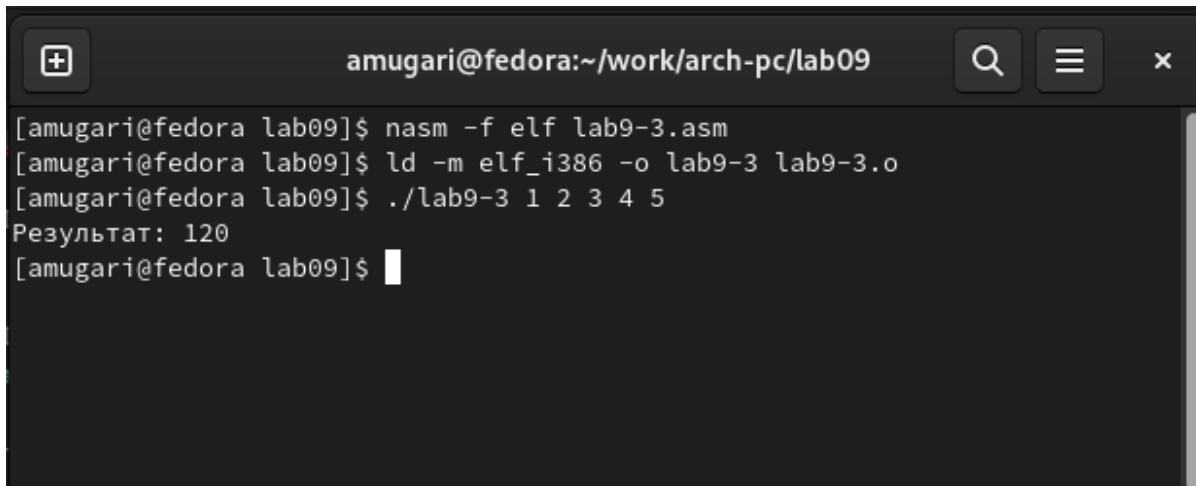
SECTION .text
global _start

_start:
    pop      ecx
    pop      edx
    sub      ecx,1
    mov      esi, 1
    next:
    cmp      ecx,0h
    jz        _end
    pop      eax
    call     atoi
    mov      ebx,eax
    mov      eax,esi
    mul      ebx
    mov      esi,eax
    loop     next
_end:
    mov      eax, msg
    call     sprint
    mov      eax, esi
    call     iprintLF
    call     quit

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

Рис. 2.10: Ресунок 10

- После этого е скомпилировал код и запустил исполняемый файл.(рис. 2.11)

A terminal window with a dark background. The title bar shows the user 'amugari@fedora' and the directory '~/work/arch-pc/lab09'. The terminal contains the following text:

```
[amugari@fedora lab09]$ nasm -f elf lab9-3.asm  
[amugari@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o  
[amugari@fedora lab09]$ ./lab9-3 1 2 3 4 5  
Результат: 120  
[amugari@fedora lab09]$
```

Рис. 2.11: Ресунок 11

2.4 Выводы по результатам выполнения заданий :

- В этой части работы мы узнали, как манипулировать циклами, как правильно использовать стек для написания программ

3 Задание для самостоятельной работы :

- В этой части мы должны были написать программу, которая находит сумму значений функции $f(x)$ для $x = x_1, x_2, \dots, x_n$
1. сначала мы создали наш файл **test.asm**, где будет находиться наш код, затем мы написали программу.(рис. 3.1)


```
GNU nano 6.0 /home/amugari/work/arch-pc/lab09/test.asm
%include 'in_out.asm'

SECTION .data
msg db "Результат: ",0
msg1 db "Функция: f(x)=12x-7",0

SECTION .text
global _start

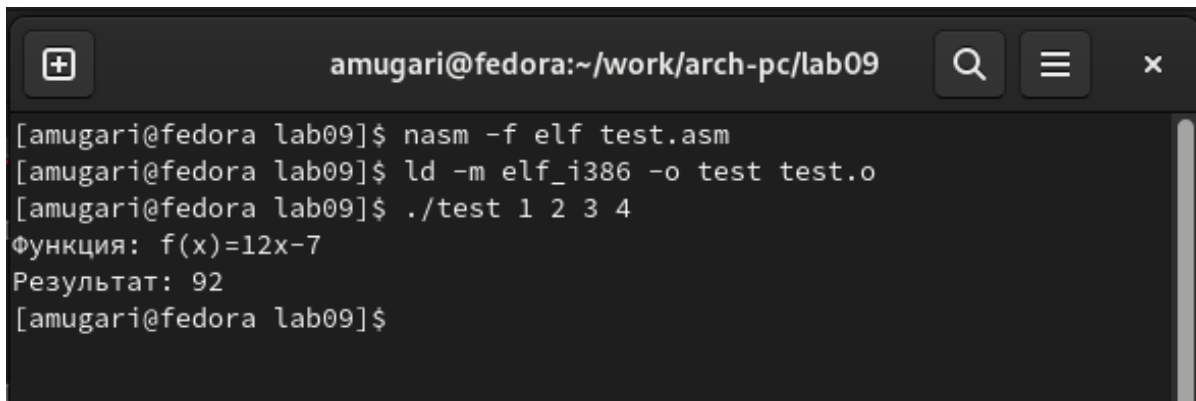
_start:
    pop    ecx
    pop    edx
    sub    ecx,1
    mov    esi, 0
next:
    cmp    ecx,0h
    jz     _end
    mov    ebx,12
    pop    eax
    call   atoi
    mul    ebx
    add    eax,-7
    add    esi,eax
    loop   next
_end:
    mov    eax, msg1
    call   sprintLF
    mov    eax, msg
    call   sprint
    mov    eax, esi
    call   iprintLF
    call   quit
```

Legend:

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify

Рис. 3.1: Ресунок 12

2. Затем мы протестировали нашу программу.(рис. 3.2)

A terminal window with a dark background. The title bar shows 'amugari@fedora:~/work/arch-pc/lab09'. The terminal contains the following text:

```
[amugari@fedora lab09]$ nasm -f elf test.asm
[amugari@fedora lab09]$ ld -m elf_i386 -o test test.o
[amugari@fedora lab09]$ ./test 1 2 3 4
Функция:  $f(x)=12x-7$ 
Результат: 92
[amugari@fedora lab09]$
```

Рис. 3.2: Рисунок 13

3.1 Выводы по результатам выполнения заданий :

В этой части мы узнали, как вычислить сложную математическую операцию, которая имеет функции, используя циклы и стек

4 Выводы, согласованные с целью работы :

- В девятой лабораторной работе мы узнали, как использовать циклы и стек в NASM.