

# **Отчёт по лабораторной работе №9**

**Программирование цикла. Обработка аргументов командной строки**

Михаил Александрович Мелкомуков

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Выполнение лабораторной работы</b>	<b>6</b>
2.1	Реализация циклов в NASM . . . . .	6
2.2	Обработка аргументов командной строки . . . . .	11
<b>3</b>	<b>Задание для самостоятельной работы</b>	<b>14</b>
<b>4</b>	<b>Выводы</b>	<b>16</b>

## Список иллюстраций

2.1	Создали каталог для программ лабораторной работы №9, перешли в него и создали файл lab9-1.asm . . . . .	6
2.2	Ввели в файл lab9-1.asm текст программы . . . . .	7
2.3	Создали исполняемый файл и проверили его работу . . . . .	8
2.4	Изменили текст программы, добавив изменение значения регистра ехх в цикле . . . . .	8
2.5	Создали исполняемый файл и проверили его работу . . . . .	9
2.6	Число N чётное . . . . .	9
2.7	Внесли изменения в текст программы, добавив команды push и pop . . . . .	10
2.8	Создали исполняемый файл и проверили его работу . . . . .	10
2.9	Создали файл lab9-2.asm и ввели в него текст программы . . . . .	11
2.10	Создали исполняемый файл и запустили его, указав аргументы . . . . .	11
2.11	Создали файл lab9-3.asm и ввели в него текст программы . . . . .	12
2.12	Создали исполняемый файл и запустили его, указав аргументы . . . . .	12
2.13	Изменили текст программы для вычисления произведения аргументов командной строки . . . . .	13
2.14	Создали исполняемый файл и проверили его работу . . . . .	13
3.1	Написали программу . . . . .	14
3.2	Создали исполняемый файл и проверили его работу на нескольких наборах . . . . .	15

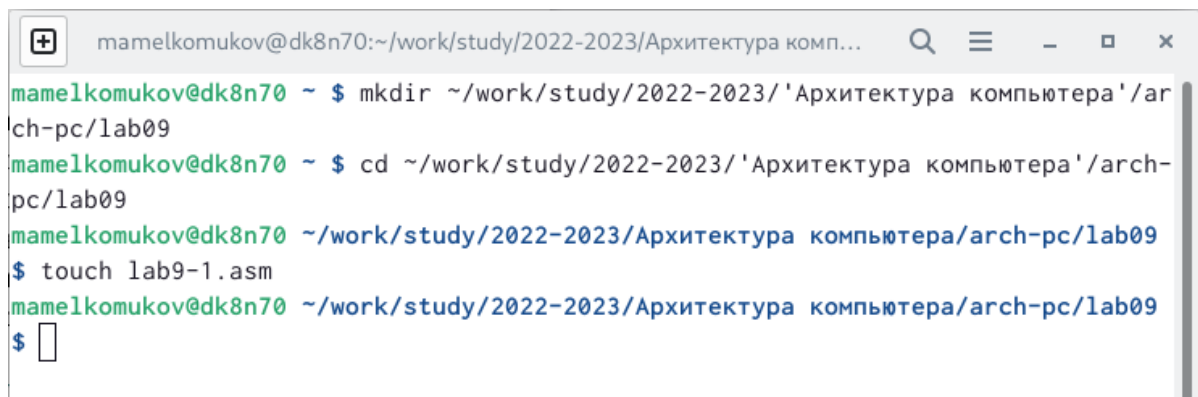
## **Список таблиц**

# 1 Цель работы

Цель работы приобрести навыки написания программ с использованием циклов и обработкой аргументов командной строки.

## 2 Выполнение лабораторной работы

### 2.1 Реализация циклов в NASM



```
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура комп...  
mamelkomukov@dk8n70 ~ $ mkdir ~/work/study/2022-2023/'Архитектура компьютера'/arch-pc/lab09  
mamelkomukov@dk8n70 ~ $ cd ~/work/study/2022-2023/'Архитектура компьютера'/arch-pc/lab09  
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09  
$ touch lab9-1.asm  
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09  
$
```

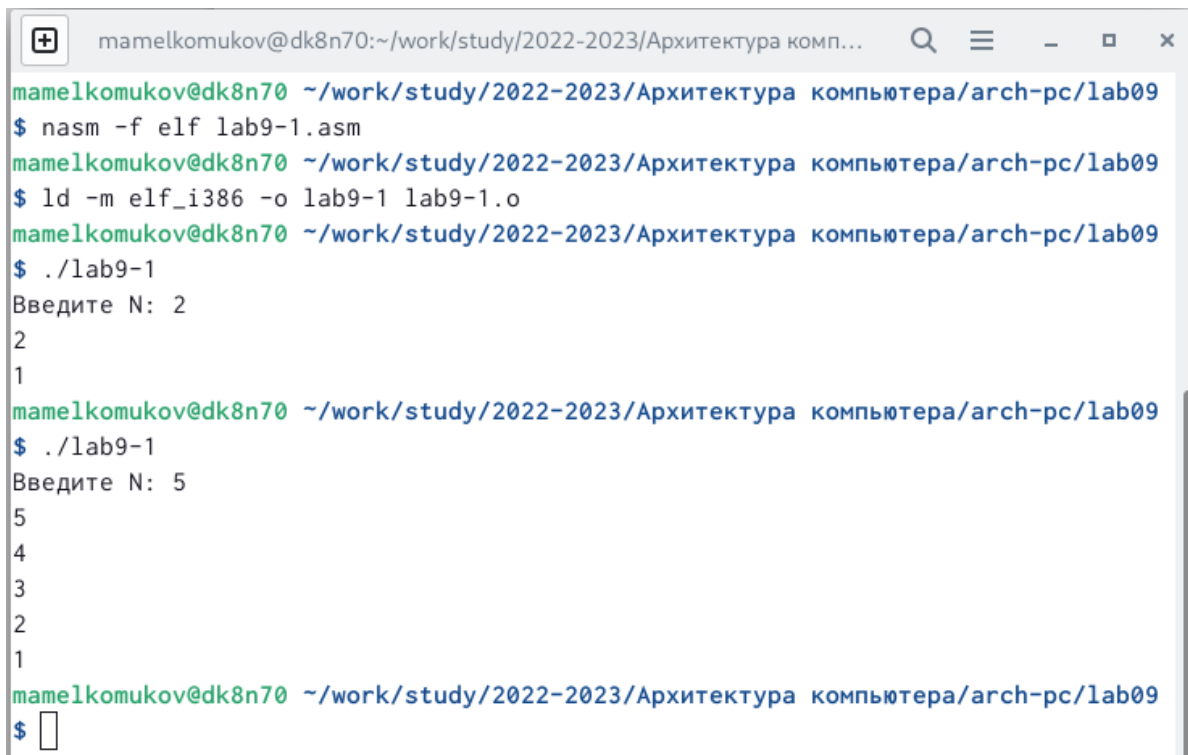
Рис. 2.1: Создали каталог для программ лабораторной работы №9, перешли в него и создали файл lab9-1.asm

```
mame1komukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ gedit lab9-1.asm

lab9-1.asm
~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
Сохранить

Открыть файл 'in_out.asm'
1 SECTION .data
2 msg1 db 'Введите N: ',0h
3
4 SECTION .bss
5 N: resb 10
6
7 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. 2.2: Ввели в файл lab9-1.asm текст программы



```
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ nasm -f elf lab9-1.asm
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ld -m elf_i386 -o lab9-1 lab9-1.o
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-1
Введите N: 2
2
1
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-1
Введите N: 5
5
4
3
2
1
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$
```

Рис. 2.3: Создали исполняемый файл и проверили его работу

```
22 label:
23 sub ecx,1
24 mov [N],ecx
25 mov eax,[N]
26 call iprintLF ; Вывод значения 'N'
27 loop label ; 'ecx=ecx-1' и если 'ecx' не '0'
```

Рис. 2.4: Изменили текст программы, добавив изменение значение регистра ecx в цикле



```
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-1
Введите N: 2
1
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-1
Введите N: 5
4
2
0
4294967294
4294967292
4294967290
4294967288
4294967286
4294967284
4294967282
4294967280
4294967278
4294967276
4294967274
4294967272
```

Рис. 2.5: Создали исполняемый файл и проверили его работу

```
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-1
Введите N: 6
5
3
1
```

Рис. 2.6: Число N чётное

Число проходов цикла не соответствует значению N, введенному с клавиатуры. Оно меньше его в два раза, если чётное. Если нечётное происходит нечто странное.

```

22 label:
23 push ecx
24 sub ecx,1
25 mov [N],ecx
26 mov eax,[N]
27 call iprintLF ; Вывод значения 'N'
28 pop ecx
29 loop label ; 'ecx=ecx-1' и если 'ecx' не '0'

```

Рис. 2.7: Внесли изменения в текст программы, добавив команды push и pop

```

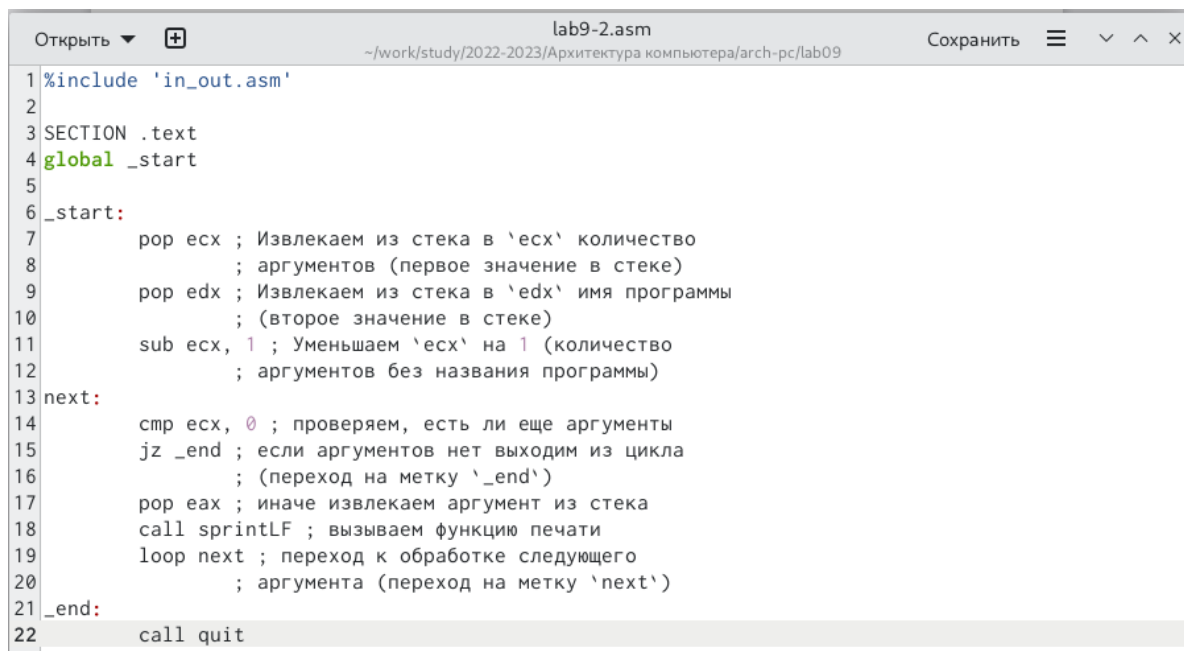
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура комп...
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ nasm -f elf lab9-1.asm
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ld -m elf_i386 -o lab9-1 lab9-1.o
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-1
Введите N: 2
1
0
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-1
Введите N: 5
4
3
2
1
0
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ 

```

Рис. 2.8: Создали исполняемый файл и проверили его работу

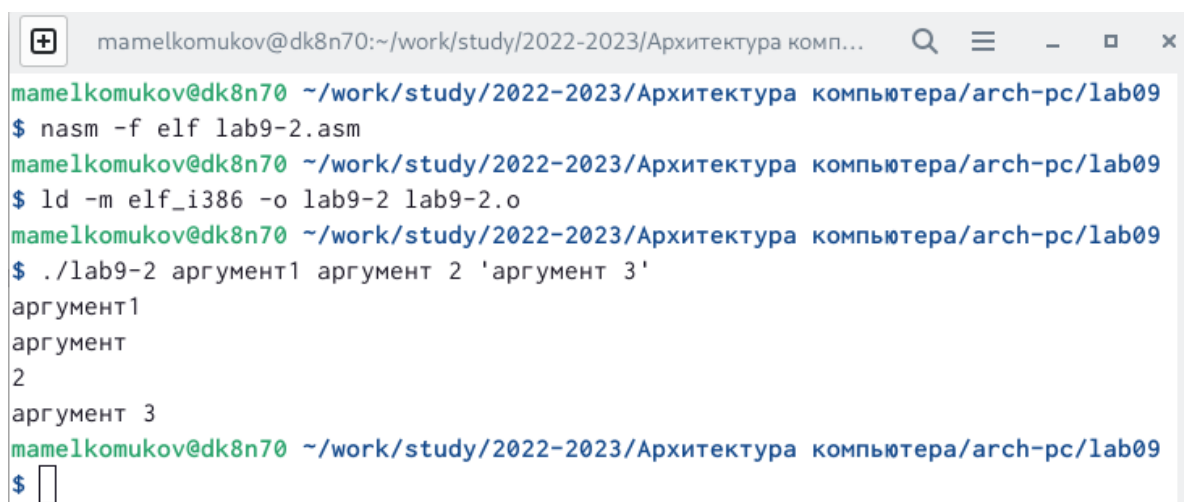
В данном случае, число проходов цикла соответствует значению N, введенному с клавиатуры.

## 2.2 Обработка аргументов командной строки



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

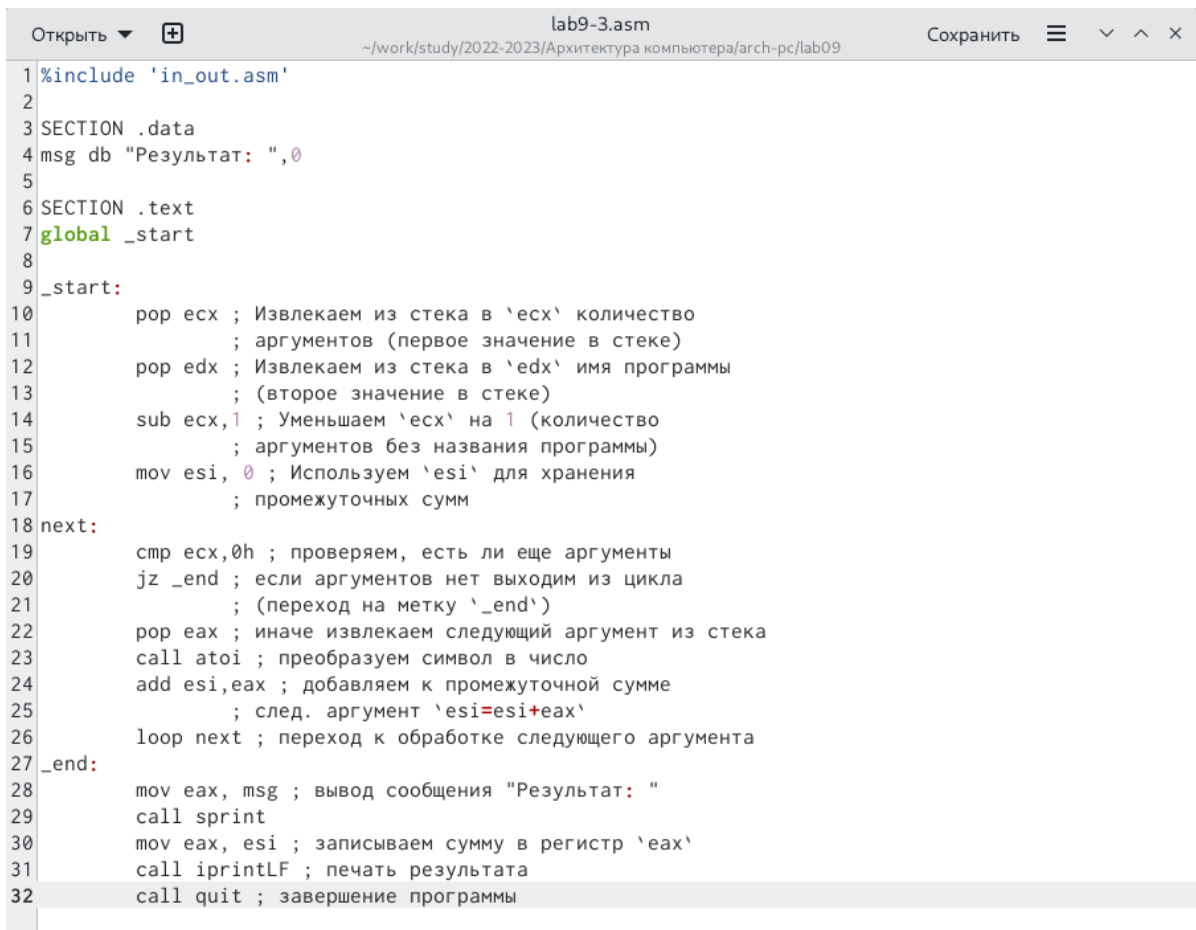
Рис. 2.9: Создали файл lab9-2.asm и ввели в него текст программы



```
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура комп...
$ nasm -f elf lab9-2.asm
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ld -m elf_i386 -o lab9-2 lab9-2.o
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент
2
аргумент 3
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$
```

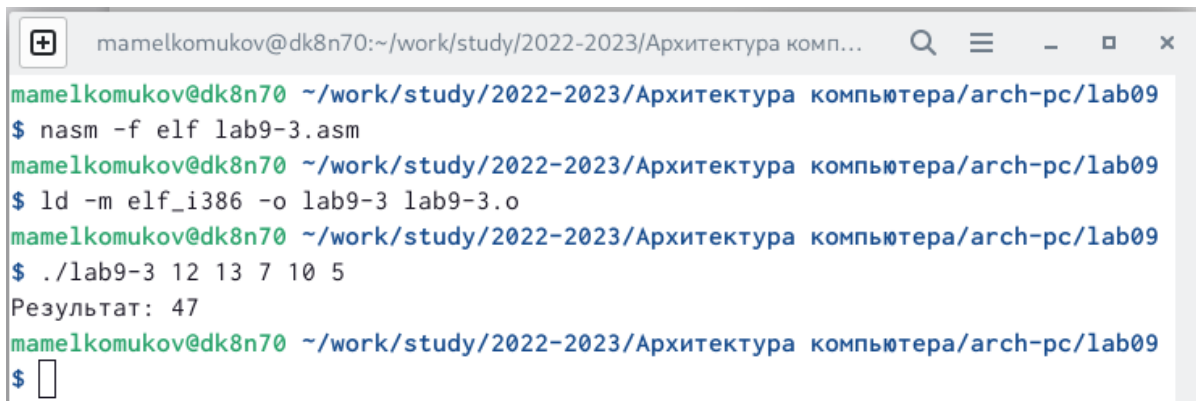
Рис. 2.10: Создали исполняемый файл и запустили его, указав аргументы

Программой было обработано 4 аргумента.



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg db "Результат: ",0
5
6 SECTION .text
7 global _start
8
9 _start:
10     pop ecx ; Извлекаем из стека в 'ecx' количество
11             ; аргументов (первое значение в стеке)
12     pop edx ; Извлекаем из стека в 'edx' имя программы
13             ; (второе значение в стеке)
14     sub ecx,1 ; Уменьшаем 'ecx' на 1 (количество
15             ; аргументов без названия программы)
16     mov esi,0 ; Используем 'esi' для хранения
17             ; промежуточных сумм
18 next:
19     cmp ecx,0h ; проверяем, есть ли еще аргументы
20     jz _end ; если аргументов нет выходим из цикла
21             ; (переход на метку '_end')
22     pop eax ; иначе извлекаем следующий аргумент из стека
23     call atoi ; преобразуем символ в число
24     add esi,eax ; добавляем к промежуточной сумме
25             ; след. аргумент 'esi=esi+eax'
26     loop next ; переход к обработке следующего аргумента
27 _end:
28     mov eax,msg ; вывод сообщения "Результат: "
29     call sprint
30     mov eax,esi ; записываем сумму в регистр 'eax'
31     call iprintLF ; печать результата
32     call quit ; завершение программы
```

Рис. 2.11: Создали файл lab9-3.asm и ввели в него текст программы



```
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура комп...
$ nasm -f elf lab9-3.asm
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ld -m elf_i386 -o lab9-3 lab9-3.o
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-3 12 13 7 10 5
Результат: 47
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$
```

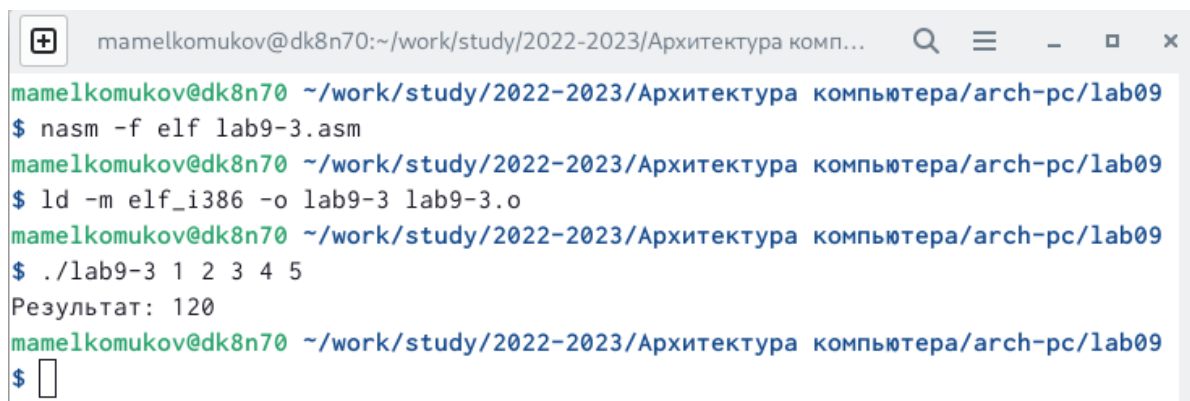
Рис. 2.12: Создали исполняемый файл и запустили его, указав аргументы

```

9 _start:
10     pop ecx ; Извлекаем из стека в 'ecx' количество
11           ; аргументов (первое значение в стеке)
12     pop edx ; Извлекаем из стека в 'edx' имя программы
13           ; (второе значение в стеке)
14     sub ecx,1 ; Уменьшаем 'ecx' на 1 (количество
15           ; аргументов без названия программы)
16     mov esi, 1
17 next:
18     cmp ecx,0h ; проверяем, есть ли еще аргументы
19     jz _end ; если аргументов нет выходим из цикла
20           ; (переход на метку '_end')
21     pop eax ; иначе извлекаем следующий аргумент из стека
22     call atoi ; преобразуем символ в число
23     mul esi ; умножаем
24           ; след. аргумент 'eax=eax*esi'
25     mov esi,eax
26     loop next ; переход к обработке следующего аргумента

```

Рис. 2.13: Изменили текст программы для вычисления произведения аргументов командной строки



```

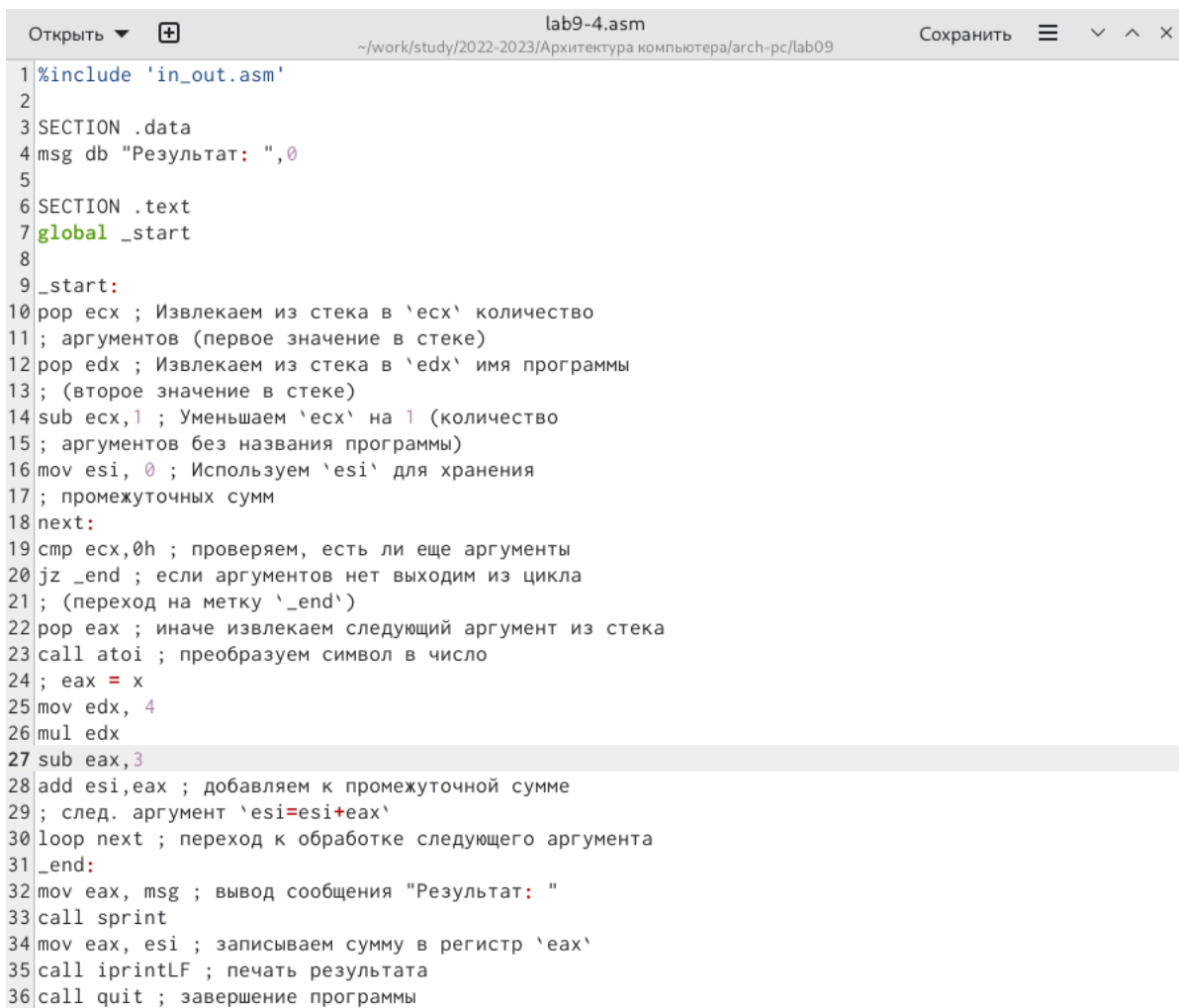
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура комп...
$ nasm -f elf lab9-3.asm
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ld -m elf_i386 -o lab9-3 lab9-3.o
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-3 1 2 3 4 5
Результат: 120
mamelkomukov@dk8n70:~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ 

```

Рис. 2.14: Создали исполняемый файл и проверили его работу

### 3 Задание для самостоятельной работы

Вариант, полученный при выполнении лабораторной работы №7 - шестой.



```
1 %include 'in_out.asm'
2
3 SECTION .data
4 msg db "Результат: ",0
5
6 SECTION .text
7 global _start
8
9 _start:
10 pop ecx ; Извлекаем из стека в 'ecx' количество
11 ; аргументов (первое значение в стеке)
12 pop edx ; Извлекаем из стека в 'edx' имя программы
13 ; (второе значение в стеке)
14 sub ecx,1 ; Уменьшаем 'ecx' на 1 (количество
15 ; аргументов без названия программы)
16 mov esi, 0 ; Используем 'esi' для хранения
17 ; промежуточных сумм
18 next:
19 cmp ecx,0h ; проверяем, есть ли еще аргументы
20 jz _end ; если аргументов нет выходим из цикла
21 ; (переход на метку '_end')
22 pop eax ; иначе извлекаем следующий аргумент из стека
23 call atoi ; преобразуем символ в число
24 ; eax = x
25 mov edx, 4
26 mul edx
27 sub eax,3
28 add esi,eax ; добавляем к промежуточной сумме
29 ; след. аргумент 'esi=esi+eax'
30 loop next ; переход к обработке следующего аргумента
31 _end:
32 mov eax, msg ; вывод сообщения "Результат: "
33 call sprint
34 mov eax, esi ; записываем сумму в регистр 'eax'
35 call iprintLF ; печать результата
36 call quit ; завершение программы
```

Рис. 3.1: Написали программу

```
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-4 1 2 3
Результат: 15
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ ./lab9-4 10 20 30
Результат: 231
mamelkomukov@dk8n70 ~/work/study/2022-2023/Архитектура компьютера/arch-pc/lab09
$ █
```

Рис. 3.2: Создали исполняемый файл и проверили его работу на нескольких наборах

## 4 Выводы

Приобрели навыки написания программ с использованием циклов и обработкой аргументов командной строки.