

# **Отчет по лабораторной работе номер 9**

Сафин Андрей Алексеевич

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Самостоятельная работа	16
5	Выводы	19

## Список иллюстраций

3.1	Текст lab9-1.asm . . . . .	8
3.2	Создание ~/work/arch-pc/lab09/lab9-1.asm . . . . .	8
3.3	Выполнение lab9-1.asm . . . . .	9
3.4	Измененный текст lab9-1.asm (1) . . . . .	9
3.5	Выполнение измененной программы lab9-1.asm (1) . . . . .	10
3.6	Измененный текст lab9-1.asm (2) . . . . .	11
3.7	Выполнение измененной программы lab9-1.asm (2) . . . . .	11
3.8	Текст lab9-2.asm . . . . .	12
3.9	Выполнение программы lab9-2.asm . . . . .	12
3.10	Текст lab9-3.asm . . . . .	13
3.11	Работа программы lab9-3.asm . . . . .	14
3.12	Измененный текст lab9-3.asm . . . . .	14
3.13	Работа измененной программы lab9-3.asm . . . . .	15
4.1	Текст sr.asm . . . . .	17
4.2	Текст sr.asm . . . . .	18

## Список таблиц

# 1 Цель работы

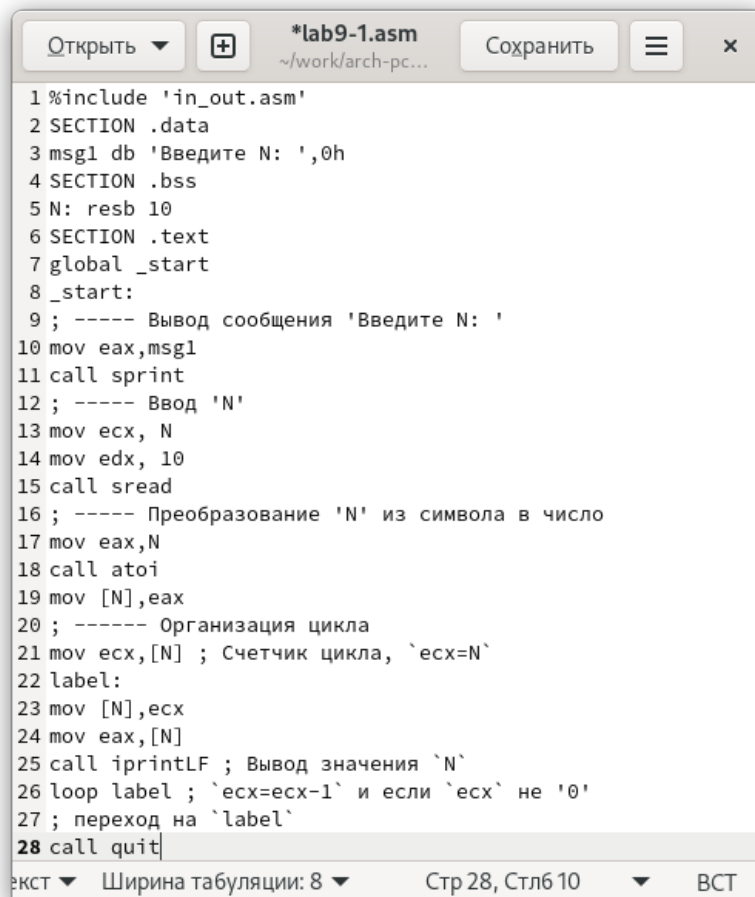
Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

## 2 Задание

Написать ряд программ, содержащих инструкции циклов и работающих со стеком.

### 3 Выполнение лабораторной работы

Создан файл lab9-1.asm в каталоге ~/work/arch-pc/lab09 (рис. 3.2). В него записана программа из листинга 9.1 (рис. 3.1), выводящая значения есх циклично с учетом их изменения при применении команды loop (рис. 3.3). В текст внесено изменение: в цикл добавлено вычитание единицы из значения в есх (рис. 3.4). В результате программа выполняется некорректно (рис. 3.5).

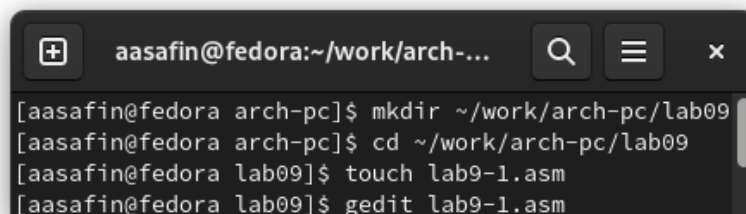


The screenshot shows a text editor window titled '\*lab9-1.asm' with a file path '~/.work/arch-pc...'. The editor contains 28 lines of assembly code. The code includes directives for sections (.data, .bss, .text), variable declarations, and instructions for printing a message, reading input, and processing it in a loop. Comments in Russian describe the steps: outputting the message, inputting 'N', converting 'N' to a number, and organizing the loop. The code ends with a call to 'quit'.

```
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
```

At the bottom of the window, status information is displayed: 'екст', 'Ширина табуляции: 8', 'Стр 28, Стлб 10', and 'ВСТ'.

Рис. 3.1: Текст lab9-1.asm

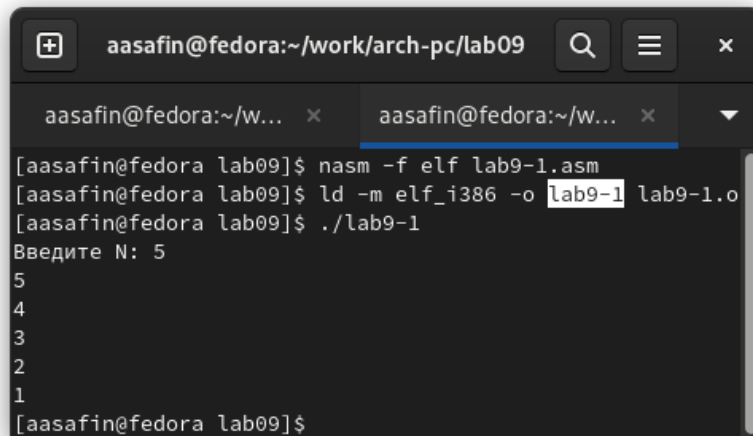


The screenshot shows a terminal window with the prompt 'aasafin@fedora:~/work/arch-...'. The user has executed a series of commands to create the directory structure and the file lab9-1.asm.

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

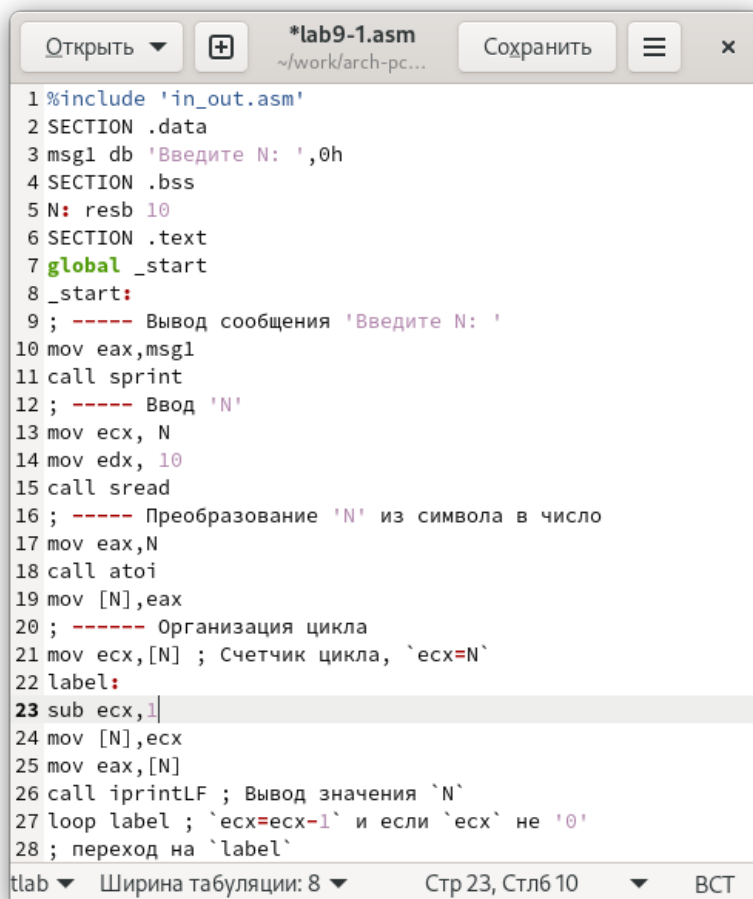
Рис. 3.2: Создание ~/work/arch-pc/lab09/lab9-1.asm





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

Рис. 3.3: Выполнение lab9-1.asm

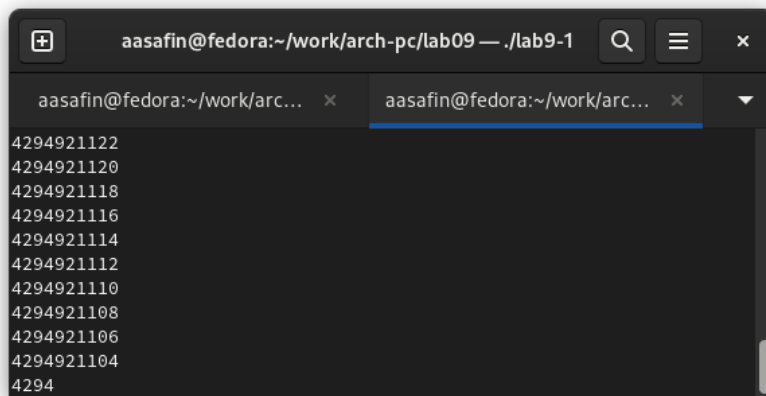


```
*lab9-1.asm
~/work/arch-pc...
Сохранить

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 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`
28 ; переход на `label`

tlab  Ширина табуляции: 8  Стр 23, Стлб 10  ВСТ
```

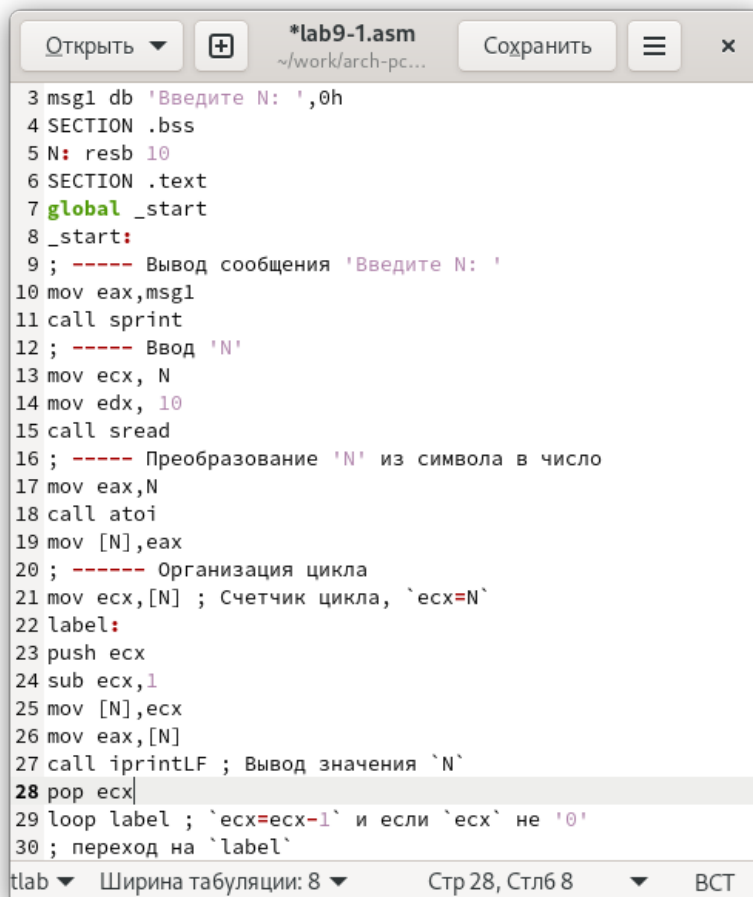
Рис. 3.4: Измененный текст lab9-1.asm (1)



```
aasafin@fedora:~/work/arch-pc/lab09 — ./lab9-1
4294921122
4294921120
4294921118
4294921116
4294921114
4294921112
4294921110
4294921108
4294921106
4294921104
4294
```

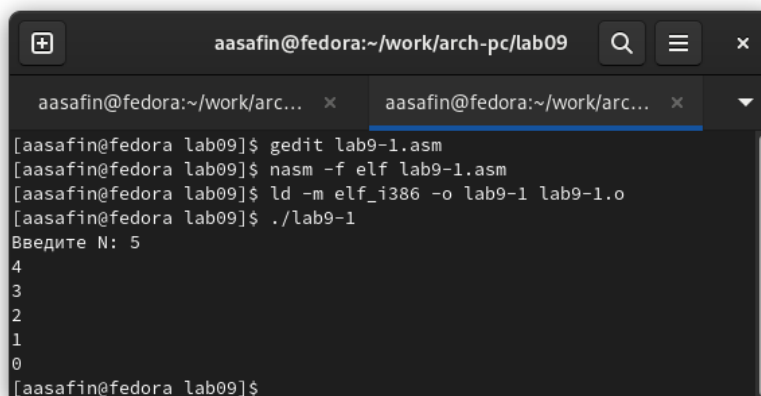
Рис. 3.5: Выполнение измененной программы lab9-1.asm (1)

Затем в тело цикла записана программа, сохраняющая корректное значение `ecx` в стеке (рис. 3.6). Число проходов цикла в ней соответствует числу  $N$ , введенному с клавиатуры (рис. 3.7).



```
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 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`
30 ; переход на `label`
```

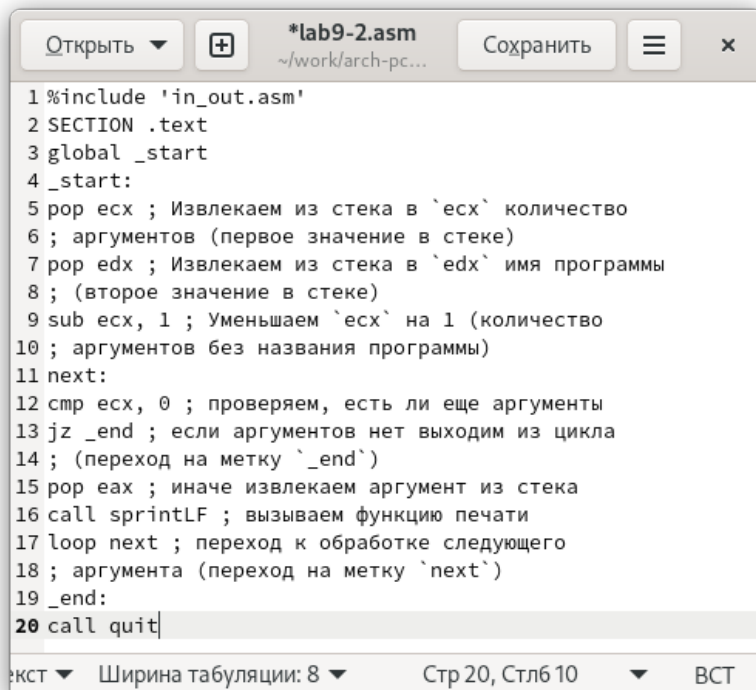
Рис. 3.6: Измененный текст lab9-1.asm (2)



```
aasafin@fedora:~/work/arch-pc/lab09
aasafin@fedora:~/work/arch-pc/lab09
[aasafin@fedora lab09]$ gedit lab9-1.asm
[aasafin@fedora lab09]$ nasm -f elf lab9-1.asm
[aasafin@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[aasafin@fedora lab09]$ ./lab9-1
Введите N: 5
4
3
2
1
0
[aasafin@fedora lab09]$
```

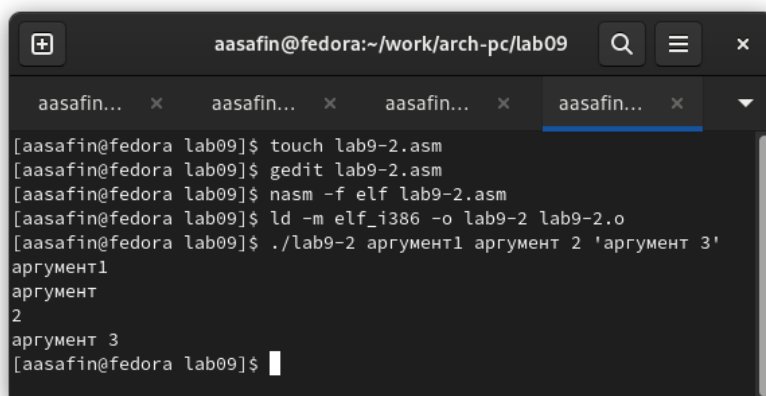
Рис. 3.7: Выполнение измененной программы lab9-1.asm (2)

Создан файл lab9-2.asm, в него введена программа из листинга 9.2, поочередно выводящая введенные аргументы (рис. 3.8). При выполнении было обработано 4 аргумента (рис. 3.9).



```
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 printf ; вызываем функцию печати
17 loop next ; переход к обработке следующего
18 ; аргумента (переход на метку `next`)
19 _end:
20 call quit
```

Рис. 3.8: Текст lab9-2.asm

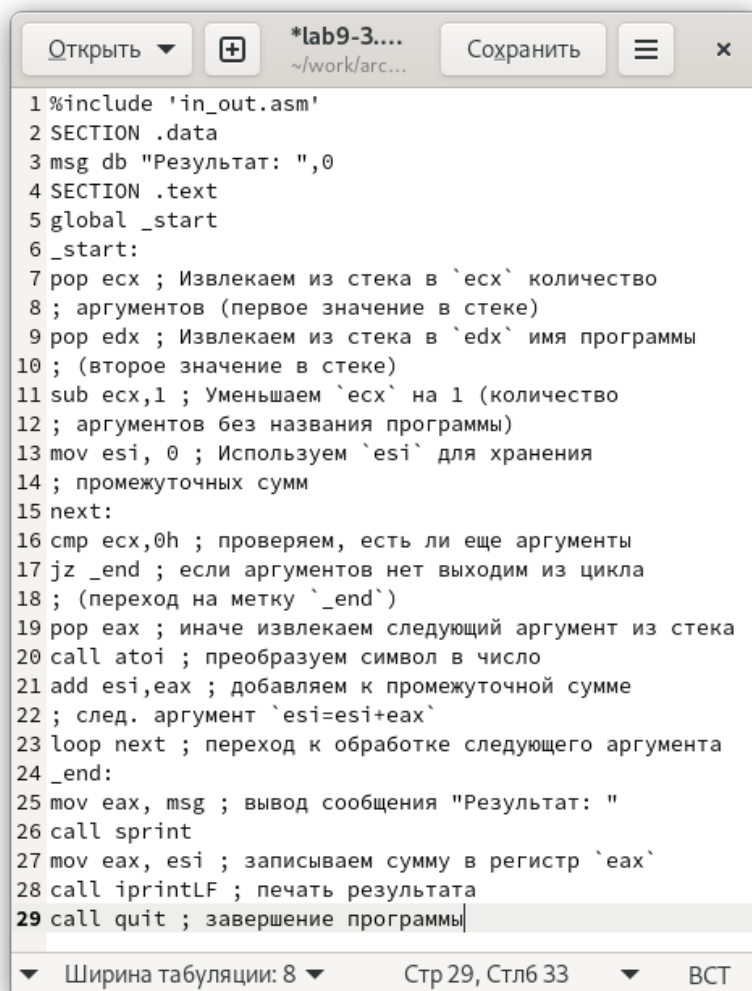


```
aasafin@fedora:~/work/arch-pc/lab09
aasafin... x aasafin... x aasafin... x aasafin... x
[aasafin@fedora lab09]$ touch lab9-2.asm
[aasafin@fedora lab09]$ gedit lab9-2.asm
[aasafin@fedora lab09]$ nasm -f elf lab9-2.asm
[aasafin@fedora lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[aasafin@fedora lab09]$ ./lab9-2 аргумент1 аргумент 2 'аргумент 3'
аргумент1
аргумент 2
аргумент 3
[aasafin@fedora lab09]$
```

Рис. 3.9: Выполнение программы lab9-2.asm

Создан файл lab9-3.asm, в него введена программа из листинга 9.3, вычисляю-

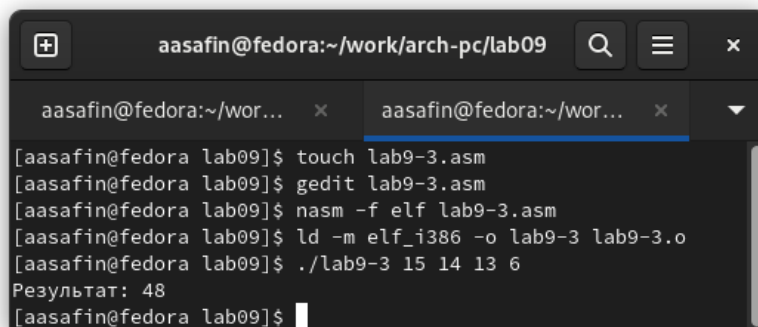
щая сумму всех аргументов, введенных с командной строки, с помощью цикла и стека (рис. 3.10). Её работа корректна (рис. 3.11). Текст изменен так (рис. 3.12), чтоб вычислялась не сумма, а произведение (рис. 3.13).



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

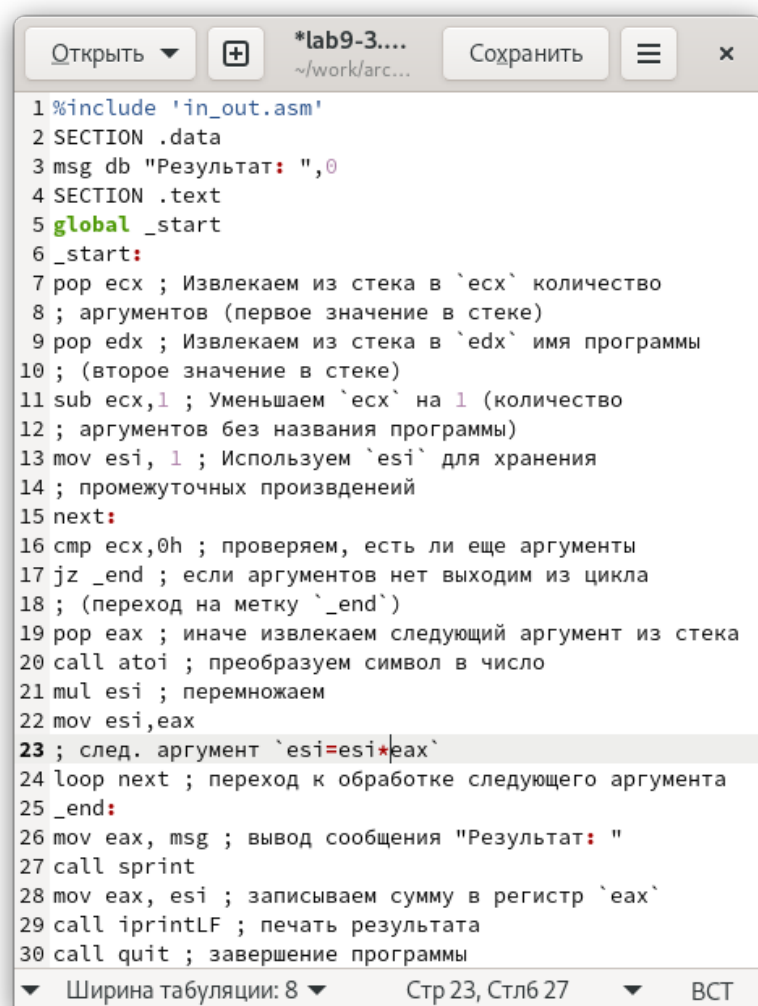
Ширина табуляции: 8 Стр 29, Стлб 33 ВСТ

Рис. 3.10: Текст lab9-3.asm



```
aasafin@fedora:~/work/arch-pc/lab09
[aasafin@fedora lab09]$ touch lab9-3.asm
[aasafin@fedora lab09]$ gedit lab9-3.asm
[aasafin@fedora lab09]$ nasm -f elf lab9-3.asm
[aasafin@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[aasafin@fedora lab09]$ ./lab9-3 15 14 13 6
Результат: 48
[aasafin@fedora lab09]$
```

Рис. 3.11: Работа программы lab9-3.asm

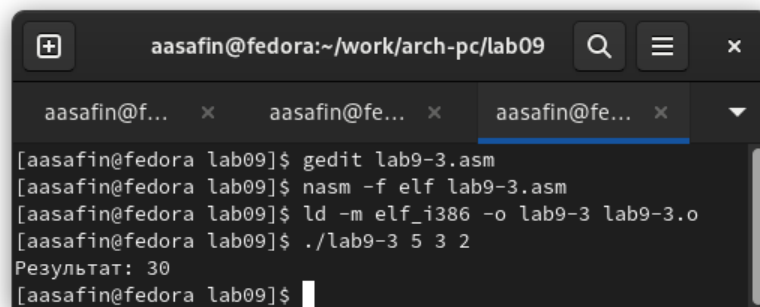


```
*lab9-3....
~/work/arc...
Сохранить

1 %include 'in_out.asm'
2 SECTION .data
3 msg db "Результат: ",0
4 SECTION .text
5 global _start
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 mov esi, 1 ; Используем `esi` для хранения
14 ; промежуточных произведений
15 next:
16 cmp ecx,0h ; проверяем, есть ли еще аргументы
17 jz _end ; если аргументов нет выходим из цикла
18 ; (переход на метку `_end`)
19 pop eax ; иначе извлекаем следующий аргумент из стека
20 call atoi ; преобразуем символ в число
21 mul esi ; перемножаем
22 mov esi,eax
23 ; след. аргумент `esi=esi*eax`
24 loop next ; переход к обработке следующего аргумента
25 _end:
26 mov eax, msg ; вывод сообщения "Результат: "
27 call sprint
28 mov eax, esi ; записываем сумму в регистр `eax`
29 call iprintLF ; печать результата
30 call quit ; завершение программы

Ширина табуляции: 8 Стр 23, Стлб 27 ВСТ
```

Рис. 3.12: Измененный текст lab9-3.asm



The image shows a terminal window with a dark background. The title bar at the top reads 'aasafin@fedora:~/work/arch-pc/lab09'. Below the title bar, there are three tabs, each labeled 'aasafin@fe...'. The terminal content shows a series of commands and their outputs:

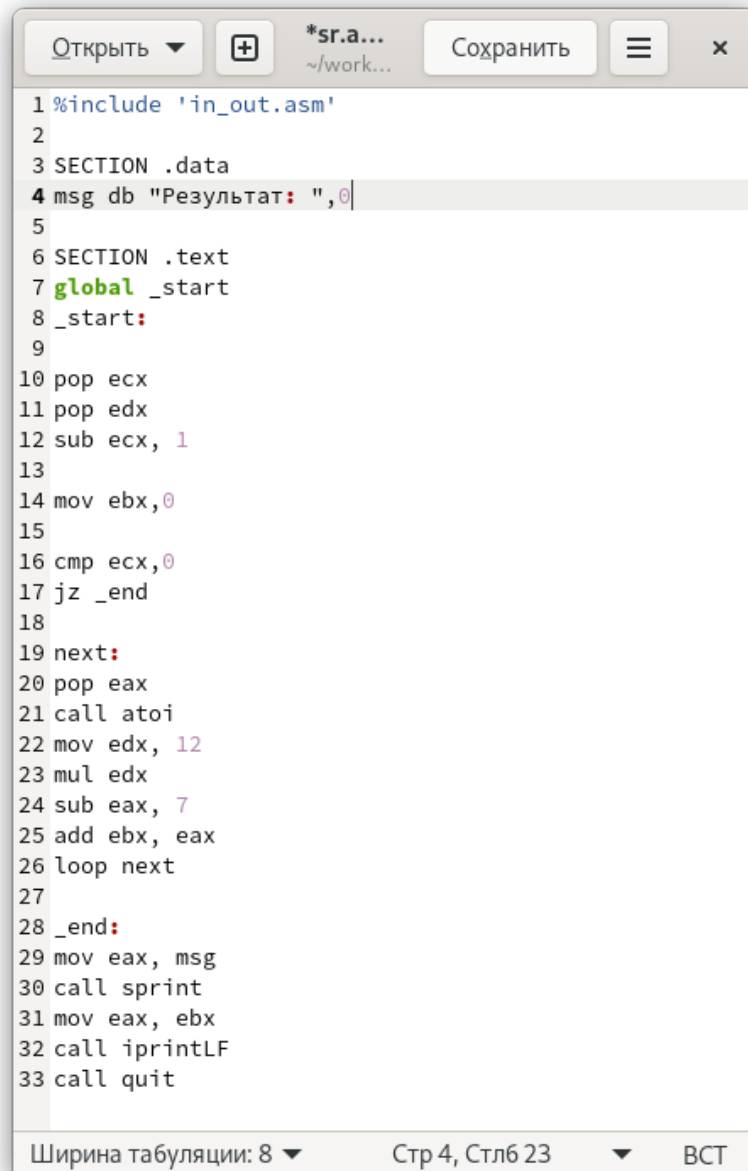
```
[aasafin@fedora lab09]$ gedit lab9-3.asm
[aasafin@fedora lab09]$ nasm -f elf lab9-3.asm
[aasafin@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[aasafin@fedora lab09]$ ./lab9-3 5 3 2
Результат: 30
[aasafin@fedora lab09]$
```

Рис. 3.13: Работа измененной программы lab9-3.asm

## 4 Самостоятельная работа

Написана программа, вычисляющая сумму значений функции  $f(x)=12x-7$  для аргументов, введенных с командной строки (рис. 4.1). Её работа проверена на нескольких примерах (рис. 4.2). Выполнение корректно.





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

Ширина табуляции: 8      Стр 4, Стлб 23      ВСТ

Рис. 4.1: Текст sr.asm

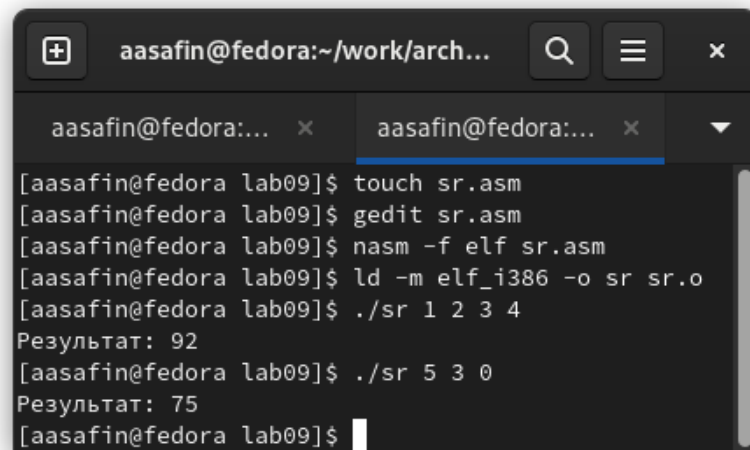
A terminal window with a dark background and light text. The window title is 'aasafin@fedora:~/work/arch...'. It shows a series of commands and their outputs. The commands are: 'touch sr.asm', 'gedit sr.asm', 'nasm -f elf sr.asm', 'ld -m elf\_i386 -o sr sr.o', './sr 1 2 3 4', and './sr 5 3 0'. The outputs are 'Результат: 92' and 'Результат: 75'. The prompt is '[aasafin@fedora lab09]\$'.

Рис. 4.2: Текст sr.asm

## 5 Выводы

Все программы написаны и выполнены корректно. В процессе лабораторной работы приобретен навык работы с командой цикла `loop` и стеком.