

Отчет по лабораторной работе №9

дисциплина: Архитектура компьютера

Колобова Елизавета Андреевна гр. НММбд-01

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Задание для самостоятельной работы	14
3	Выводы	17

Список иллюстраций

2.1	Рис. 1. Создание каталога для лаб. работы №9 и файла lab9-1.asm .	6
2.2	Рис. 2. Ввод текста из листинга 9.1	7
2.3	Рис. 3. Компоновка и запуск файла lab9-1	8
2.4	Рис. 4. Изменение текста программы	9
2.5	Рис. 5. Компоновка и запуск измененного файла	9
2.6	Рис. 6. Изменение текста программы	11
2.7	Рис. 7. Компоновка и запуск измененного файла	11
2.8	Рис. 8. Ввод текста из листинга 9.2	12
2.9	Рис. 9. Компоновка и запуск файла lab9-2	12
2.10	Рис. 10. Ввод текста из листинга 9.3	13
2.11	Рис. 11. Компоновка и запуск файла	13
2.12	Рис. 12. Изменение текста программы	14
2.13	Рис. 13. Компоновка и запуск измененного файла	14
2.14	Рис. 14. Текст программы	15
2.15	Рис. 15. Компоновка и запуск файла	15
2.16	Рис. 16. Компоновка и запуск файла	16

Список таблиц

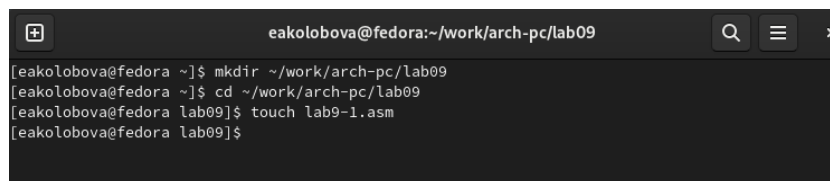
1 Цель работы

Целью работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

2 Выполнение лабораторной работы

1. Создадим каталог для программ лабораторной работы No 9, перейдем в него и создадим файл lab9-1.asm (рис. 2.1):

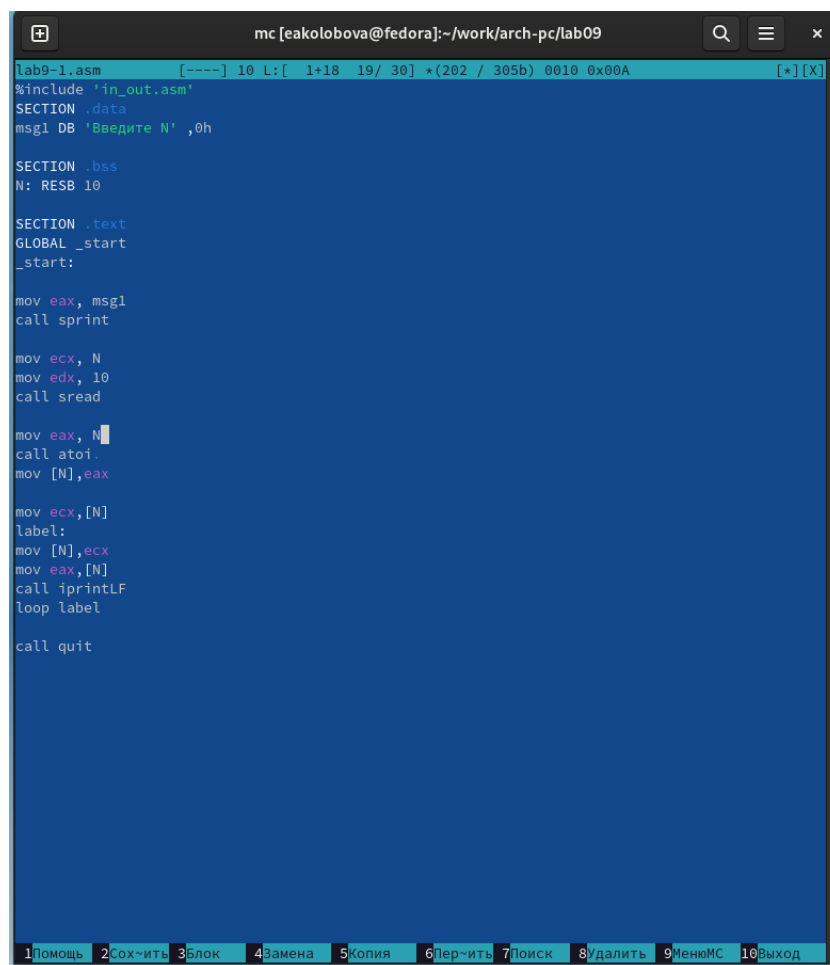
```
mkdir ~/work/arch-pc/lab09  
cd ~/work/arch-pc/lab09  
touch lab9-1.asm
```

A screenshot of a terminal window with a dark background. The title bar shows the user 'eakolobova' on a 'fedora' machine, in the directory '~/work/arch-pc/lab09'. The terminal contains four lines of text: a prompt followed by 'mkdir ~/work/arch-pc/lab09', a prompt followed by 'cd ~/work/arch-pc/lab09', a prompt followed by 'touch lab9-1.asm', and a final prompt. The window has standard Linux window controls (minimize, maximize, close) and a search icon in the top right corner.

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

Рис. 2.1: Рис. 1. Создание каталога для лаб. работы №9 и файла lab9-1.asm

2. Рассмотрим программу, которая выводит значение регистра esx. Введем в файл lab9-1.asm текст программы из листинга 9.1. Создадим исполняемый файл и проверим его работу (рис. 2.2), (рис. 2.3).



```
lab9-1.asm [----] 10 L: [ 1+18 19/ 30] *(202 / 305b) 0010 0x00A [*] [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
call sprint

mov ecx, N
mov edx, 10
call sread

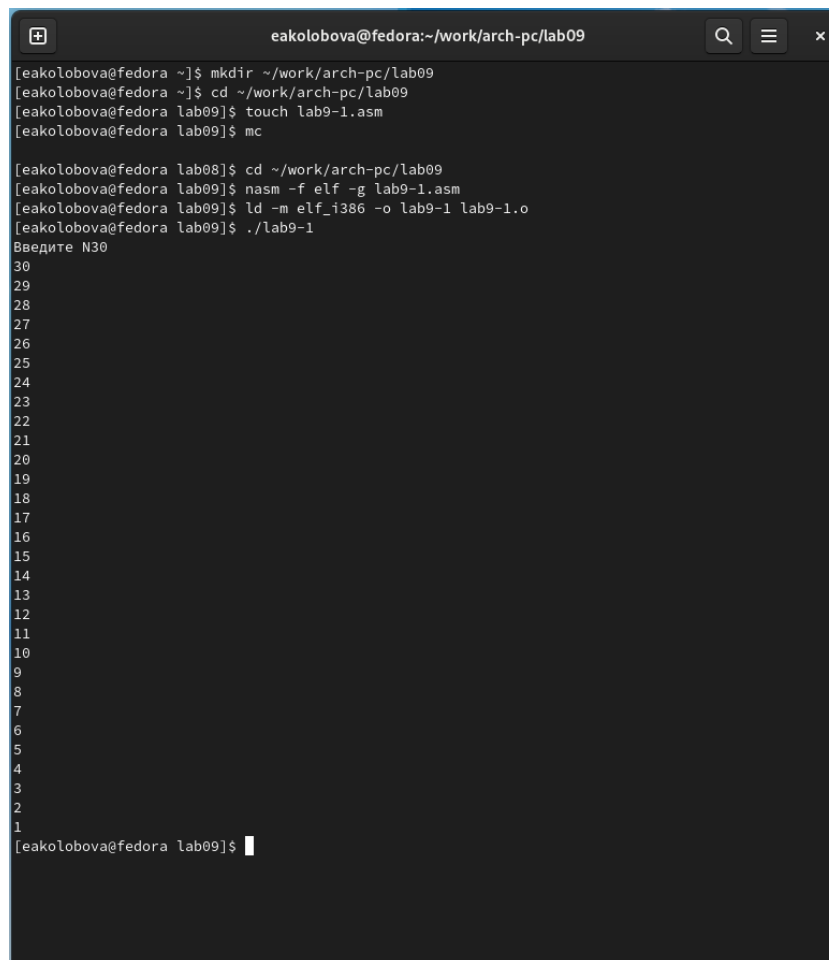
mov eax, N
call atoi
mov [N], eax

mov ecx, [N]
label:
mov [N], ecx
mov eax, [N]
call iprintLF
loop label

call quit
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС 10Выход

Рис. 2.2: Рис. 2. Ввод текста из листинга 9.1

A terminal window titled 'eakolobova@fedora:~/work/arch-pc/lab09'. The terminal shows the following commands and output:

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

[eakolobova@fedora lab08]$ cd ~/work/arch-pc/lab09
[eakolobova@fedora lab09]$ nasm -f elf -g lab9-1.asm
[eakolobova@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[eakolobova@fedora lab09]$ ./lab9-1
Введите N30
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
[eakolobova@fedora lab09]$
```

Рис. 2.3: Рис. 3. Компоновка и запуск файла lab9-1

Данный пример показывает, что использование регистра ecx в теле цикла loop может привести к некорректной работе программы.

3. Изменим текст программы, добавив изменение значения регистра ecx в цикле (рис. 2.4, 2.5):

```
label:
sub ecx,1
mov [N],ecx
mov eax,[N]
call iprintLF
loop label 5
```


Создадим исполняемый файл и проверим его работу. Регистр `ecx` в цикле принимает в первом подходе введенное значение, уменьшенное на 1, в каждом следующем - уменьшенное на 2. Число проходов цикла не соответствует значению `N`, введенному с клавиатуры.

```
lab9-1.asm [----] 9 L: [ 1+32 33/ 33] *(332 / 332b) <EOF> [*] [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
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
```

Рис. 2.4: Рис. 4. Изменение текста программы

```
[eakolobova@fedora ~]$ cd ~/work/arch-pc/lab09
[eakolobova@fedora lab09]$ ./lab9-1
Введите N2
1
[eakolobova@fedora lab09]$ ./lab9-1
Введите N6
5
3
1
[eakolobova@fedora lab09]$
```

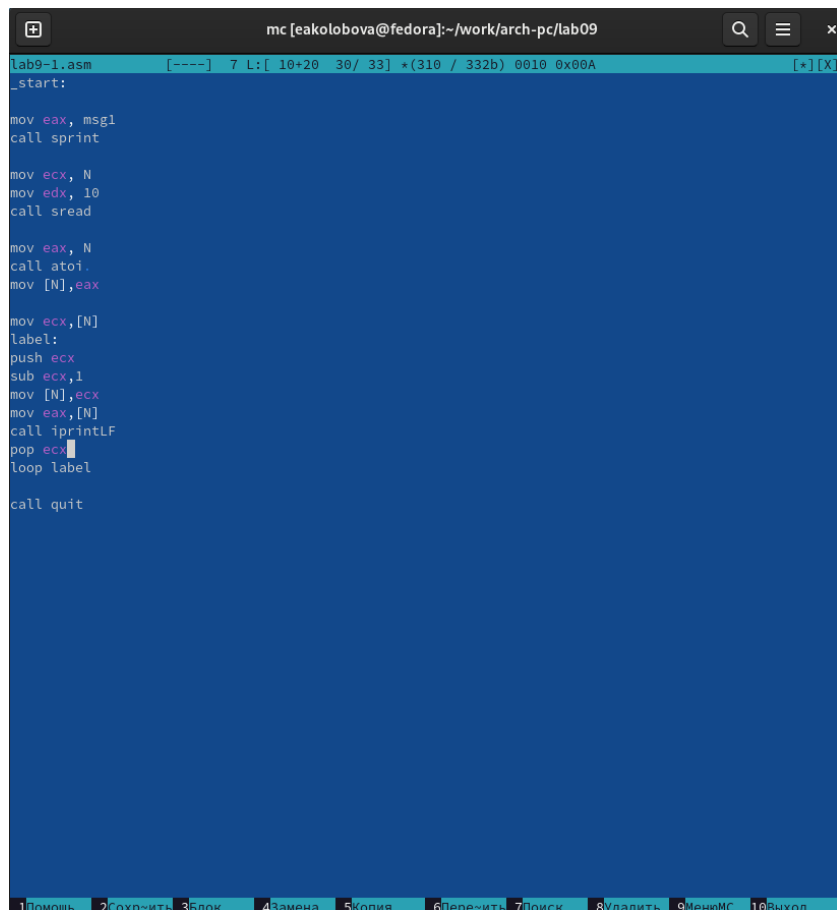
Рис. 2.5: Рис. 5. Компоновка и запуск измененного файла

4. Внесем изменения в текст программы, добавив команды `push` и `pop` (до-

бавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop: (рис. 2.6):

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

Создадим исполняемый файл и проверим его работу. Соответствует ли в данном случае число проходов цикла значению N введенному с клавиатур (рис. 2.7)



```
mc [eakolobova@fedora]:~/work/arch-pc/lab09
lab9-1.asm [----] 7 L: [ 10+20 30/ 33] *(310 / 332b) 0010 0x00A [*] [X]
_start:
mov eax, msg1
call sprint

mov ecx, N
mov edx, 10
call sread

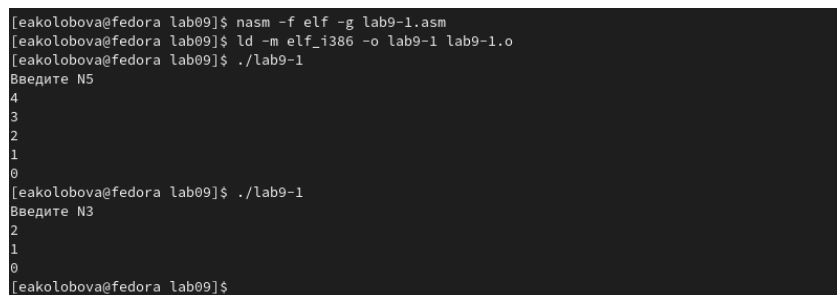
mov eax, N
call atoi
mov [N], eax

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

call quit
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Переместить 7Поиск 8Удалить 9МенюМС 10Выход

Рис. 2.6: Рис. 6. Изменение текста программы



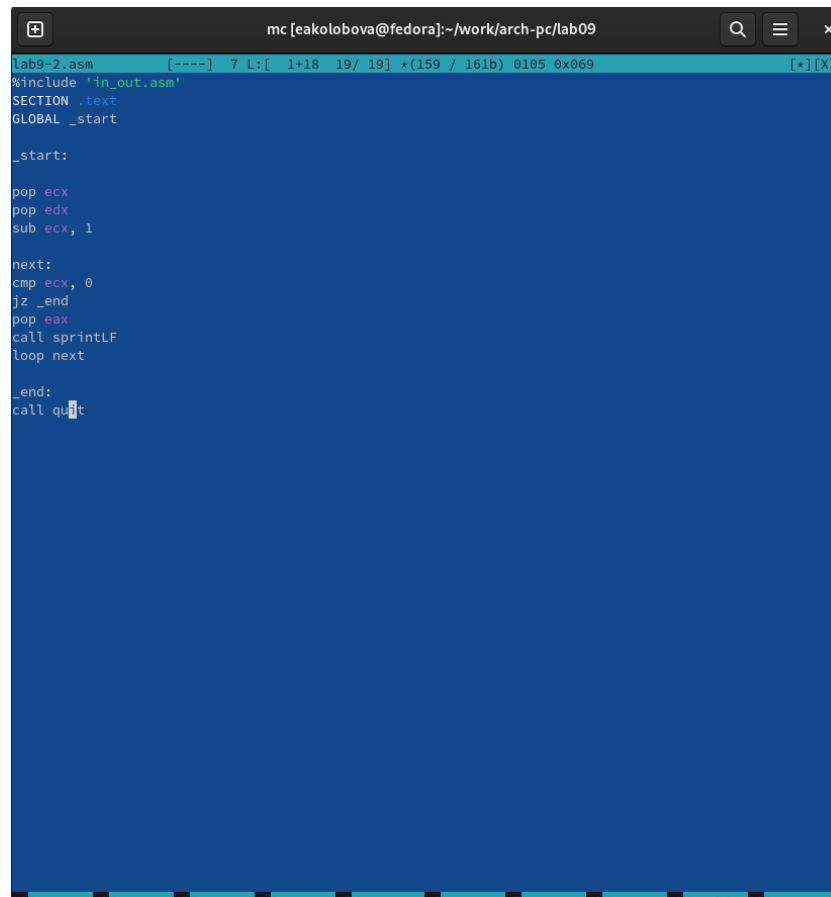
```
[eakolobova@fedora lab09]$ nasm -f elf -g lab9-1.asm
[eakolobova@fedora lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[eakolobova@fedora lab09]$ ./lab9-1
Введите N5
4
3
2
1
0
[eakolobova@fedora lab09]$ ./lab9-1
Введите N3
2
1
0
[eakolobova@fedora lab09]$
```

Рис. 2.7: Рис. 7. Компоновка и запуск измененного файла

5. Рассмотрим программу, которая выводит на экран аргументы командной строки. Создадим файл lab9-2.asm в каталоге ~/work/arch-pc/lab09 и введем в него текст программы из листинга 9.2. Создадим исполняемый файл и запустим его, указав аргументы:

user@dk4n31:~\$./lab9-2 аргумент1 аргумент 2 'аргумент 3'

Сколько аргументов было обработано программой?(рис. 2.8, 2.9):



```
lab9-2.asm [----] 7 L: [ 1+18 19/ 19] *(159 / 161b) 0105 0x069 [*] [X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start

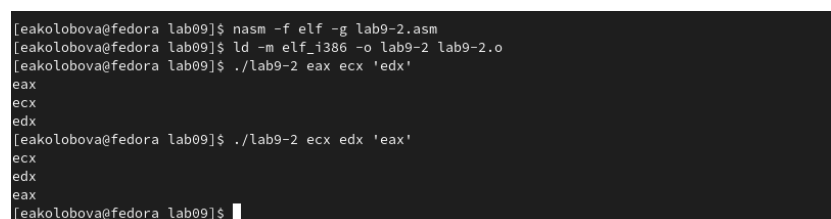
_start:

pop ecx
pop edx
sub ecx, 1

next:
cmp ecx, 0
jz _end
pop eax
call printf
loop next

_end:
call quit
```

Рис. 2.8: Рис. 8. Ввод текста из листинга 9.2



```
[eakolobova@fedora lab09]$ nasm -f elf -g lab9-2.asm
[eakolobova@fedora lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[eakolobova@fedora lab09]$ ./lab9-2 eax ecx 'edx'
eax
ecx
edx
[eakolobova@fedora lab09]$ ./lab9-2 ecx edx 'eax'
ecx
edx
eax
[eakolobova@fedora lab09]$
```

Рис. 2.9: Рис. 9. Компоновка и запуск файла lab9-2

6. Рассмотрим еще один пример программы которая выводит сумму чисел, которые передаются в программу как аргументы. Создадим файл lab9-3.asm в каталоге ~/work/arch-pc/lab09 и введем в него текст программы из листинга

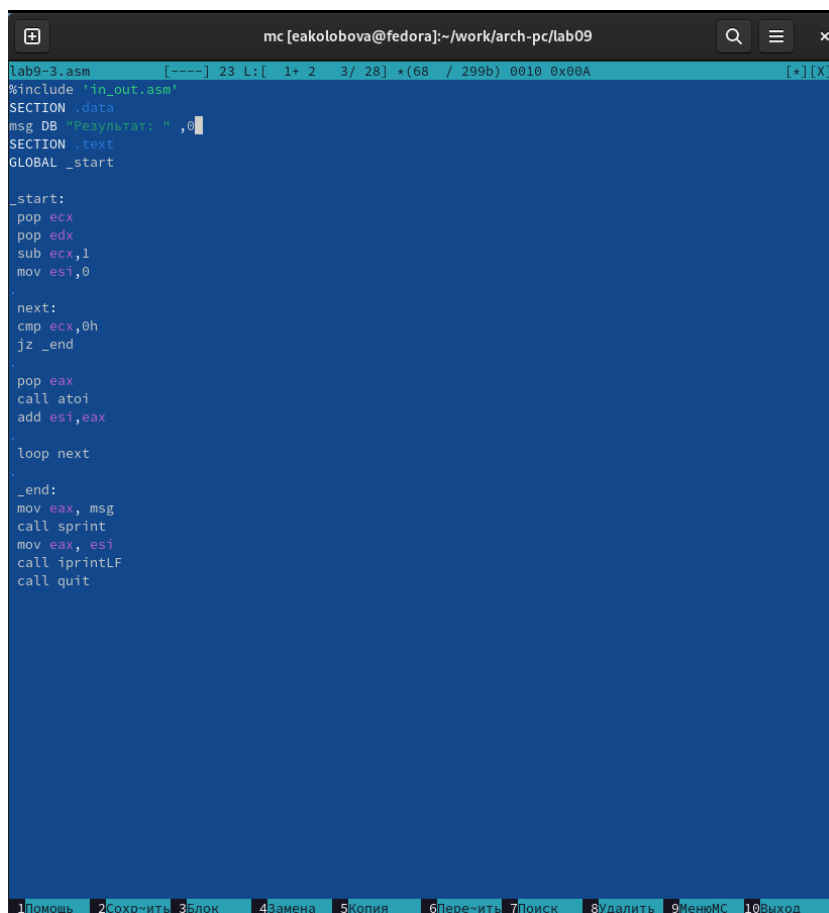
9.3 Создадим исполняемый файл и запустим его, указав аргументы. Пример результата работы программы:

```
user@dk4n31:~$ ./main 12 13 7 10 5
```

Результат: 47

```
user@dk4n31:~$
```

(рис. 2.10, 2.11)



```
lab9-3.asm [----] 23 L: [ 1+ 2 3/ 28] *(68 / 299b) 0010 0x00A [*] [X]
%include 'in_out.asm'
SECTION .data
msg DB "Результат: ",0
SECTION .text
GLOBAL _start

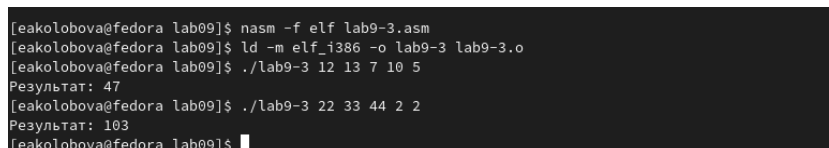
_start:
pop ecx
pop edx
sub ecx,1
mov esi,0

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
```

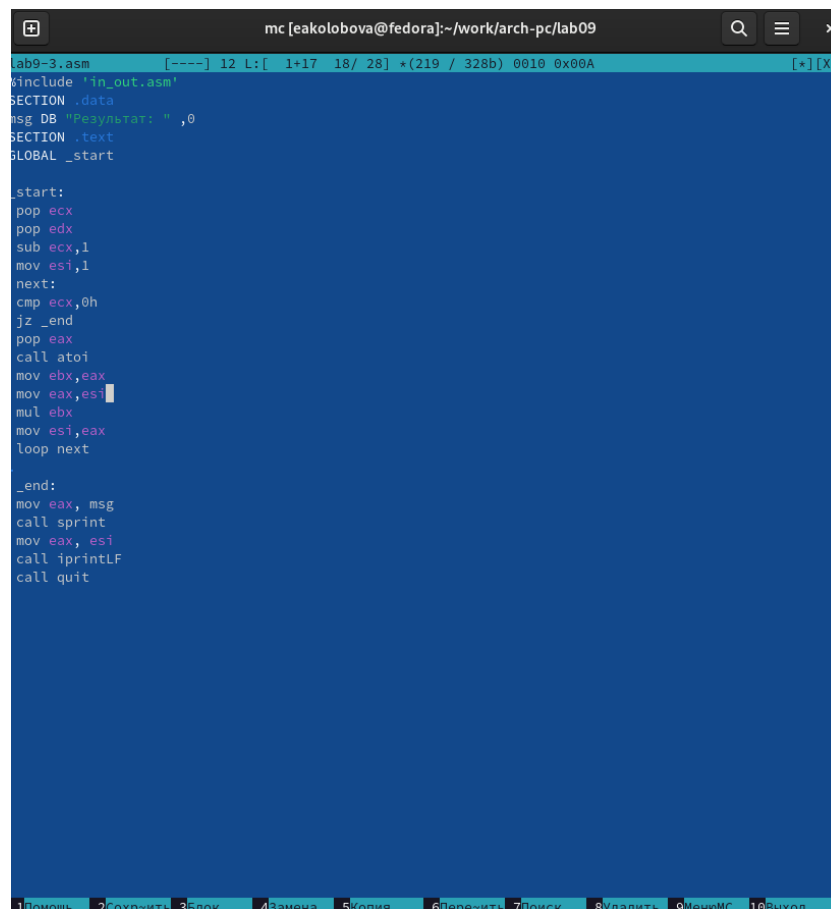
Рис. 2.10: Рис. 10. Ввод текста из листинга 9.3



```
[eakolobova@fedora lab09]$ nasm -f elf lab9-3.asm
[eakolobova@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[eakolobova@fedora lab09]$ ./lab9-3 12 13 7 10 5
Результат: 47
[eakolobova@fedora lab09]$ ./lab9-3 22 33 44 2 2
Результат: 103
[eakolobova@fedora lab09]$
```

Рис. 2.11: Рис. 11. Компоновка и запуск файла

7. Изменим текст программы из листинга 9.3 для вычисления произведения аргументов командной строки. (рис. 2.12, 2.13)

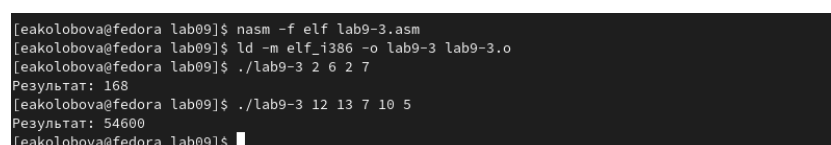


```
lab9-3.asm [----] 12 L: [ 1+17 18/ 28] *(219 / 328b) 0010 0x00A [*] [X]
#include 'in_out.asm'
SECTION .data
msg DB "Результат: ",0
SECTION .text
GLOBAL _start

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

_end:
    mov eax,msg
    call sprint
    mov eax,esi
    call iprintf
    call quit
```

Рис. 2.12: Рис. 12. Изменение текста программы



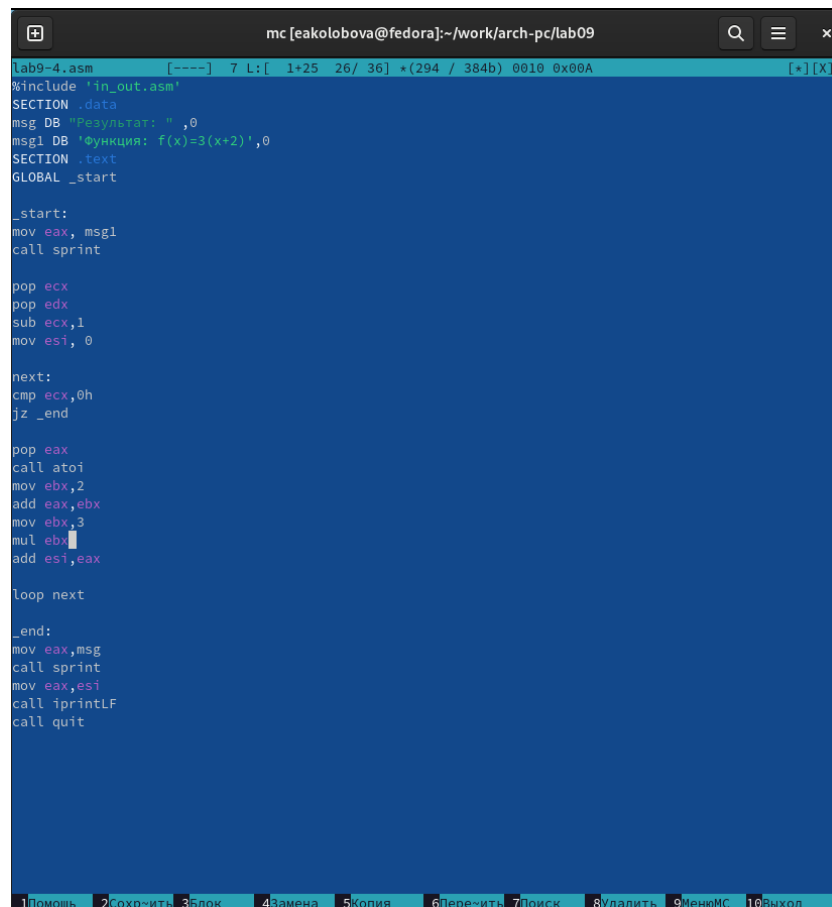
```
[eakolobova@fedora lab09]$ nasm -f elf lab9-3.asm
[eakolobova@fedora lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[eakolobova@fedora lab09]$ ./lab9-3 2 6 2 7
Результат: 168
[eakolobova@fedora lab09]$ ./lab9-3 12 13 7 10 5
Результат: 54600
[eakolobova@fedora lab09]$
```

Рис. 2.13: Рис. 13. Компоновка и запуск измененного файла

2.1 Задание для самостоятельной работы

В ходе выполнения работы № 7 был получен вариант 7, выражение в табл. 9.4:
 $f(x) = 6a$,

1. Написать 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 9.4 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы No 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах $x = x_1, x_2, \dots, x_n$ (рис. 2.14, 2.15, 2.16)



```
lab9-4.asm [----] 7 L: [ 1+25 26/ 36] *(294 / 384b) 0010 0x00A [*][X]
#include "in_out.asm"
SECTION .data
msg DB "Результат: ",0
msg1 DB "Функция: f(x)=3(x+2)",0
SECTION .text
GLOBAL _start
_start:
mov eax, msg1
call sprint

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

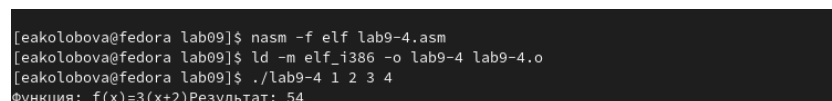
next:
cmp ecx, 0h
jz _end

pop eax
call atoi
mov ebx, 2
add eax, ebx
mov ebx, 3
mul ebx
add esi, eax

loop next

_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис. 2.14: Рис. 14. Текст программы



```
[eakolobova@fedora lab09]$ nasm -f elf lab9-4.asm
[eakolobova@fedora lab09]$ ld -m elf_i386 -o lab9-4 lab9-4.o
[eakolobova@fedora lab09]$ ./lab9-4 1 2 3 4
Функция: f(x)=3(x+2)Результат: 54
```

Рис. 2.15: Рис. 15. Компоновка и запуск файла

```
[eakolobova@fedora lab09]$ ./lab9-4 12 13 7 10 5  
Функция:  $f(x)=3(x+2)$ Результат: 171  
[eakolobova@fedora lab09]$
```

Рис. 2.16: Рис. 16. Компоновка и запуск файла

Ссылка на репозиторий: https://github.com/eakolobova/study_2022-2023_arch-pc/tree/master/labs/lab09/report

3 Выводы

Результатом проведенной работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.