

Отчёт по лабораторной работе №9

Нефёдова Наталия Николаевна

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
2.1	Реализация циклов в NASM	5
2.2	Обработка аргументов командной строки	8
2.3	Листинг 9.3. Программа вычисления суммы аргументов командной строки	9
2.4	Задание для самостоятельной работы	10
3	Выводы	13

Список иллюстраций

[illegible]

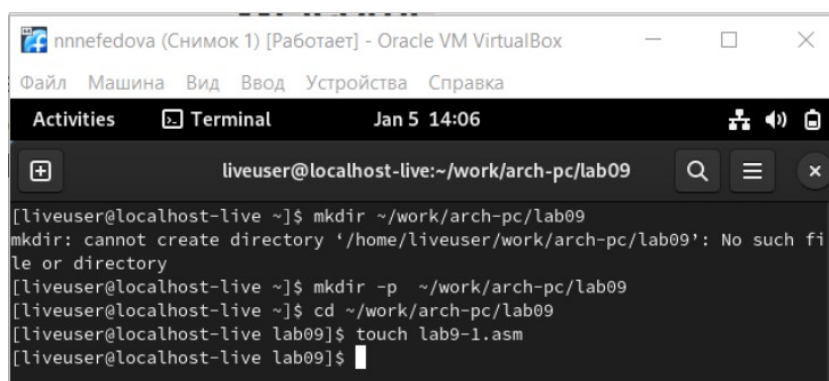
1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Реализация циклов в NASM

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



```
nnnefedova (Снимок 1) [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Activities Terminal Jan 5 14:06
liveuser@localhost-live:~/work/arch-pc/lab09

[liveuser@localhost-live ~]$ mkdir ~/work/arch-pc/lab09
mkdir: cannot create directory '/home/liveuser/work/arch-pc/lab09': No such file or directory
[liveuser@localhost-live ~]$ mkdir -p ~/work/arch-pc/lab09
[liveuser@localhost-live ~]$ cd ~/work/arch-pc/lab09
[liveuser@localhost-live lab09]$ touch lab9-1.asm
[liveuser@localhost-live lab09]$
```

Рис. 2.1: 1

Введем в файл lab9-1.asm текст программы из листинга 9.1. Создадим исполняемый файл и проверим его работу. Данный пример показывает, что использование регистра esx в теле цикла loop может привести к некорректной работе программы. Изменим текст программы добавив изменение значение регистра esx в цикле. (рис. 2.2), (рис. 2.3)

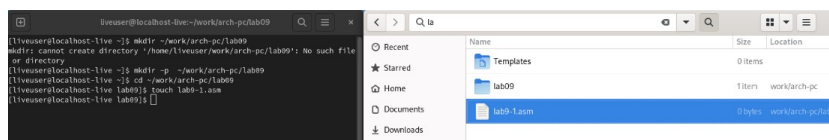


Рис. 2.2: 2



```
Open  + lab9-1.asm
~\work\arch-pc\lab09

%include "in_out.asm"
SECTION .data
msg1 db 'Введите N: ',0h
SECTION .bss
N: resb 10
SECTION .text
global _start
_start:
; ----- Вывод сообщения 'Введите N: '

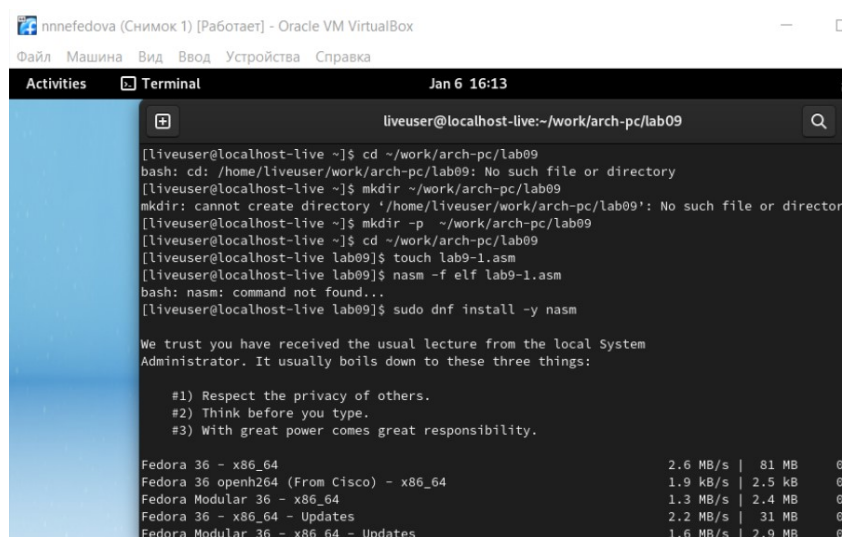
mov eax,msg1
call sprint
; ----- Ввод 'N'
mov ecx, N
mov edx, 10
call sread

; ----- Преобразование 'N' из символа в число

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

Рис. 2.3: 3

Установим Nasm: (рис. 2.4), (рис. 2.5), (рис. 2.6), (рис. 2.7)



```
nnnefedova (Снимок 1) [Работаer] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
Activities Terminal Jan 6 16:13
liveuser@localhost-live:~/work/arch-pc/lab09

[liveuser@localhost-live ~]$ cd ~/work/arch-pc/lab09
bash: cd: /home/liveuser/work/arch-pc/lab09: No such file or directory
[liveuser@localhost-live ~]$ mkdir ~/work/arch-pc/lab09
mkdir: cannot create directory '/home/liveuser/work/arch-pc/lab09': No such file or director
[liveuser@localhost-live ~]$ mkdir -p ~/work/arch-pc/lab09
[liveuser@localhost-live ~]$ cd ~/work/arch-pc/lab09
[liveuser@localhost-live lab09]$ touch lab9-1.asm
[liveuser@localhost-live lab09]$ nasm -f elf lab9-1.asm
bash: nasm: command not found...
[liveuser@localhost-live lab09]$ sudo dnf install -y nasm

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.

Fedora 36 - x86_64 2.6 MB/s | 81 MB 0
Fedora 36 openh264 (From Cisco) - x86_64 1.9 kB/s | 2.5 kB 0
Fedora Modular 36 - x86_64 1.3 MB/s | 2.4 MB 0
Fedora 36 - x86_64 - Updates 2.2 MB/s | 31 MB 0
Fedora Modular 36 - x86_64 - Updates 1.6 MB/s | 2.9 MB 0
```

Рис. 2.4: 4

```
liveuser@localhost-live:~/work/study/arch-pc/lab09

Installed:
  nasm-2.15.05-2.fc36.x86_64

Complete!
[liveuser@localhost-live lab09]$ nasm -f elf lab9-1.asm
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
ld: warning: cannot find entry symbol _start; not setting start address
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
ld: warning: cannot find entry symbol _start; not setting start address
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
ld: cannot find lab9-1.o: No such file or directory
[liveuser@localhost-live lab09]$ touch lab9-1.asm
[liveuser@localhost-live lab09]$ nasm -f elf lab9-1.asm
lab9-1.asm:1: error: unable to open include file 'in_out.asm': No such file or directory
[liveuser@localhost-live lab09]$ nasm -f elf lab9-1.asm
lab9-1.asm:1: error: unable to open include file 'in_out.asm': No such file or directory
[liveuser@localhost-live lab09]$ nasm -f elf lab9-1.asm
lab9-1.asm:1: error: unable to open include file 'in_out.asm': No such file or directory
[liveuser@localhost-live lab09]$ nasm -f elf lab9-1.asm
[liveuser@localhost-live lab09]$ d -m elf_i386 -o lab9-1 lab9-1.o
bash: d: command not found...
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[liveuser@localhost-live lab09]$
```

Рис. 2.5: 5

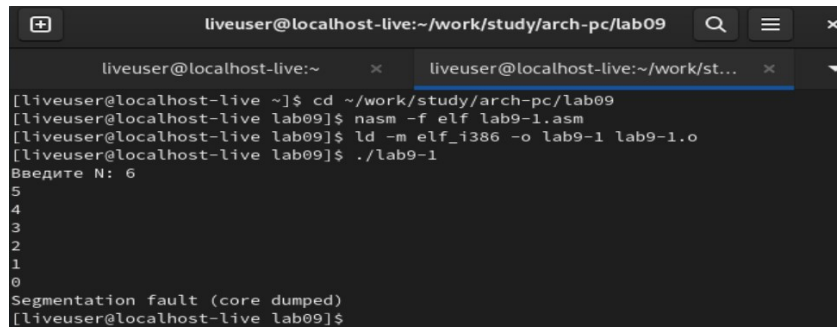
```
[liveuser@localhost-live lab09]$ nasm -f elf lab9-1.asm
[liveuser@localhost-live lab09]$ d -m elf_i386 -o lab9-1 lab9-1.o
bash: d: command not found...
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[liveuser@localhost-live lab09]$ ./lab9-1
Введите N: 5
5
4
3
2
1
[liveuser@localhost-live lab09]$
```

Рис. 2.6: 6

```
Введите N: 5
4
2
0
4294967294
4294967292
4294967290
```

Рис. 2.7: 7

Для использования регистра есх в цикле и сохранения корректности работы программы можно использовать стек. Внесем изменения в текст программы добавив команды `push` и `pop` (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла `loop`: (рис. 2.8)

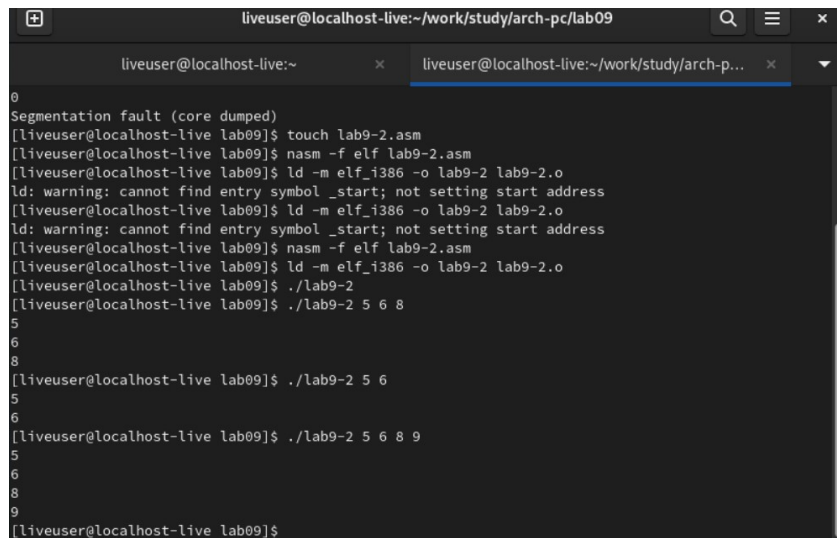


```
liveuser@localhost-live:~/work/study/arch-pc/lab09
liveuser@localhost-live:~ x liveuser@localhost-live:~/work/st... x
[liveuser@localhost-live ~]$ cd ~/work/study/arch-pc/lab09
[liveuser@localhost-live lab09]$ nasm -f elf lab9-1.asm
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-1 lab9-1.o
[liveuser@localhost-live lab09]$ ./lab9-1
Введите N: 6
5
4
3
2
1
0
Segmentation fault (core dumped)
[liveuser@localhost-live lab09]$
```

Рис. 2.8: 8

2.2 Обработка аргументов 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 (рис. 2.9), (рис. 2.10)



```
liveuser@localhost-live:~/work/study/arch-pc/lab09
liveuser@localhost-live:~ x liveuser@localhost-live:~/work/study/arch-p... x
0
Segmentation fault (core dumped)
[liveuser@localhost-live lab09]$ touch lab9-2.asm
[liveuser@localhost-live lab09]$ nasm -f elf lab9-2.asm
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
ld: warning: cannot find entry symbol _start; not setting start address
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
ld: warning: cannot find entry symbol _start; not setting start address
[liveuser@localhost-live lab09]$ nasm -f elf lab9-2.asm
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-2 lab9-2.o
[liveuser@localhost-live lab09]$ ./lab9-2
[liveuser@localhost-live lab09]$ ./lab9-2 5 6 8
5
6
8
[liveuser@localhost-live lab09]$ ./lab9-2 5 6
5
6
[liveuser@localhost-live lab09]$ ./lab9-2 5 6 8 9
5
6
8
9
[liveuser@localhost-live lab09]$
```

Рис. 2.9: 9



```

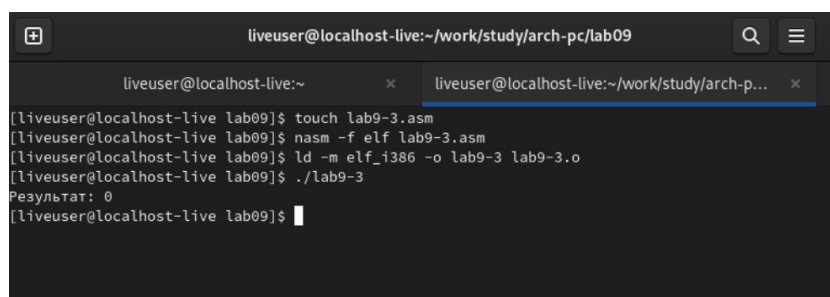
%include 'in out.asm'
SECTION .text
global _start
_start:
pop ecx ; Извлекаем из стека в `ecx` количество
; аргументов (первое значение в стеке)
pop edx ; Извлекаем из стека в `edx` имя программы
; (второе значение в стеке)
sub ecx, 1 ; Уменьшаем `ecx` на 1 (количество
; аргументов без названия программы)
next:
cmp ecx, 0 ; проверяем, есть ли еще аргументы
jz _end ; если аргументов нет выходим из цикла
; (переход на метку `_end`)
pop eax ; иначе извлекаем аргумент из стека
call printf ; вызываем функцию печати
loop next ; переход к обработке следующего
; аргумента (переход на метку `next`)
_end:
call quit

```

Рис. 2.10: 10

2.3 Листинг 9.3. Программа вычисления суммы аргументов командной строки

(рис. 2.11), (рис. 2.12)

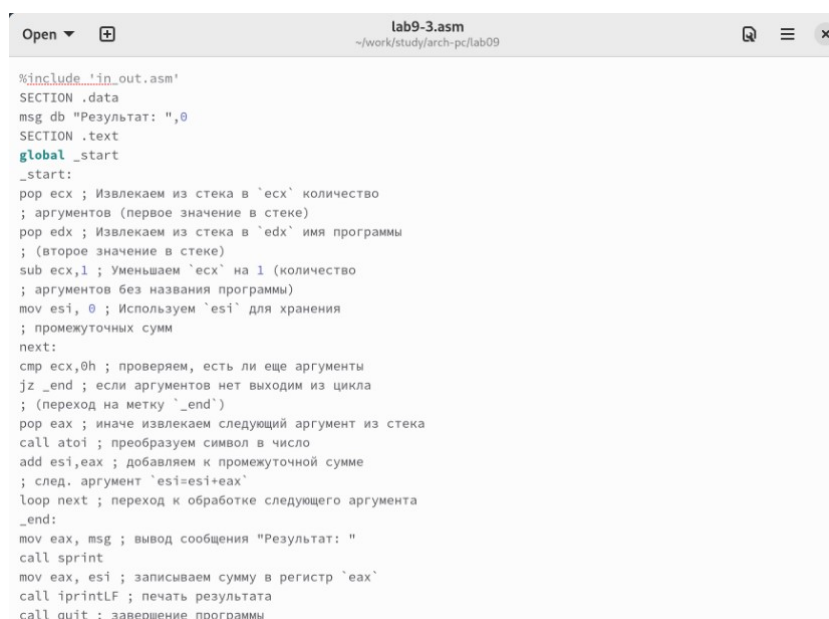


```

liveuser@localhost-live:~/work/study/arch-pc/lab09
liveuser@localhost-live:~ × liveuser@localhost-live:~/work/study/arch-p... ×
[liveuser@localhost-live lab09]$ touch lab9-3.asm
[liveuser@localhost-live lab09]$ nasm -f elf lab9-3.asm
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[liveuser@localhost-live lab09]$ ./lab9-3
Результат: 0
[liveuser@localhost-live lab09]$

```

Рис. 2.11: 11



```

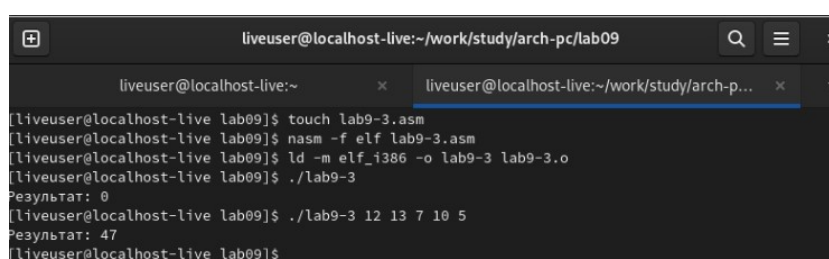
Open  + lab9-3.asm
~/work/study/arch-pc/lab09

%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
    pop ecx ; Извлекаем из стека в `ecx` количество
    ; аргументов (первое значение в стеке)
    pop edx ; Извлекаем из стека в `edx` имя программы
    ; (второе значение в стеке)
    sub ecx,1 ; Уменьшаем `ecx` на 1 (количество
    ; аргументов без названия программы)
    mov esi, 0 ; Используем `esi` для хранения
    ; промежуточных сумм
next:
    cmp ecx,0h ; проверяем, есть ли еще аргументы
    jz _end ; если аргументов нет выходим из цикла
    ; (переход на метку `_end`)
    pop eax ; иначе извлекаем следующий аргумент из стека
    call atoi ; преобразуем символ в число
    add esi,eax ; добавляем к промежуточной сумме
    ; след. аргумент `esi=esi+eax`
    loop next ; переход к обработке следующего аргумента
_end:
    mov eax, msg ; вывод сообщения "Результат: "
    call sprint
    mov eax, esi ; записываем сумму в регистр `eax`
    call iprintLF ; печать результата
    call quit ; завершение программы

```

Рис. 2.12: 12

Проверим работу программы и увидим, что она работает верно. (рис. 2.13)



```

liveuser@localhost-live:~/work/study/arch-pc/lab09

liveuser@localhost-live:~ x liveuser@localhost-live:~/work/study/arch-p... x
[liveuser@localhost-live lab09]$ touch lab9-3.asm
[liveuser@localhost-live lab09]$ nasm -f elf lab9-3.asm
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o lab9-3 lab9-3.o
[liveuser@localhost-live lab09]$ ./lab9-3
Результат: 0
[liveuser@localhost-live lab09]$ ./lab9-3 12 13 7 10 5
Результат: 47
[liveuser@localhost-live lab09]$

```

Рис. 2.13: 13

2.4 Задание для самостоятельной работы

Напишем программу, которая находит сумму значений функции. Значения передаются как аргументы. Создадим исполняемый файл и проверим его работу на нескольких наборах: (рис. 2.14), (рис. 2.15), (рис. 2.16)

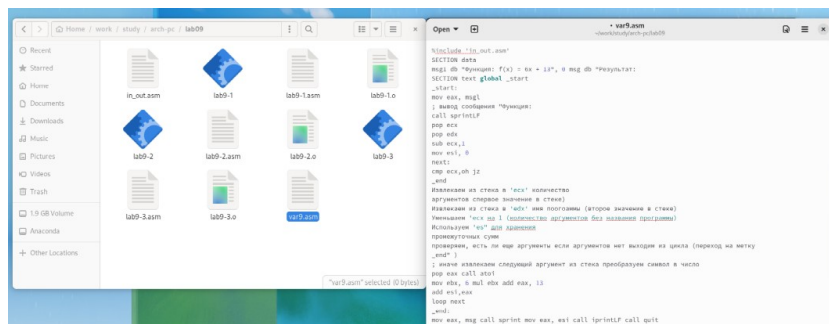


Рис. 2.14: 14

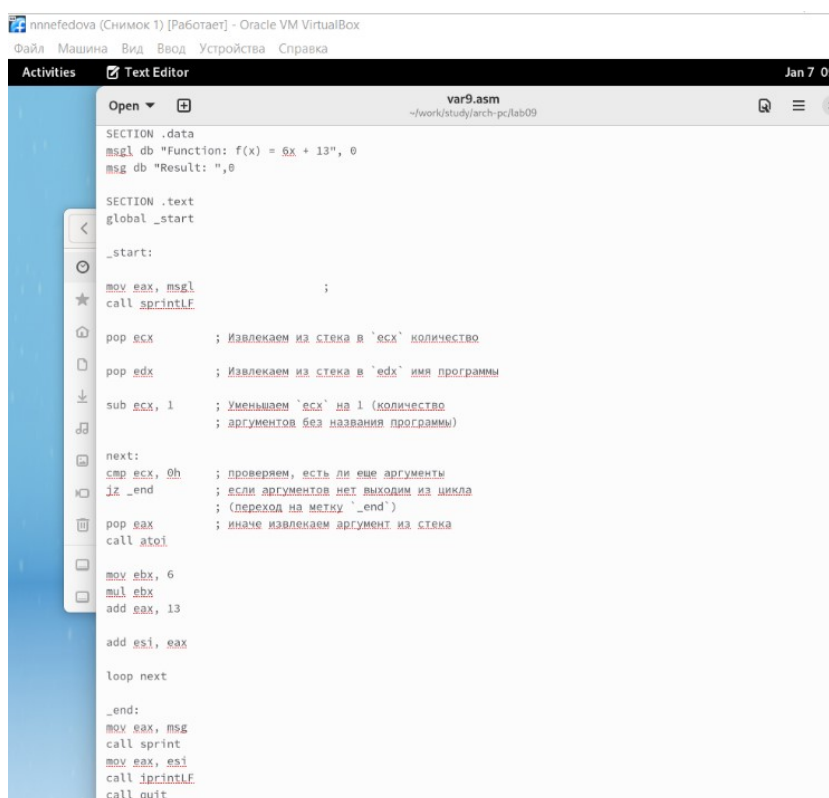


Рис. 2.15: 15

```
liveuser@localhost-live:~/work/study/arch-pc/lab09
[liveuser@localhost-live lab09]$ nasm -f elf var9.asm
var9.asm:6: error: label 'msgl' inconsistently redefined
var9.asm:5: info: label 'msgl' originally defined here
[liveuser@localhost-live lab09]$ nasm -f elf lab0-3.asm
[liveuser@localhost-live lab09]$ nasm -f elf var9.asm
var9.asm:6: error: label 'msg' inconsistently redefined
var9.asm:5: info: label 'msg' originally defined here
[liveuser@localhost-live lab09]$ nasm -f elf var9.asm
var9.asm:14: error: symbol 'sprinlLF' not defined
[liveuser@localhost-live lab09]$ nasm -f elf var9.asm
[liveuser@localhost-live lab09]$ ld -m elf_i386 -o var9 var9.o
[liveuser@localhost-live lab09]$ ./var9 1 2
Function: f(x) = 6x + 13
Result: 44
[liveuser@localhost-live lab09]$ ./var9 1 2 3
Function: f(x) = 6x + 13
Result: 75
[liveuser@localhost-live lab09]$ ./var9 5 6 9
Function: f(x) = 6x + 13
Result: 159
[liveuser@localhost-live lab09]$
```

Рис. 2.16: 16

3 Выводы

В ходе данной работы были приобретены навыки написания программ с использованием циклов и обработкой аргументов командной строки.