

**РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ**

**Факультет физико-математических и естественных наук**

**Кафедра прикладной информатики и теории вероятностей**

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ № 8**

*дисциплина:* Архитектура компьютера

Студент: Безходарнова А.В

Группа: НКАбд-01-25

**МОСКВА**

2025 г.

## СОДЕРЖАНИЕ

1.Цель работы .....	3
2. Выполнение лабораторной работы.....	4
2.1 Реализация переходов в NASM .....	4
2.2 Изучение структуры файла листинга .....	9
3. Задание для самостоятельной работы.....	12
4. Выводы .....	14

## **1.Цель работы**

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

## 2. Выполнение лабораторной работы

### 2.1 Реализация циклов в NASM

Создаю каталог для программ лабораторной работы №8, перейдя в него создаю файл lab8-1.asm:.(рис.2.1.1)

```
avbezhodarnova1@avbezhodarnova1:~$ mkdir -p ~/work/arch-pc/  
avbezhodarnova1@avbezhodarnova1:~$ mkdir ~/work/arch-pc/lab08  
avbezhodarnova1@avbezhodarnova1:~$ cd ~/work/arch-pc/lab08  
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ touch lab8-1.asm  
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ S
```

Рис. 2.1.1. Создание файла

В созданном файле ввожу программу из листинга. (рис.2.1.3)

```
%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] ; Счетчик цикла, ecx=N  
label:  
mov [N],ecx  
mov eax,[N]  
call iprintLF ; Вывод значения N  
loop label ; ecx=ecx-1 и если ecx не '0'  
call quit
```

Рис. 2.1.2 Загрузка текста из листинга

Далее запускаю исходный код. (рис.2.1.3)

```
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 1
1
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
10
9
8
7
6
5
4
3
2
1
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$
```

Рис. 2.1.3 Запуск программы

Далее я изменяю программу для получения другого результата.  
(Рис.2.1.4) и (рис.2.1.5)

```

%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:
sub ecx,1 |
mov [N],ecx
mov eax,[N]
call iprintLF

```

Рис.2.1.4 Изменение текста

```

avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
7
5
3
1
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$

```

Рис.2.1.5 Запуск программы

В варианте с `sub ecx,1` регистр `ecx` уменьшается на 1 каждый раз: сначала на 1 в команде `sub`, потом еще на 1 в команде `loop`. Число проходов не соответствует изначальному числу `N`.

Теперь добавляю команды `push` и `pop` в программу. (рис.2.1.6) и (рис.2.1.7)

```

%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
loop label
pop ecx
call quit

```

Рис.2.1.6 Изменение текста

```
0
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$
```

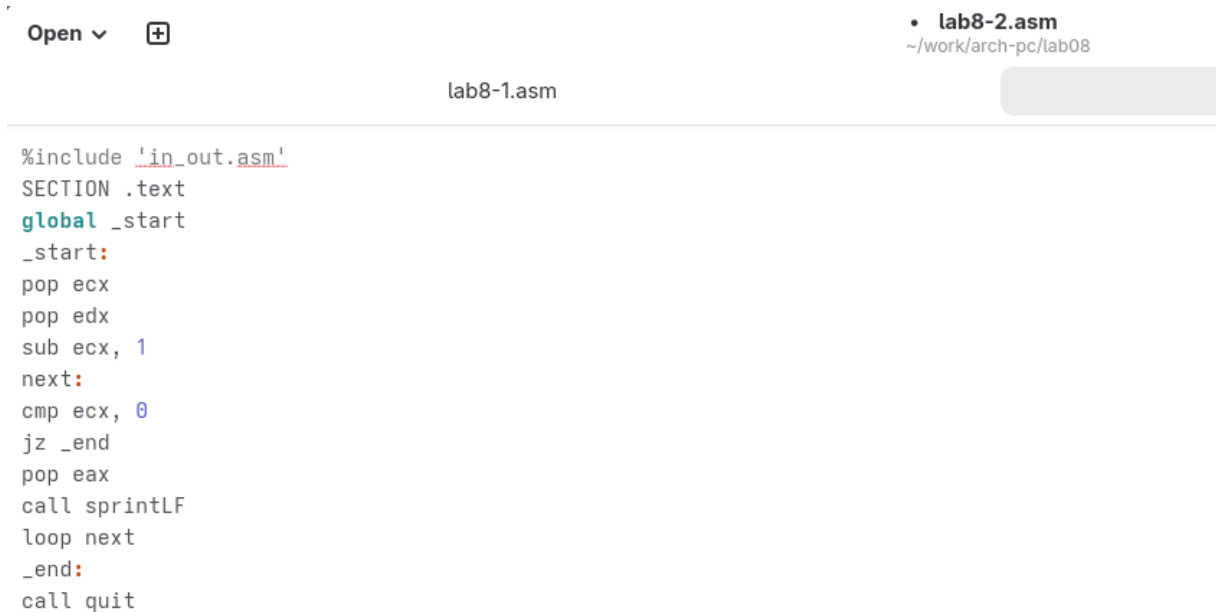
Рис.2.1.7 Вывод результата

Теперь число проходов цикла совпадает с N, но произошло смещение выводимых чисел на -1.



## 2.2 Обработка аргументов командной строки

Создаю новый файл и ввожу в него код из листинга. (рис. 2.2.1)




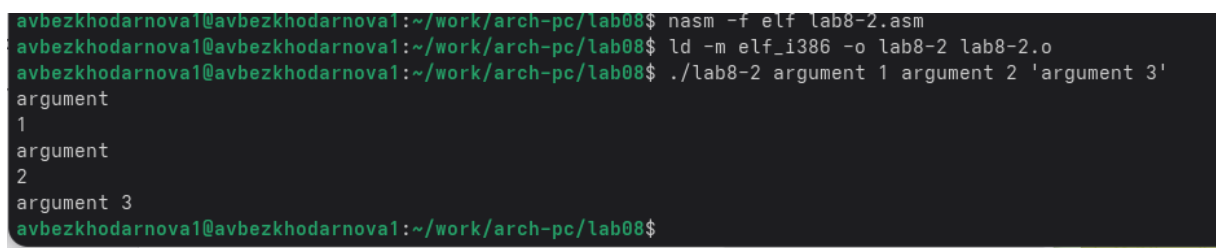
```
Open ▾  lab8-2.asm  
~/work/arch-pc/lab08  
lab8-1.asm  
%include 'in_out.asm'  
SECTION .text  
global _start  
_start:  
pop ecx  
pop edx  
sub ecx, 1  
next:  
cmp ecx, 0  
jz _end  
pop eax  
call printf  
loop next  
_end:  
call quit
```

Рис.2.2.1 Ввод кода из листинга.

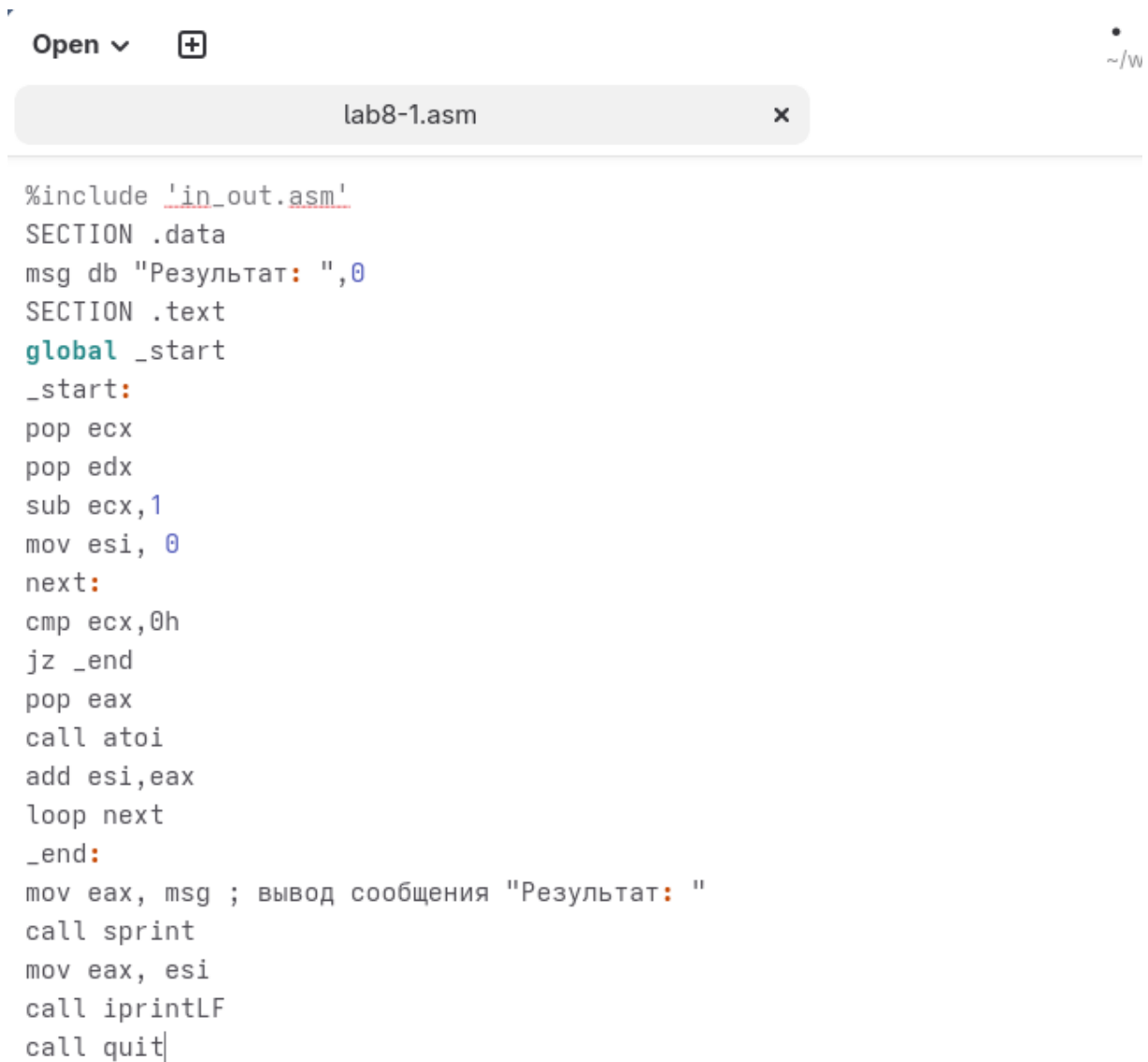
Запускаю программу и задаю аргументы. (Рис.2.2.2)



```
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ nasm -f elf lab8-2.asm  
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-2 lab8-2.o  
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ./lab8-2 argument 1 argument 2 'argument 3'  
argument  
1  
argument  
2  
argument 3  
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$
```

Рис. 2.2.2 Запуск программы

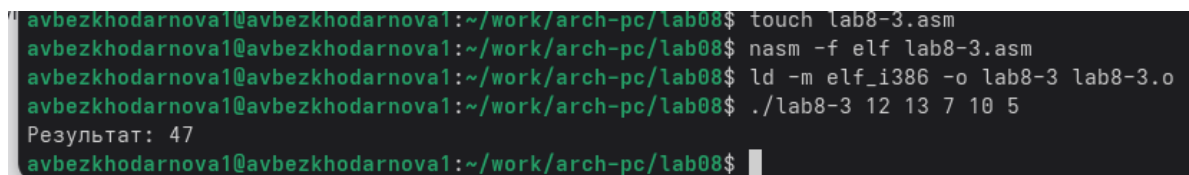
Создаю новый файл и загружаю туда текст из листинга. (Рис.2.2.3)



```
%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 ; вывод сообщения "Результат: "
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис. 2.2.3 Изменение кода.

Запускаю программу, ввожу туда заданные аргументы и на выходе получаю их сумму. (Рис.2.2.4)



```
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab08$ touch lab8-3.asm
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab08$ ./lab8-3 12 13 7 10 5
Результат: 47
avbezkhodarnova1@avbezkhodarnova1:~/work/arch-pc/lab08$
```

Рис.2.2.4 Запуск программы.

Далее я изменяю текст программы так, чтобы в результате получить умножение заданных аргументов. (рис.2.2.5)

```

#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
mul esi
mov esi, eax
loop next
_end:
mov eax, msg ; вывод сообщения "Результат: "
call sprint
mov eax, esi
call iprintLF
call quit

```

Рис.2.2.5 Изменение текста листинга.

Запускаю программу и убеждаюсь в том, что теперь она выводит результат произведения. (Рис.2.2.6)

```

avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ nasm -f elf lab8-3.asm
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-3 lab8-3.o
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ./lab8-3 5 6 7
Результат: 210
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$

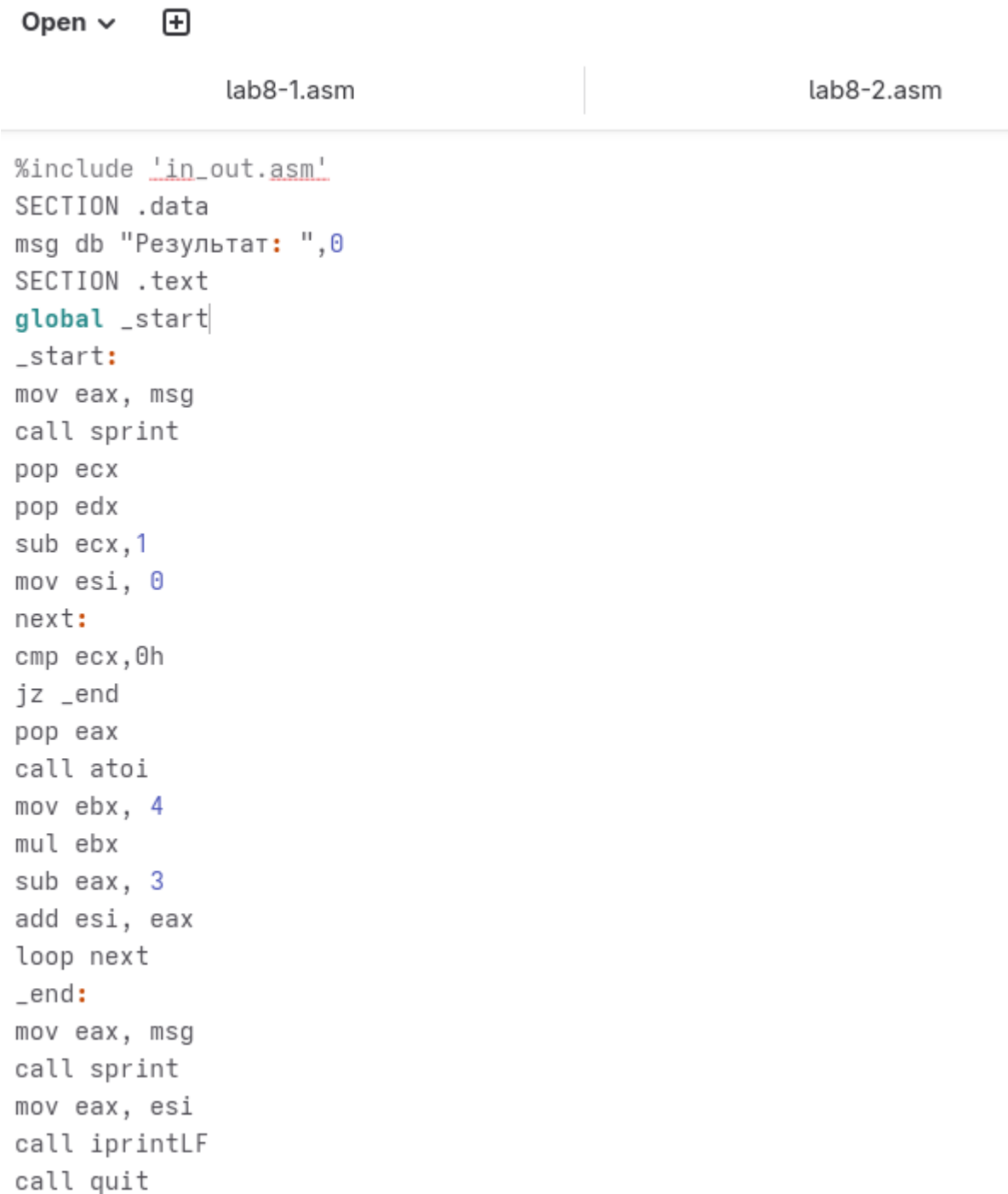
```

Рис.2.2.6 Запуск программы

### 3. Задание для самостоятельной работы

Из предложенных вариантов, я выбираю 6, как мне и вышло в 6 лабораторной работе, где  $f(x) = 4x - 3$

Создаю новый файл и пишу в нем программу. (рис.3.1) и (рис.3.2)




```
Open ▾ 
lab8-1.asm | lab8-2.asm
%include 'in_out.asm'
SECTION .data
msg db "Результат: ",0
SECTION .text
global _start
_start:
mov eax, msg
call sprint
pop ecx
pop edx
sub ecx, 1
mov esi, 0
next:
cmp ecx, 0h
jz _end
pop eax
call atoi
mov ebx, 4
mul ebx
sub eax, 3
add esi, eax
loop next
_end:
mov eax, msg
call sprint
mov eax, esi
call iprintLF
call quit
```

Рис.3.1 Редактирование файла.

```
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$ ./lab8-4 2 3 4
Результат: Результат: 27
avbezhodarnova1@avbezhodarnova1:~/work/arch-pc/lab08$
```

Рис.3.2 Запуск программы.

#### **4. Выводы**

В ходе выполнения данной лабораторной работы я приобрела навыки написания программ с использованием циклов и также научилась обрабатывать аргументы командной строки.