

# 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: Гасанова Шакира Чингизовна

Группа: НКАбд-05-24

МОСКВА

2024 г.

# Содержание

<b>1 Цель работы .....</b>	<b>3</b>
<b>2 Задание .....</b>	<b>4</b>
<b>3 Теоретическое введение .....</b>	<b>5</b>
<b>4 Выполнение лабораторной работы</b>	
4.1 Реализация циклов в NASM.....	6
4.2 Обработка аргументов командной строки.....	12
4.3 Выполнение заданий для самостоятельной работы.....	18
<b>5 Выводы .....</b>	<b>22</b>
<b>6 Источники .....</b>	<b>23</b>

## **1 Цель работы**

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

## **2 Задания**

1. Реализация циклов в NASM.
2. Обработка аргументов командной строки.
3. Самостоятельное написание программы по материалам лабораторной работы.

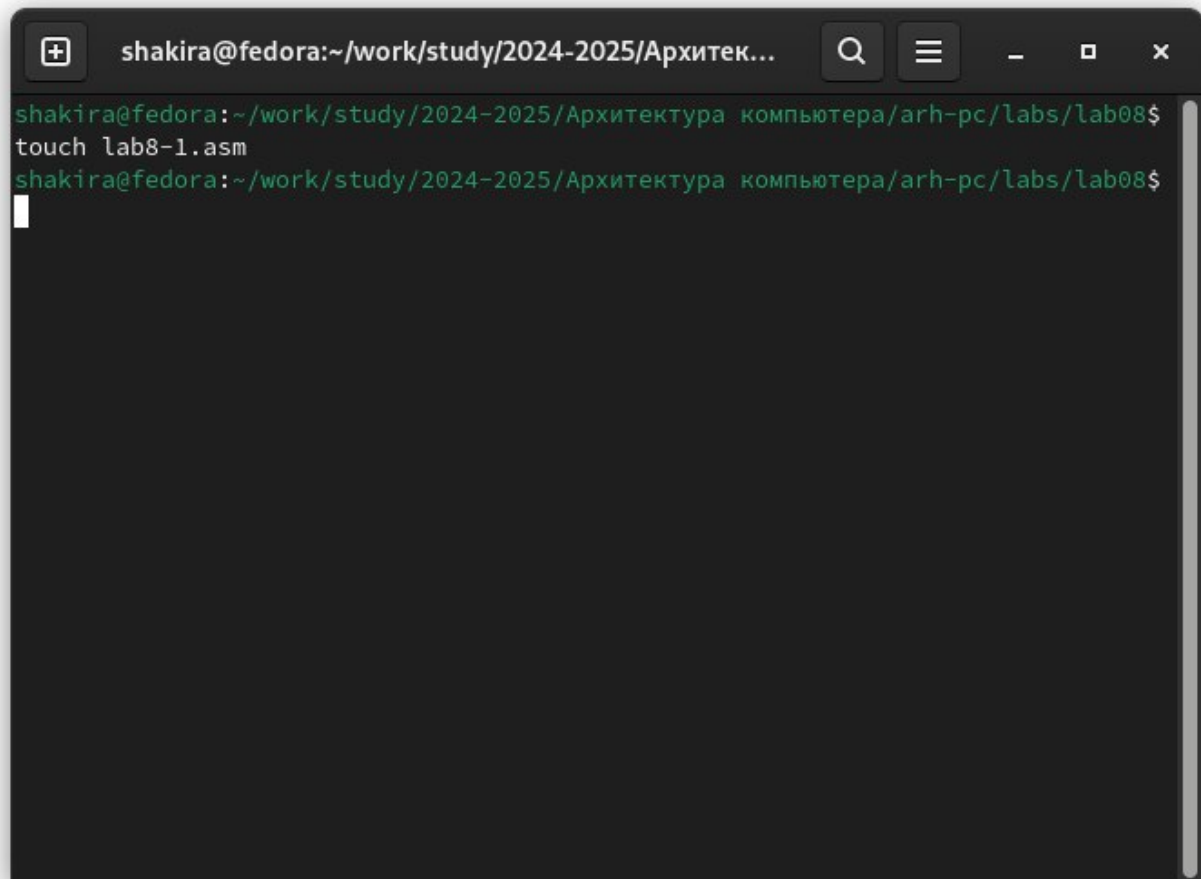
### **3 Теоретическое введение**

Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нём выделяется память для локальных переменных и могут временно храниться значения регистров.

## 4 Выполнение лабораторной работы

### 4.1 Реализация циклов в NASM

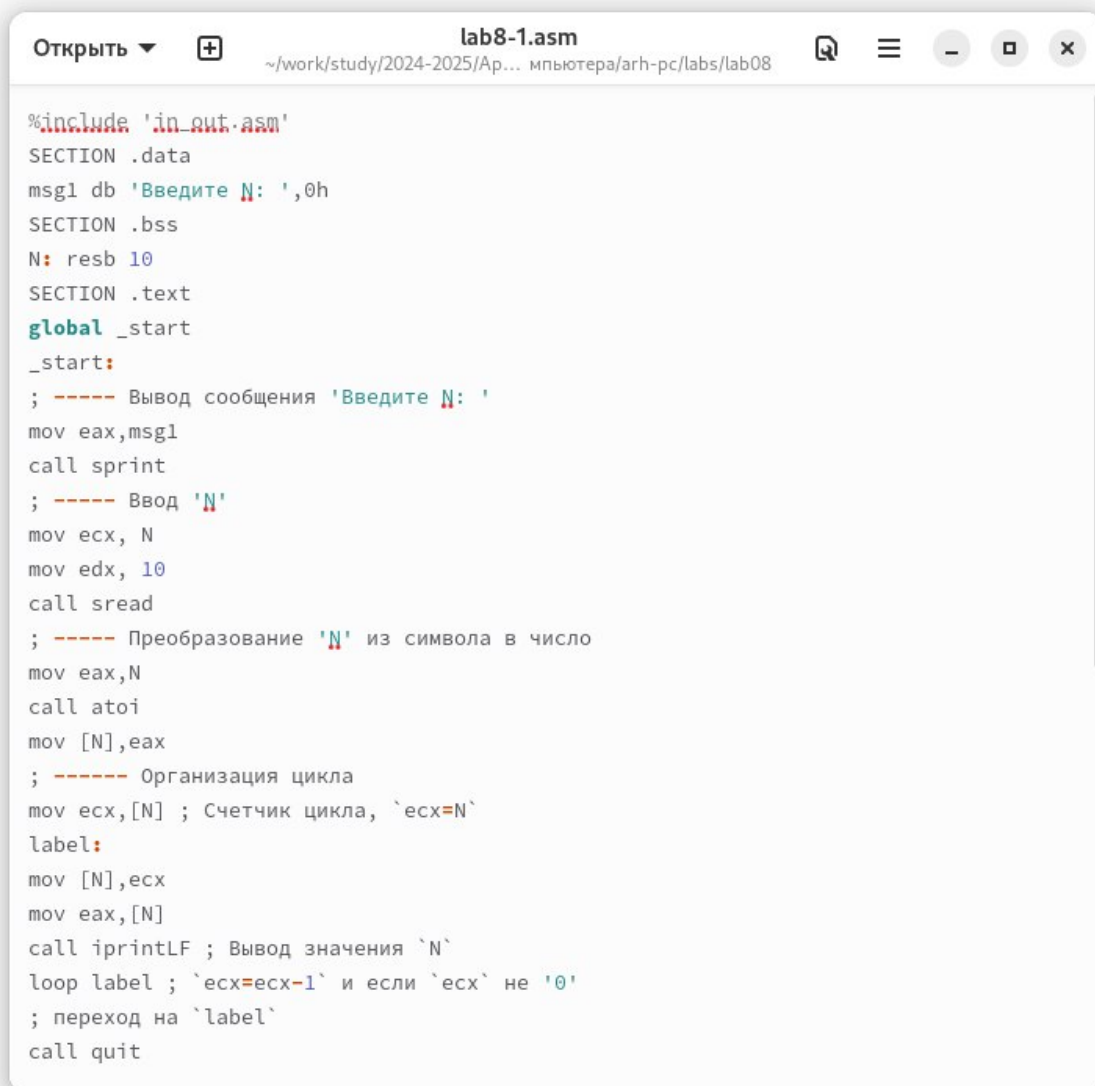
Создаю каталог для программ лабораторной работы №8 и файл для написания программы (рис. 1).

A screenshot of a terminal window with a dark background. The window title bar shows the user 'shakira@fedora' and the current directory '~/work/study/2024-2025/Архитек...'. The terminal displays the following commands and prompts: 'shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08\$ touch lab8-1.asm' followed by 'shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08\$' and a blank line with a cursor.

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$ touch lab8-1.asm
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$
```

Рис.1 Создание файла

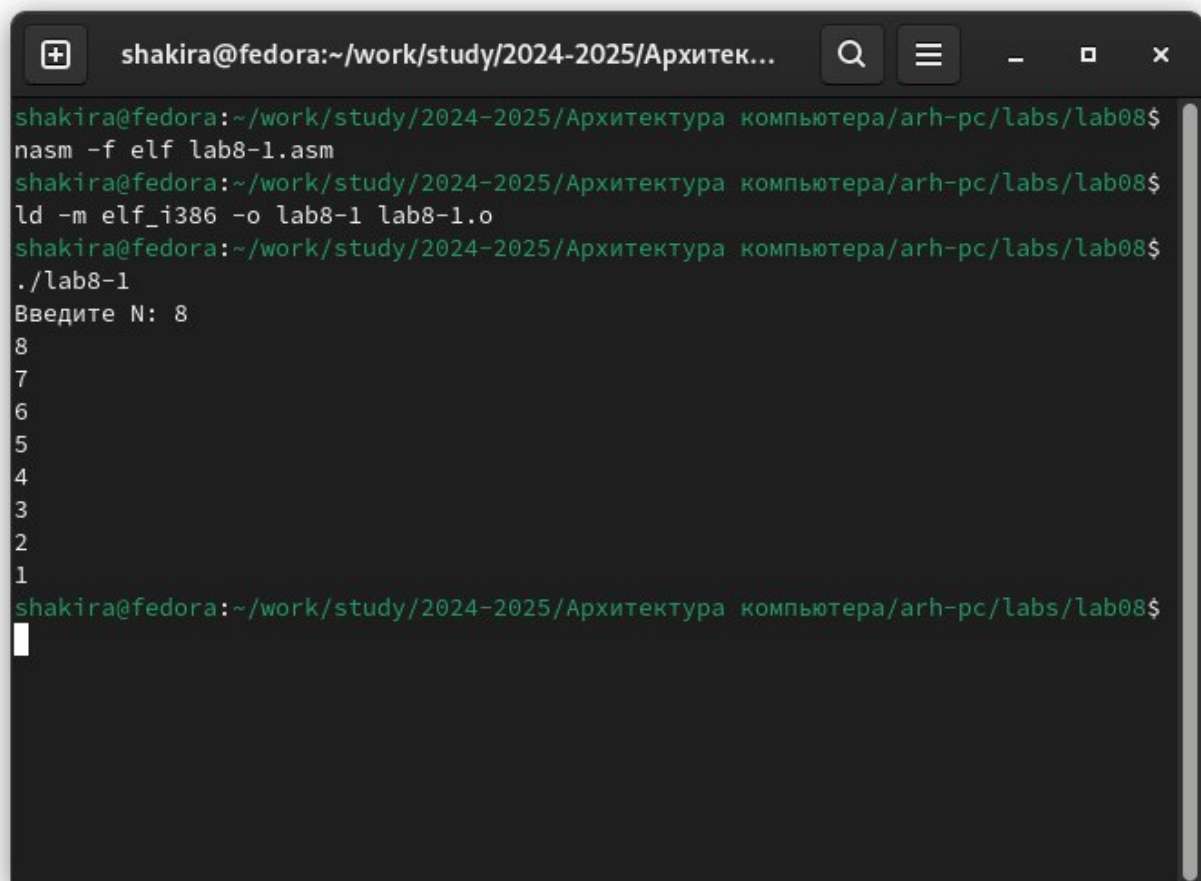
Копирую в созданный файл программу из листинга (рис. 2).



```
%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
; ----- Организация цикла
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`
; переход на `label`
call quit
```

Рис.2 Копирование программы из листинга

Запускаю программу, она показывает работу циклов в NASM (рис. 3).

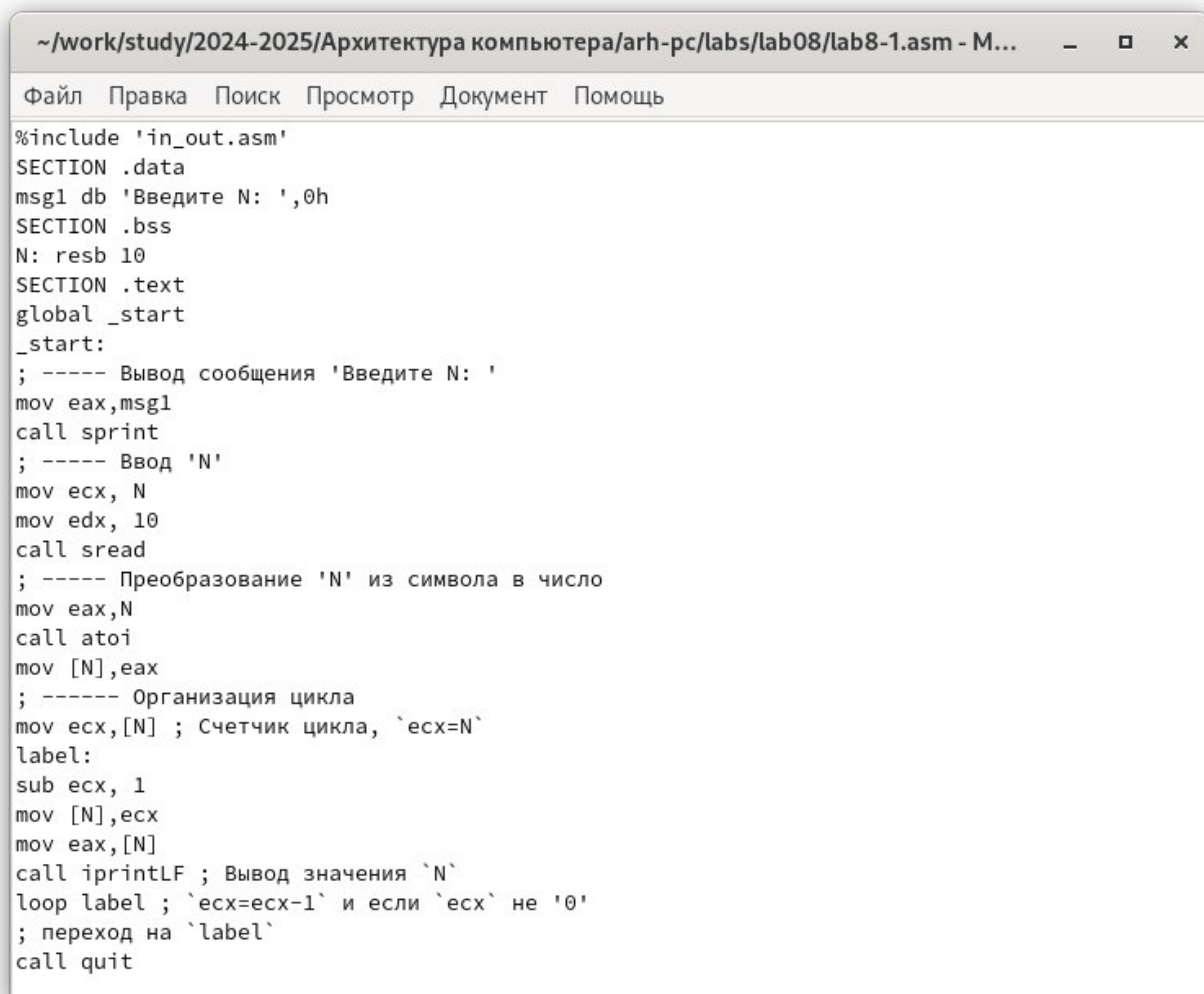
A terminal window with a dark background and green text. The window title is "shakira@fedora:~/work/study/2024-2025/Архитек...". The terminal shows the following commands and output:

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
nasm -f elf lab8-1.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
ld -m elf_i386 -o lab8-1 lab8-1.o  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
./lab8-1  
Введите N: 8  
8  
7  
6  
5  
4  
3  
2  
1  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$
```

Рис.3 Запуск исполняемого файла

Заменяю программу изначальную так, что в теле цикла я изменяю значение регистра есх (рис. 4).



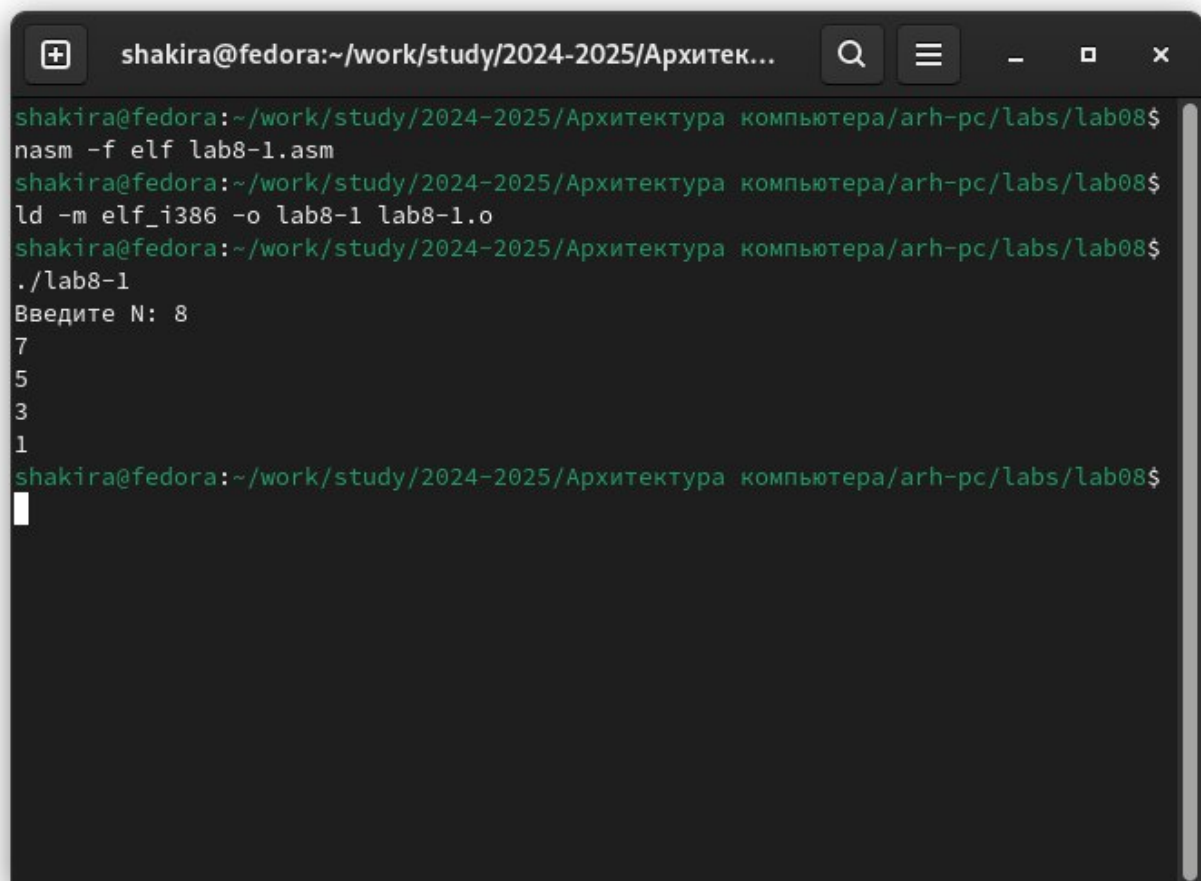


```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08/lab8-1.asm - M...
Файл  Правка  Поиск  Просмотр  Документ  Помощь

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

Рис.4 Изменение программы

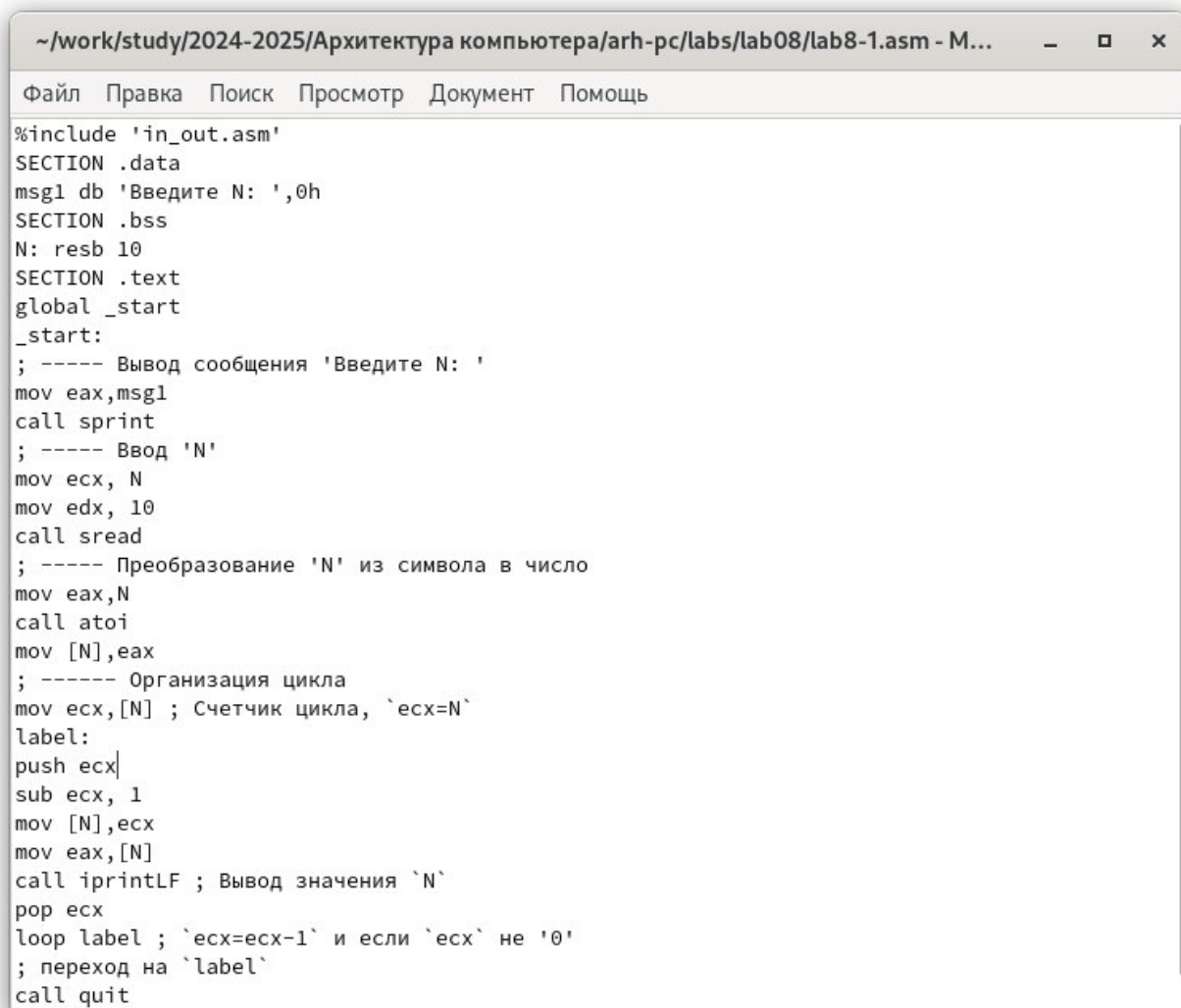
Из-за того, что теперь регистр ecx на каждой итерации уменьшается на 2 значения, количество итераций уменьшается вдвое (рис. 5).



```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
nasm -f elf lab8-1.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
ld -m elf_i386 -o lab8-1 lab8-1.o  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
./lab8-1  
Введите N: 8  
7  
5  
3  
1  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
█
```

Рис.5 Запуск изменённой программы

Добавляю команды push и pop в программу (рис. 6).

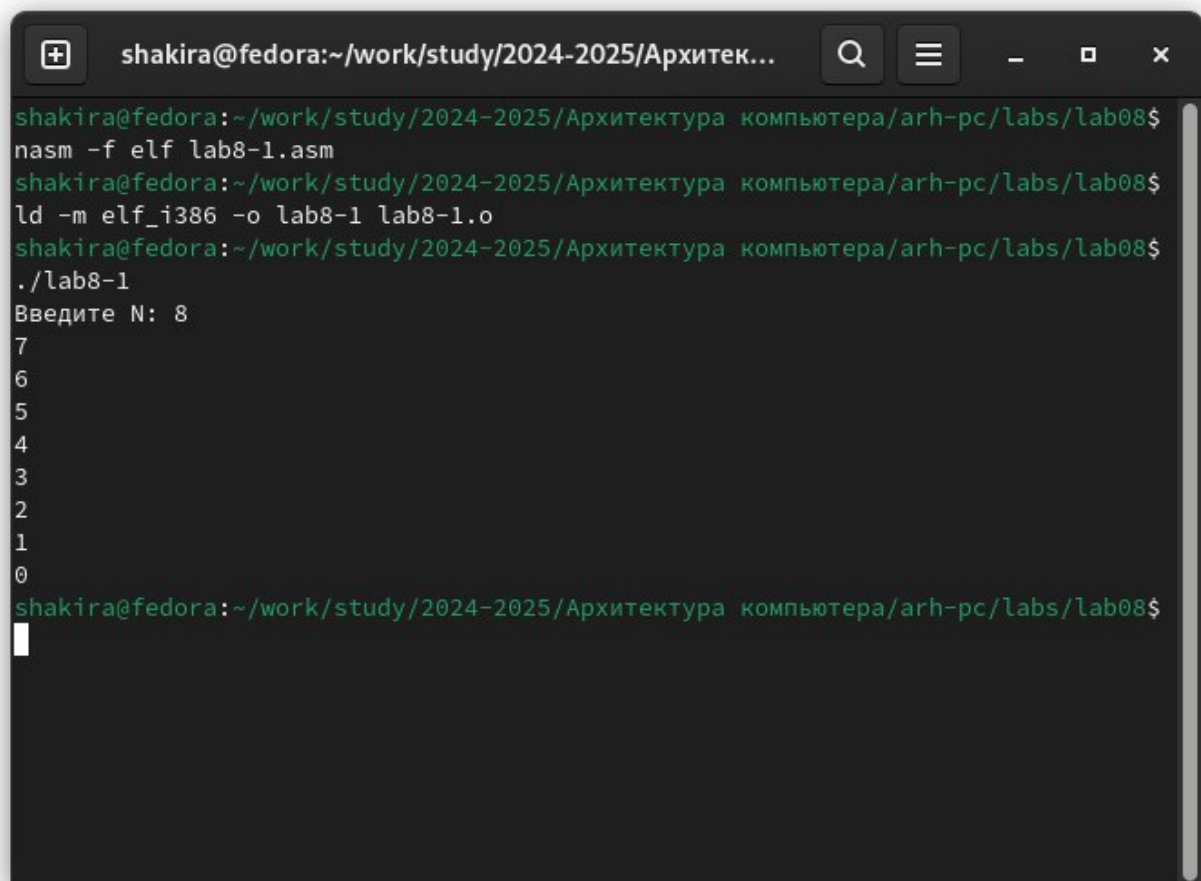


```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08/lab8-1.asm - M...
Файл  Правка  Поиск  Просмотр  Документ  Помощь

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

Рис.6 Добавление push и pop в цикл программы

Теперь количество итераций совпадает введенному N, но произошло смещение выводимых чисел на -1 (рис. 7).

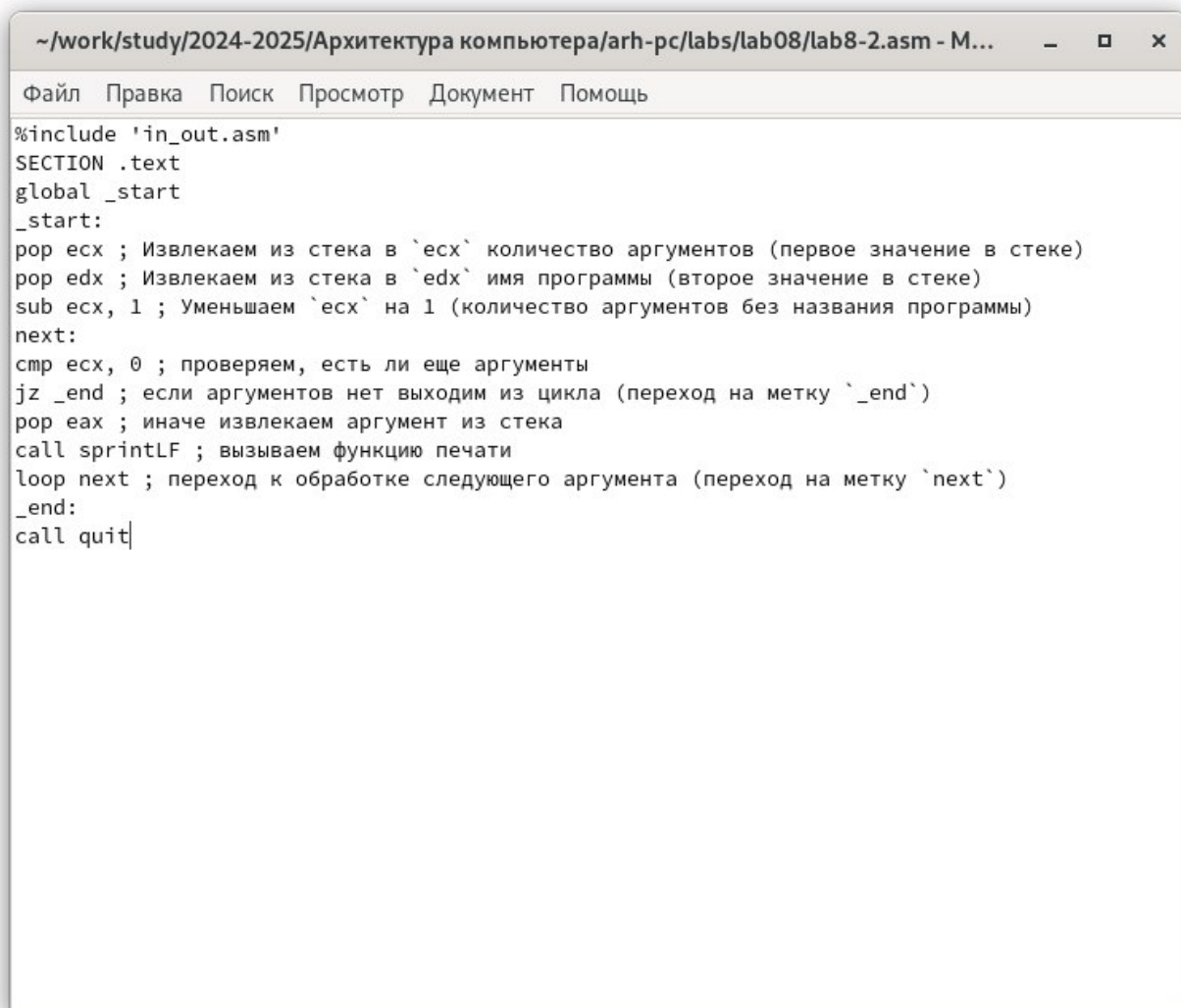


```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
nasm -f elf lab8-1.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
ld -m elf_i386 -o lab8-1 lab8-1.o  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
./lab8-1  
Введите N: 8  
7  
6  
5  
4  
3  
2  
1  
0  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$
```

Рис.7 Запуск изменённой программы

## 4.2 Обработка аргументов командной строки

Создаю новый файл для программы и копирую в него код из следующего листинга (рис. 8).

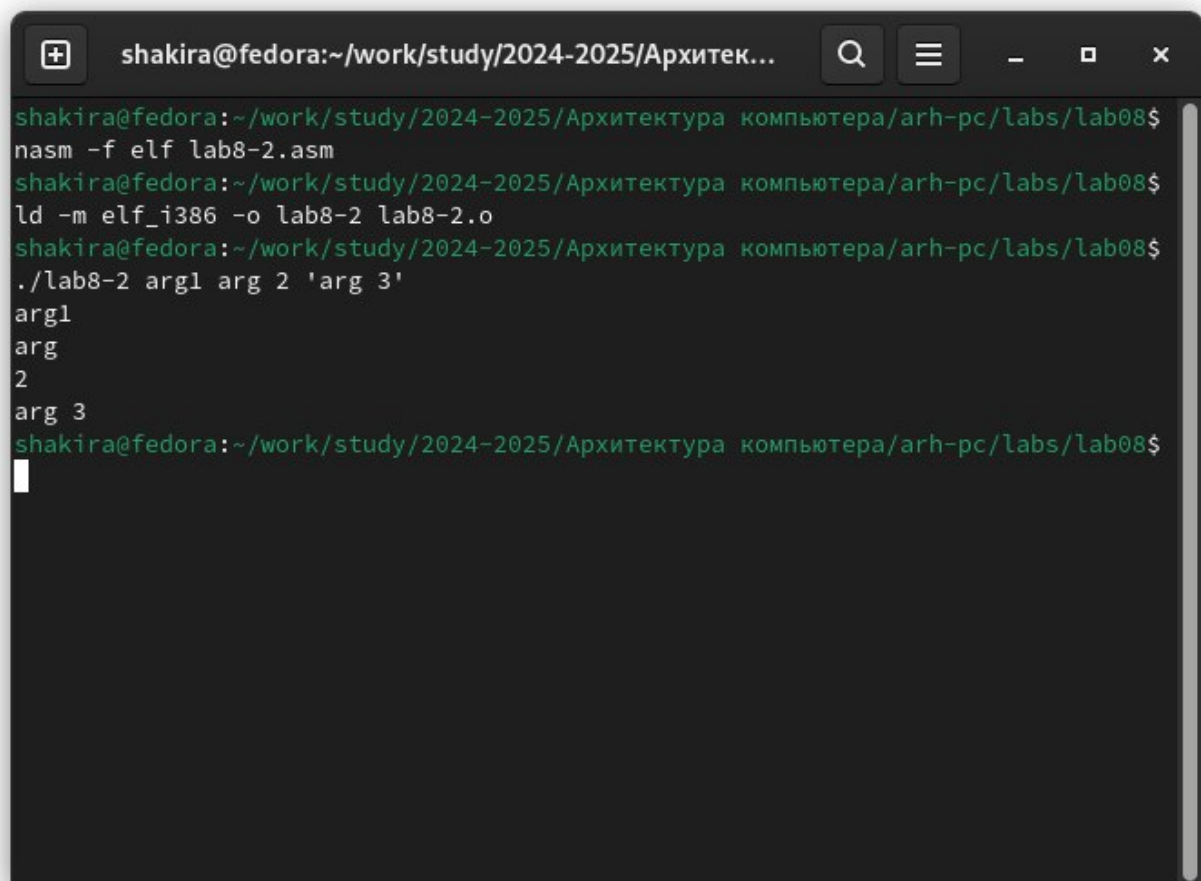


```
~/.work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08/lab8-2.asm - M...
Файл  Правка  Поиск  Просмотр  Документ  Помощь

%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 sprintf ; вызываем функцию печати
loop next ; переход к обработке следующего аргумента (переход на метку `next`)
_end:
call quit
```

Рис.8 Копирование программы из листинга

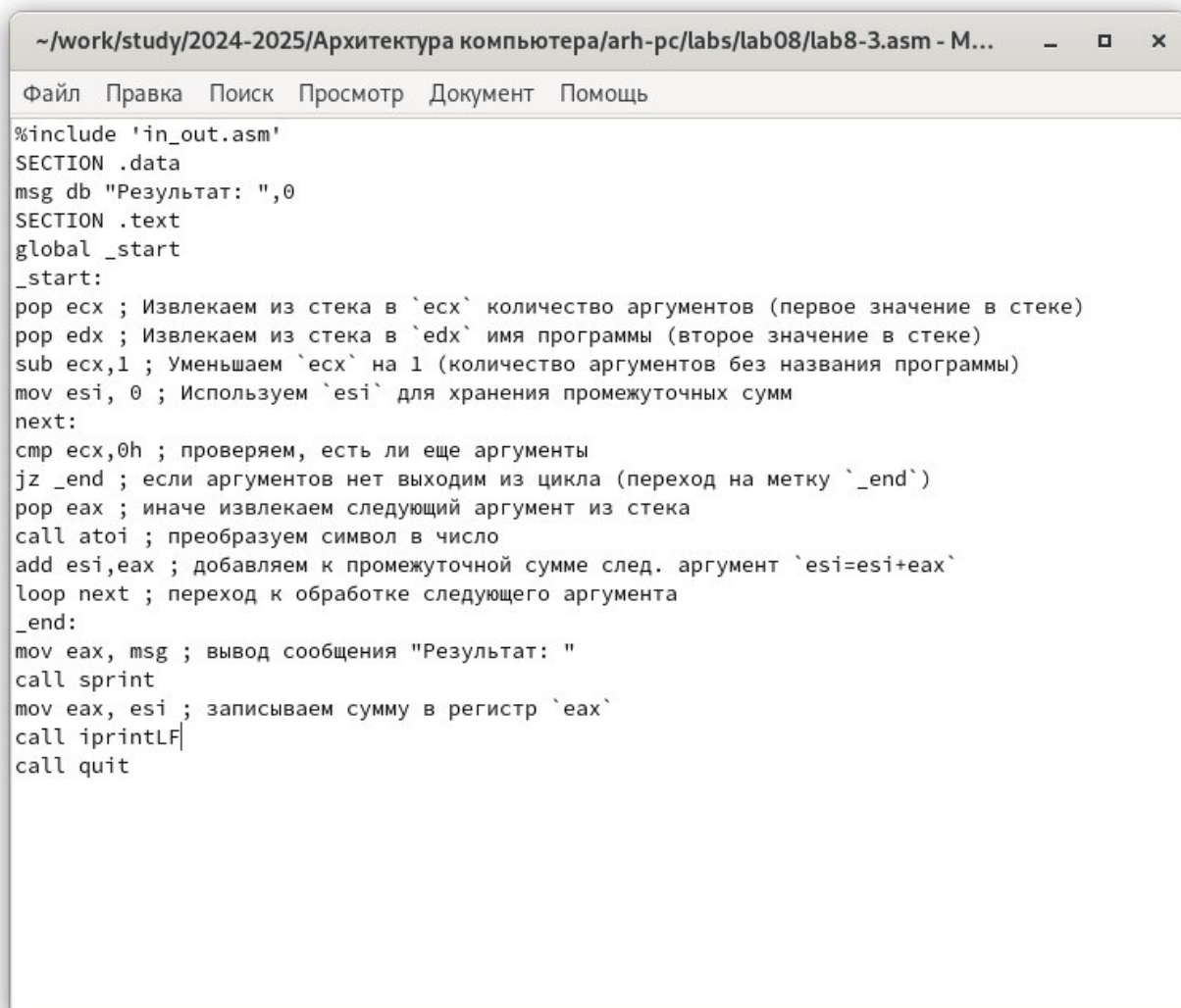
Компилирую программу и запускаю, указав аргументы. Программой было обработано то же количество аргументов, что и было введено (рис. 9).

A terminal window with a dark background and light green text. The window title is "shakira@fedora:~/work/study/2024-2025/Архитек...". The terminal shows the following commands and output:

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
nasm -f elf lab8-2.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
ld -m elf_i386 -o lab8-2 lab8-2.o  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
./lab8-2 arg1 arg 2 'arg 3'  
arg1  
arg  
2  
arg 3  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
█
```

Рис.9 Запуск исполняемого файла

Создаю новый файл для программы и копирую в него код из третьего листинга (рис. 10).

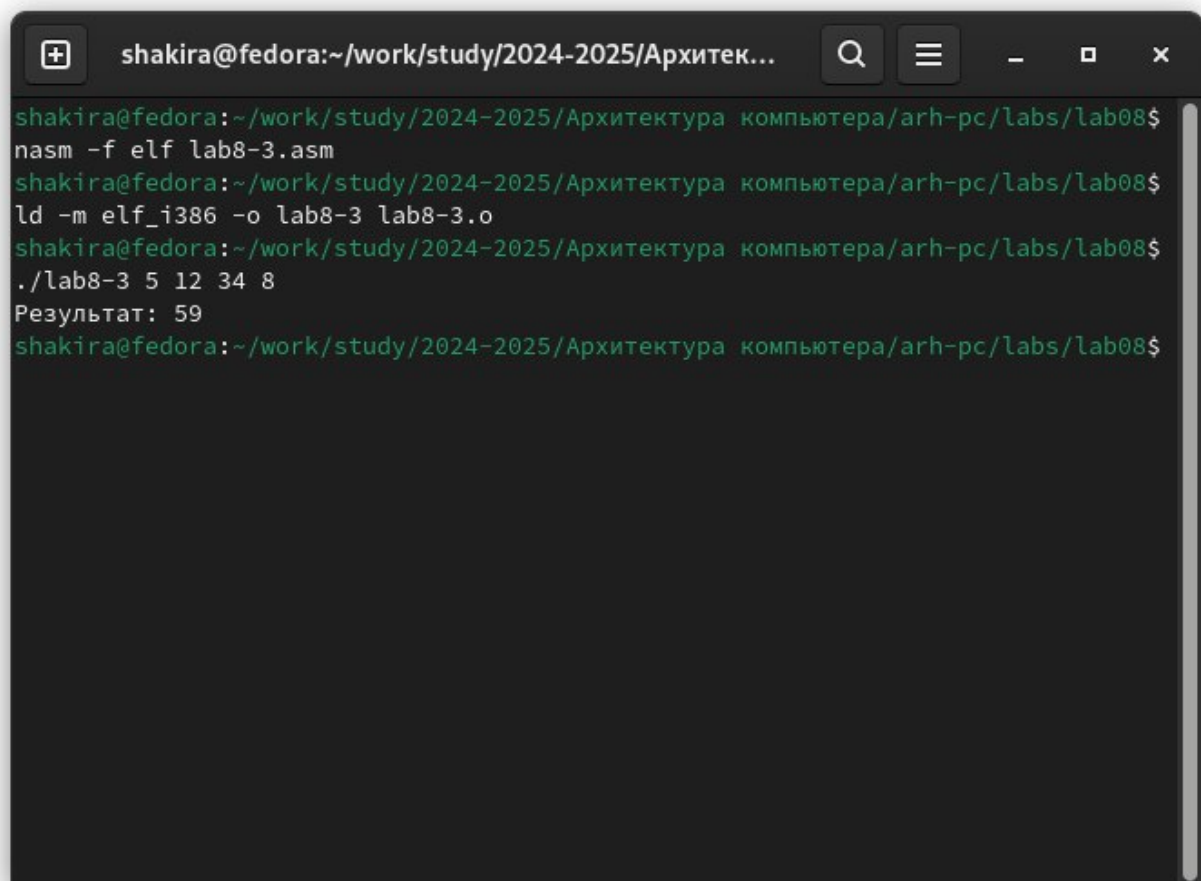


```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08/lab8-3.asm - M...
Файл Правка Поиск Просмотр Документ Помощь

%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
call quit
```

Рис.10 Копирование программы из листинга

Компилирую программу и запускаю, указав в качестве аргументов некоторые числа, программа их складывает (рис. 11).

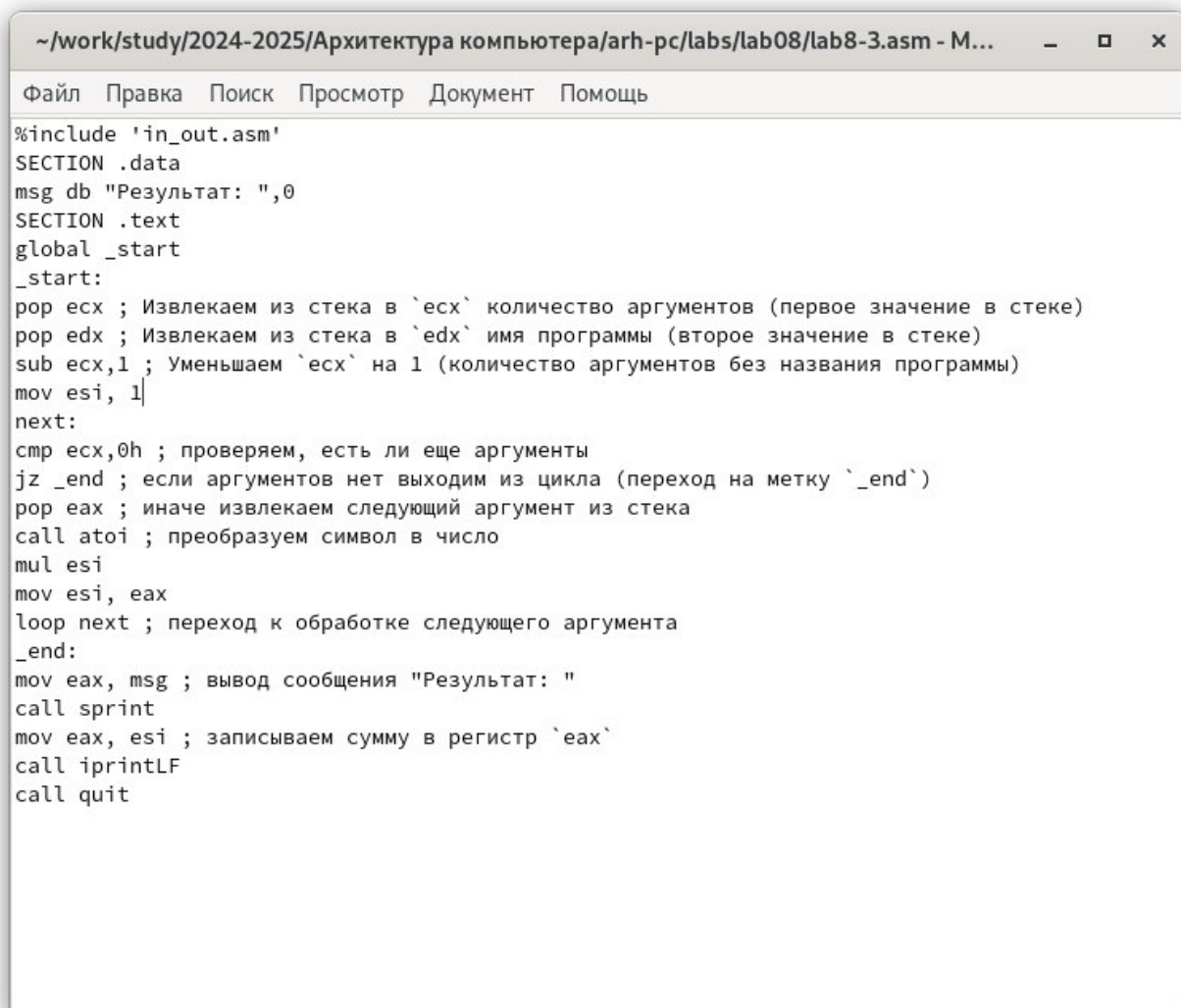
A terminal window with a dark background and light green text. The window title is "shakira@fedora:~/work/study/2024-2025/Архитек...". The terminal shows the following commands and output:

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
nasm -f elf lab8-3.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
ld -m elf_i386 -o lab8-3 lab8-3.o  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
./lab8-3 5 12 34 8  
Результат: 59  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$
```

Рис.11 Запуск исполняемого файла

Изменяю поведение программы так, чтобы указанные аргументы она умножала, а не складывала (рис. 12).



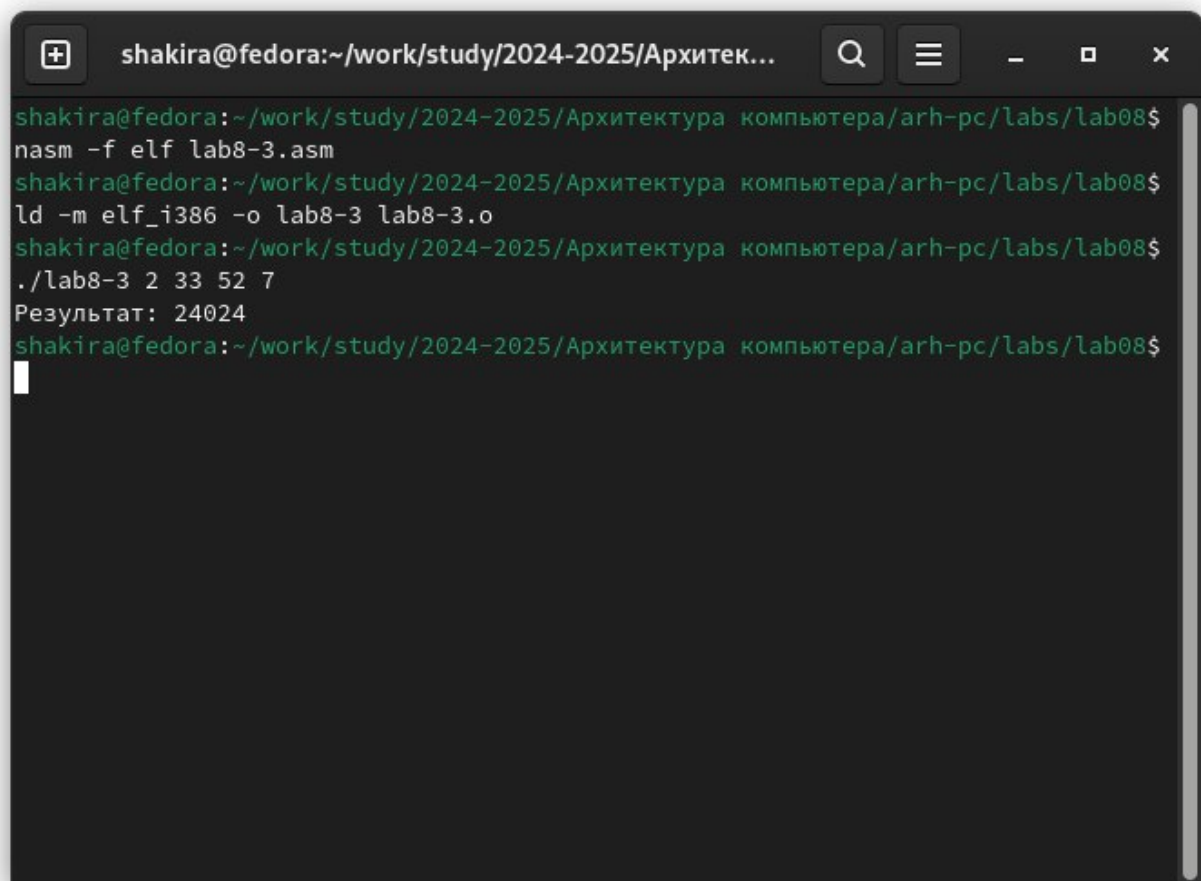


```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08/lab8-3.asm - M...
Файл  Правка  Поиск  Просмотр  Документ  Помощь

%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, 1
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
mul esi
mov 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
call quit
```

Рис.12 Изменение программы

Программа действительно теперь умножает данные на вход числа (рис. 13).

A terminal window with a dark background and light green text. The window title is 'shakira@fedora:~/work/study/2024-2025/Архитек...'. The terminal shows the following commands and output:

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
nasm -f elf lab8-3.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
ld -m elf_i386 -o lab8-3 lab8-3.o  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
./lab8-3 2 33 52 7  
Результат: 24024  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$
```

Рис.13 Запуск изменённой программы

### 4.3 Выполнение заданий для самостоятельной работы

Пишу программу, которая будет находить сумму значений для функции  $f(x) = 2(x - 1)$ , которая соответствует 4 варианту (рис.14).

```
~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08/8-4.asm - Mouse...
Файл  Правка  Поиск  Просмотр  Документ  Помощь

%include 'in_out.asm'
SECTION .data
msg_func db "Функция: f(x) = 2(x - 1)",0
msg_res db "Результат: ",0
SECTION .text
global _start
_start:
mov eax, msg_func
call sprintLF
pop ecx
pop edx
sub ecx,1
mov esi, 0
next:
cmp ecx,0h
jz _end
pop eax
call atoi
sub eax, 1
mov ebx, 2
mul ebx
add esi, eax
loop next
_end:
mov eax, msg_res
call sprint
mov eax, esi
call iprintLF
call quit
```

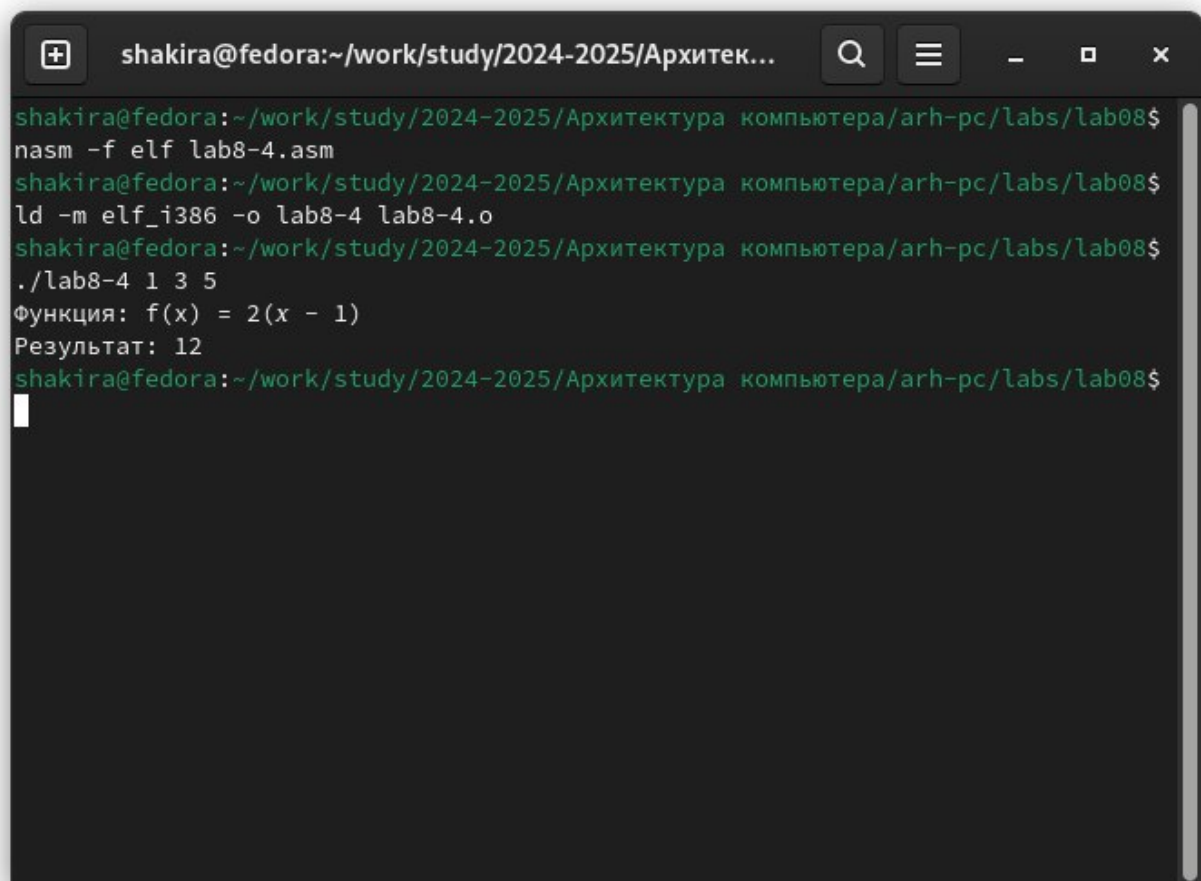
Рис.14 Написание программы

Код программы:

```
%include 'in_out.asm'
SECTION .data
msg_func db "Функция: f(x) = 2(x - 1)",0
msg_res db "Результат: ",0
SECTION .text
global _start
_start:
mov eax, msg_func
call sprintLF
pop ecx
pop edx
```

```
sub ecx,1
mov esi, 0
next:
cmp ecx,0h
jz _end
pop eax
call atoi
sub eax, 1
mov ebx, 2
mul ebx
add esi, eax
loop next
_end:
mov eax, msg_res
call sprint
mov eax, esi
call iprintLF
call quit
```

Проверяю работу программы, указав в качестве аргумента несколько чисел (рис. 15).



A terminal window with a dark background and light green text. The window title is "shakira@fedora:~/work/study/2024-2025/Архитек...". The terminal shows the following commands and output:

```
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
nasm -f elf lab8-4.asm  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
ld -m elf_i386 -o lab8-4 lab8-4.o  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$  
./lab8-4 1 3 5  
Функция:  $f(x) = 2(x - 1)$   
Результат: 12  
shakira@fedora:~/work/study/2024-2025/Архитектура компьютера/arh-pc/labs/lab08$
```

Рис.15 Запуск исполняемого файла

## **5 Выводы**

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

## **6 Источники**

1. [Архитектура ЭВМ \(rudn.ru\)](http://rudn.ru)