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: Сухова Арина

Группа: НПИбд-02-25

МОСКВА

2025 г.

1. Цель работы

Изучить работу циклов и обработку аргументов командной строки.

2. Задание

Написать программы с использованием циклов и обработкой аргументов командной строки.

3. Выполнение лабораторной работы

3.1 Реализация циклов в NASM

Создаем каталог для программ ЛБ8, и в нем создаем файл (рис. 1).

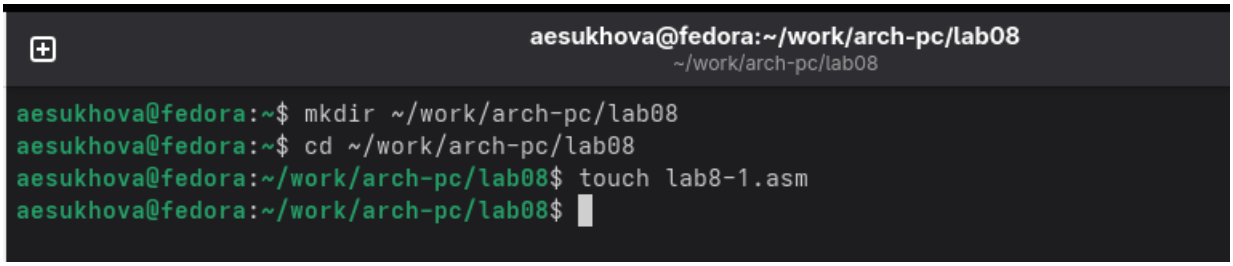

Рис.1 Создаем каталог с помощью команды mkdir и файл с помощью команды touch

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 8.1 (рис. 2).

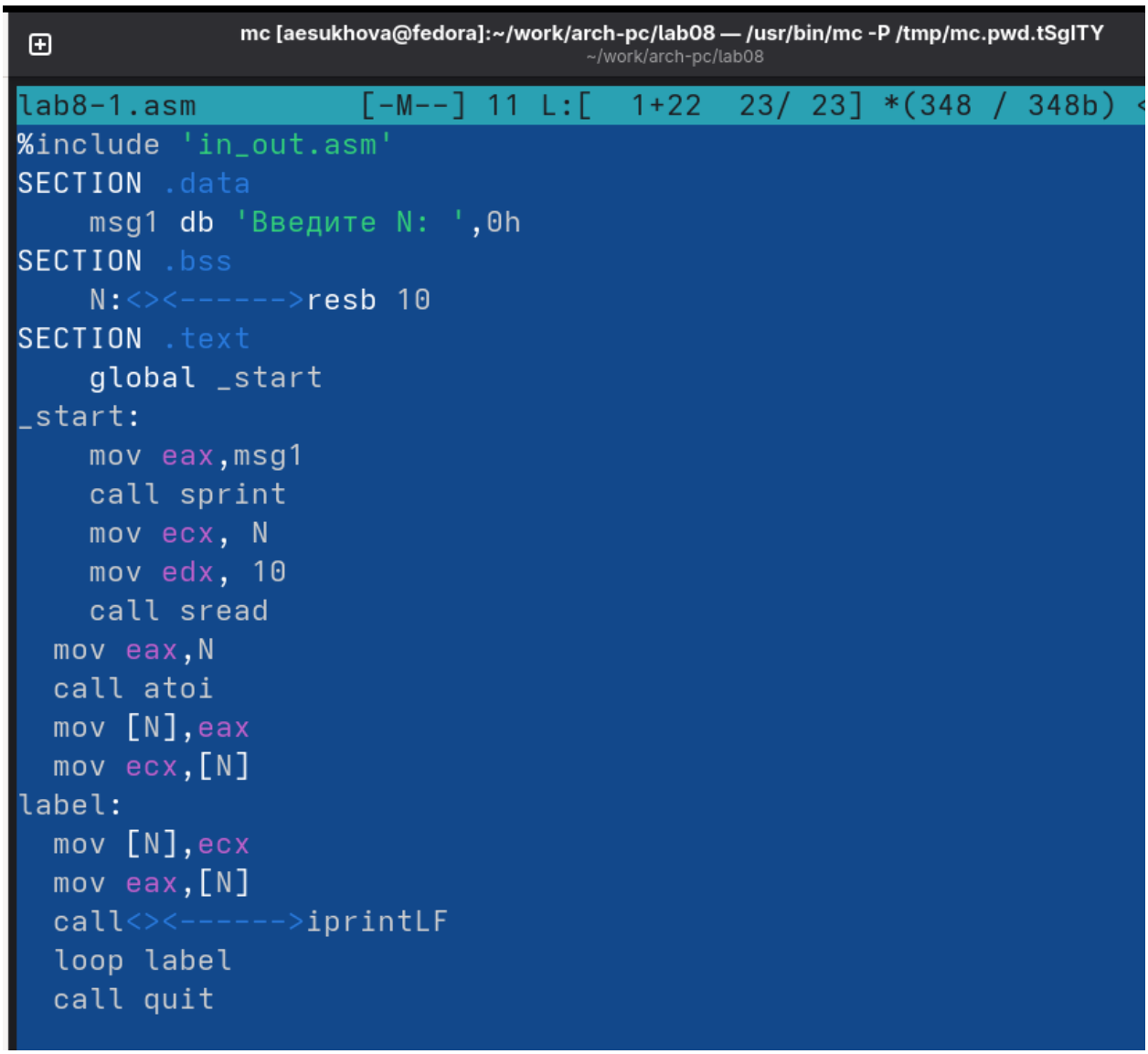

Рис.2 Заполняем файл

Создаем исполняемый файл и запускаем его (рис. 3).

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

Рис.3 Запускаем файл и проверяем его работу

Снова открываем файл для редактирования и изменяем его, добавив изменение значения регистра в цикле (рис. 4).

```
_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
  loop label
```

Рис.4 Изменяем файл

Создаем исполняемый файл и запускаем его (рис. 5).

```
aesukhova@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
aesukhova@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
aesukhova@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
7
5
3
1
Ошибка сегментирования (образ памяти сброшен на диск)
aesukhova@fedora:~/work/arch-pc/lab08$
```


Рис.5 Запускаем файл и смотрим на его работу

Регистр ecx принимает значения 9,7,5,3,1(на вход подается число 10, в цикле label данный регистр уменьшается на 2 командой sub и loop).

Число проходов цикла не соответсвует числу N, так как уменьшается на 2.

Снова открываем файл для редактирования и изменяем его, чтобы все корректно работало (рис. 6).

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

Рис.6 Редактируем файл

Создаем исполняемый файл и запускаем его (рис. 7).

```
aesukhova@fedora:~/work/arch-pc/lab08$ nasm -f elf lab8-1.asm
aesukhova@fedora:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-1 lab8-1.o
aesukhova@fedora:~/work/arch-pc/lab08$ ./lab8-1
Введите N: 10
9
8
7
6
5
4
3
2
1
0
Ошибка сегментирования (образ памяти сброшен на диск)
aesukhova@fedora:~/work/arch-pc/lab08$
```


Рис.7 Проверяем, сошелся ли наш вывод с данным в условии выводом

В данном случае число проходов цикла равна числу N.

3.2 Обработка аргументов командной строки

Создаем новый файл (рис. 8).

```
aesukhova@fedora:~/work/arch-pc/lab08$ touch lab8-2.asm
aesukhova@fedora:~/work/arch-pc/lab08$
```


Рис.8 Создаем файл командой touch

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 8.2 (рис. 9).

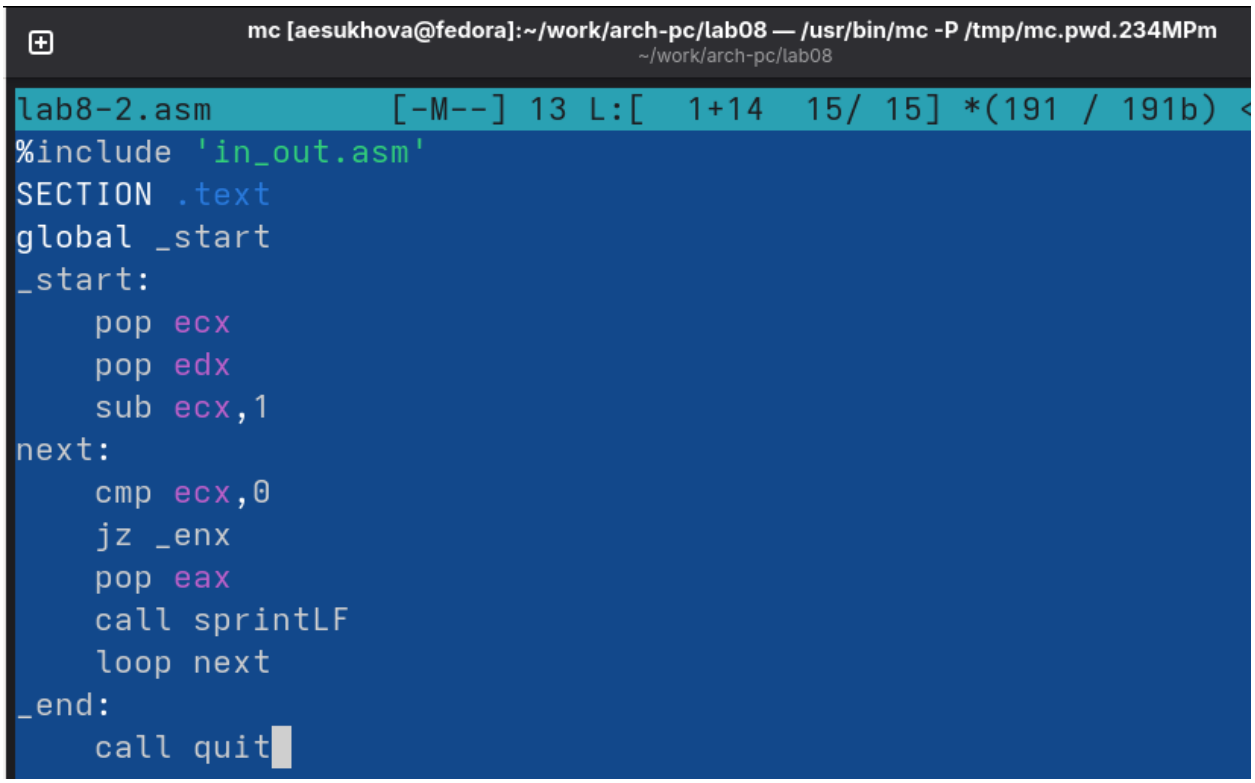

Рис.9 Заполняем файл

Создаем исполняемый файл и проверяем его работу, указав аргументы (рис. 10).

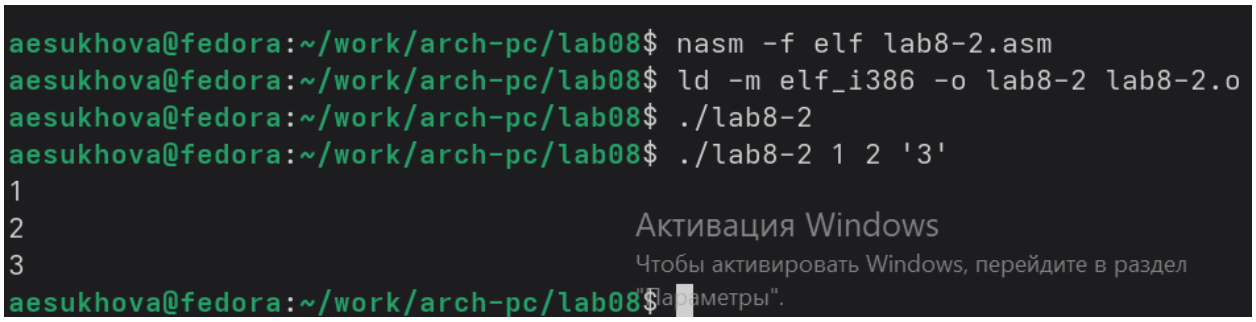

Рис.10 Смотрим на работу программ

Програмой было обработано 3 аргумента.

Создаем новый файл lab8-3.asm (рис. 11).

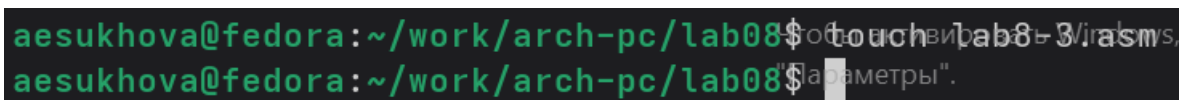

Рис.11 Создаем файл командой touch

Открываем файл и заполняем его в соответствии с листингом 8.3 (рис. 12).

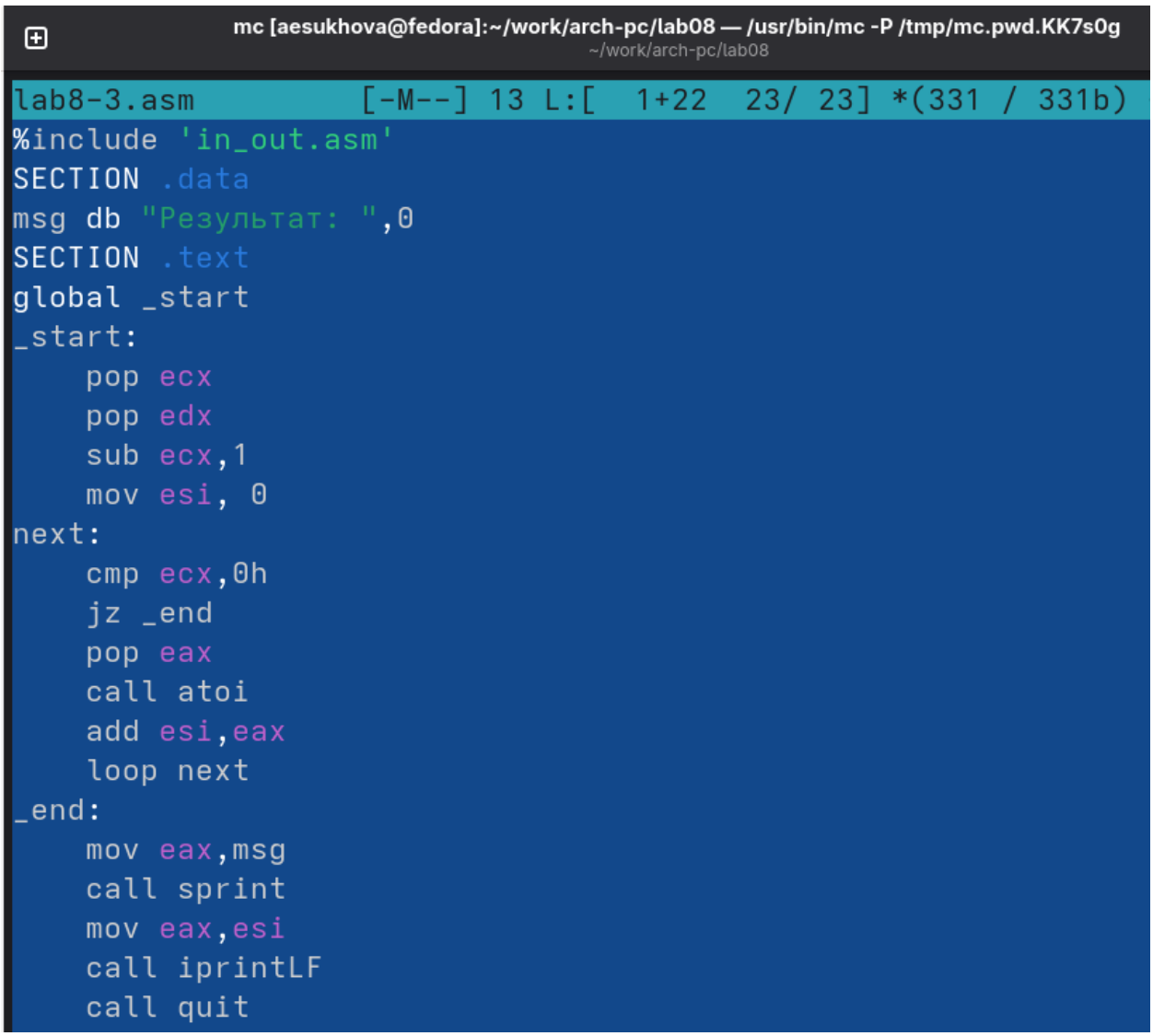

Рис.12 Заполняем файл

Создаём исполняемый файл и запускаем его, указав аргументы (рис. 13).

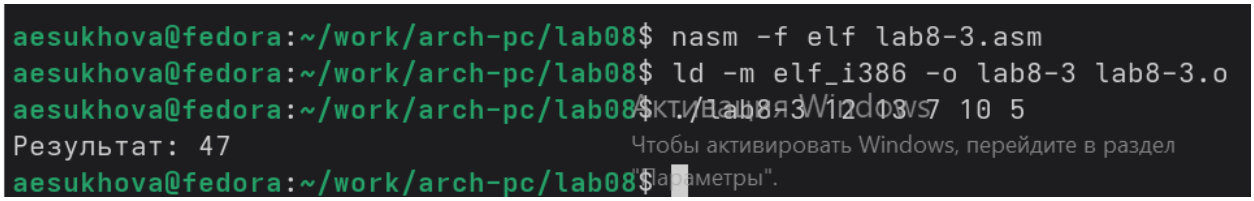

Рис.13 Смотрим на работу программы

Снова открываем файл для редактирования и изменяем его, чтобы вычислялось произведение вводимых значений (рис. 14).

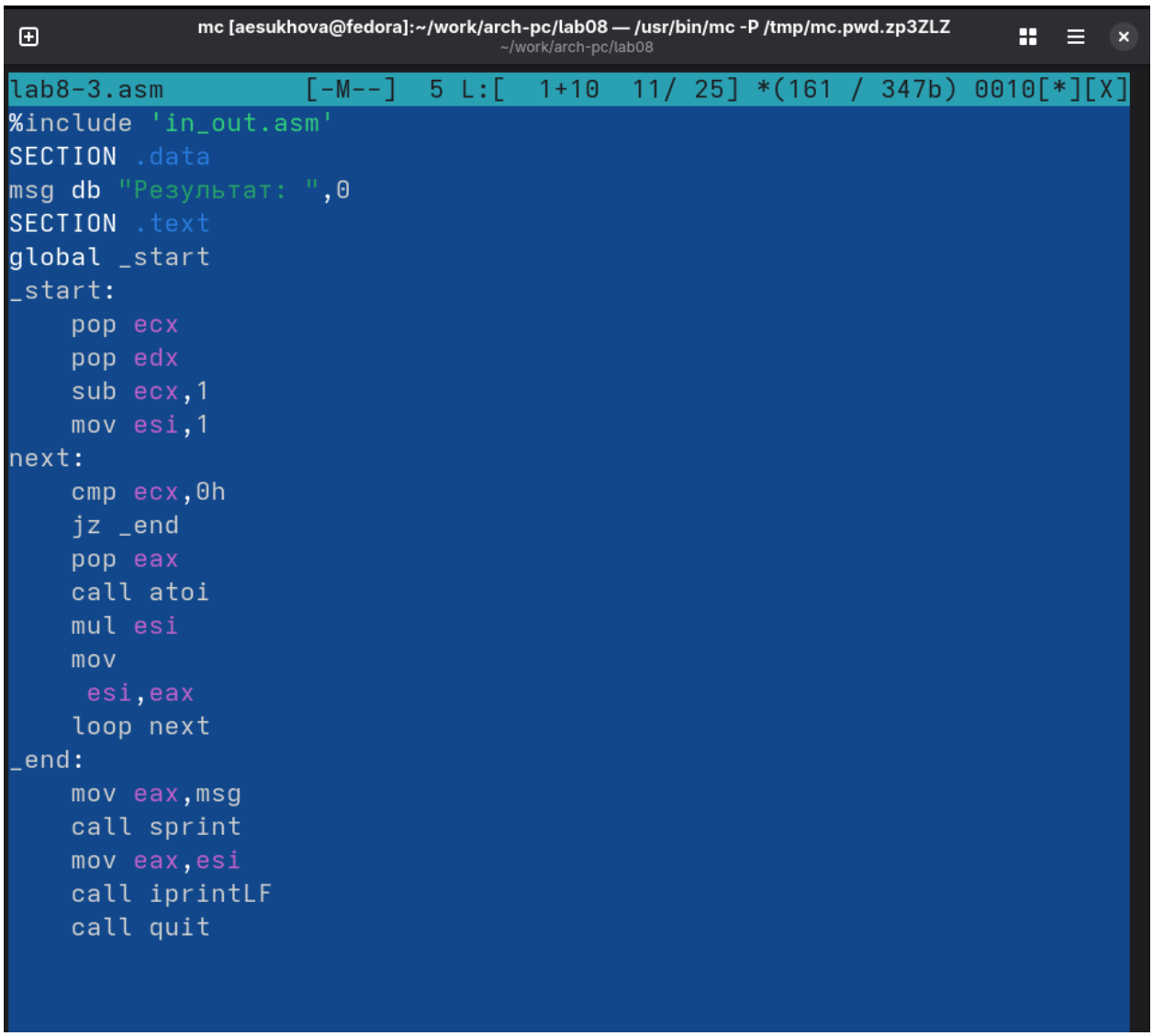

Рис.14 Изменяем файл

Создаём исполняемый файл и запускаем его, указав аргументы (рис. 14).

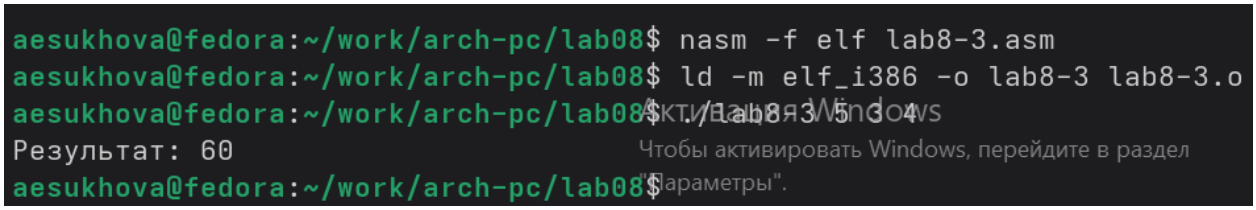

Рис.14 Проверяем работу файла(работает правильно)

3.3Задание для самостоятельной работы

ВАРИАНТ-5

1. Напишите программу, которая находит сумму значений функции $f(x)$ для $x = x1$, $x2$, …, xn, т.е. программа должна выводить значение $f(x1) + f(x2) + … + f(xn)$. Значения xi передаются как аргументы. Вид функции $f(x)$ выбрать из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным при выполнении

лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах $x = x1, x2, \ldots, xn$.

Создаем новый файл (рис. 15).

```
aesukhova@fedora:~/work/arch-pc/lab08$ touch lab8-4.1.asm
aesukhova@fedora:~/work/arch-pc/lab08$ mc
```

Рис.15 Создаем файл командой touch

Открываем его и пишем программу, которая выведет сумму значений, получившихся после решения выражения 3(10+x) (рис. 16).

mc [aesukhova@fedora]:~/work/arch-pc/lab08 — /usr/bin/mc -P /tmp/mc.pwd.4BZREl
~/work/arch-pc/lab08

/home/aesukhova/wor~lab08/lab8-4.1.asm 862/3373 25%

```
section .data
    msg_func db "Функция: f(x)=4x+3", 0xA, 0
    msg_res  db "Результат: ", 0
    newline  db 0xA, 0

section .bss
    sum resd 1
    buffer resb 10

section .text
    global _start

_start:
    ; Инициализация суммы
    mov dword [sum], 0

    ; Получаем аргументы командной строки
    pop ecx          ; argc
    pop edx          ; argv[0] (имя программы)
    dec ecx          ; количество чисел

    ; Выводим информацию о функции
    mov eax, msg_func
    call sprint

    ; Проверяем есть ли аргументы
    cmp ecx, 0
    je print_result

process_args:
    pop eax          ; получаем аргумент (строку)
    call atoi        ; преобразуем в число
```


Рис.16 Пишем программу

Транслируем файл и смотрим на работу программы (рис. 17).

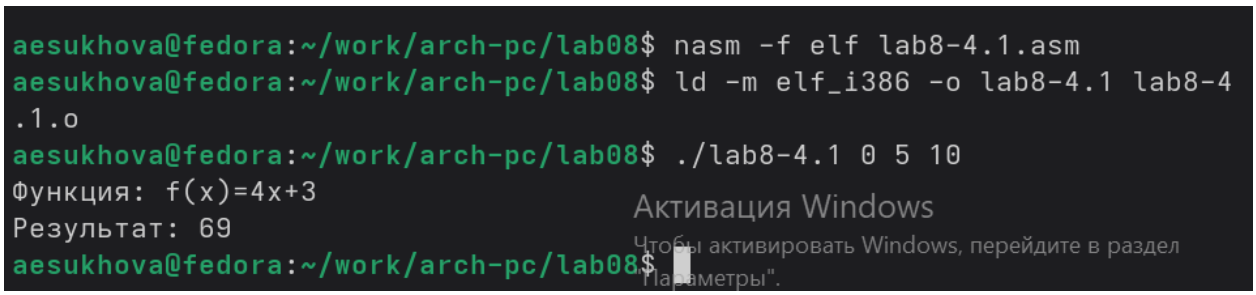

Рис.17 Смотрим на рабботу программы при x1=0 x2=5 x3=10(всё верно)

Транслируем файл и смотрим на работу программы (рис. 18).

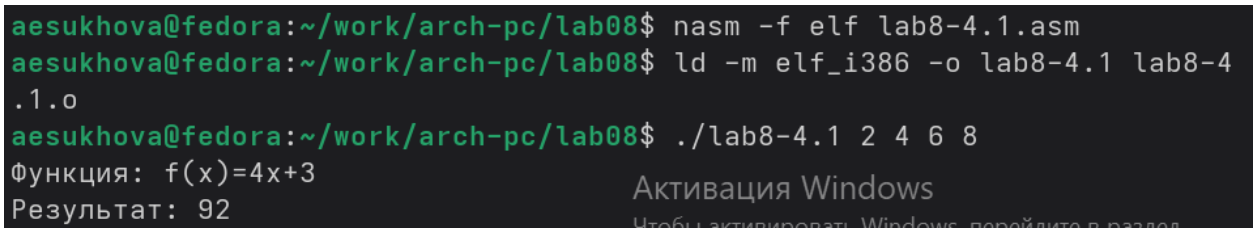

Рис.18 Смотрим на рабботу программы при x1=2 x2= 4 x3=6 x4=8(всё верно)

4. Выводы

Мы научились решать программы с использованием циклов и обработкой аргументов командной строки.