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: Юсупова Алина Руслановна

Группа: НКАбд-06-25

МОСКВА

2025 г.

Содержание

1 Цель работы

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 программ 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 NASM.

Создаю каталог для программам лабораторной работы № 8, перехожу в него и создаю файл lab8-1.asm(рис.1).

```
aryusupova@aryusupova-VirtualBox:~$ mkdir ~/work/arch-pc/lab08
aryusupova@aryusupova-VirtualBox:~$ cd ~/work/arch-pc/lab08
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ touch lab8-1.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ls
lab8-1.asm
```

Рис.1. Создание файла lab8-1.asm.

Открываю и вхожу в файл lab8-1.asm. Далее вставляю текст программы из листинга 8.1. (рис.2).

~/work/arch-pc/lab08/lab8-1.asm - Mousepad

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:
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 8.1.

Компоную и запускаю программу lab8-1.asm. Программа показывает работу циклов в NASM (рис.3).

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

Рис. 3. Запуск программы lab8-1.asm.

Изменяю программу так, что в теле цикла я изменяю значение регистра ecx (рис. 4).

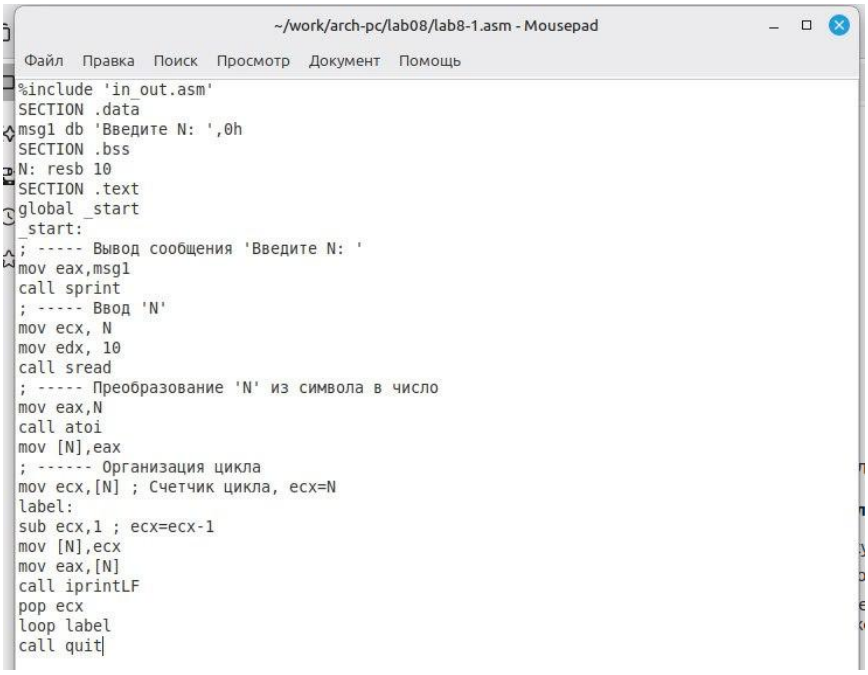

Рис. 4. Изменённая программа lab8-1.asm.

Далее компоную и запускаю изменённую программу lab8-1.asm(рис.5).

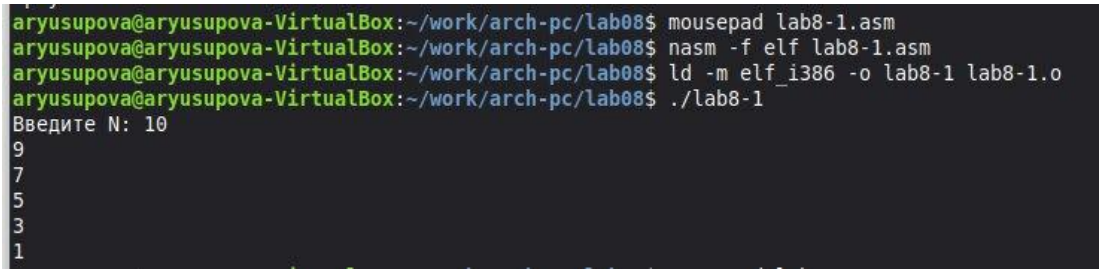

Рис.5. Запуск изменённой программы lab8-1.asm.

Добавляю команды pop и push в программу lab8-1.asm(рис.6).

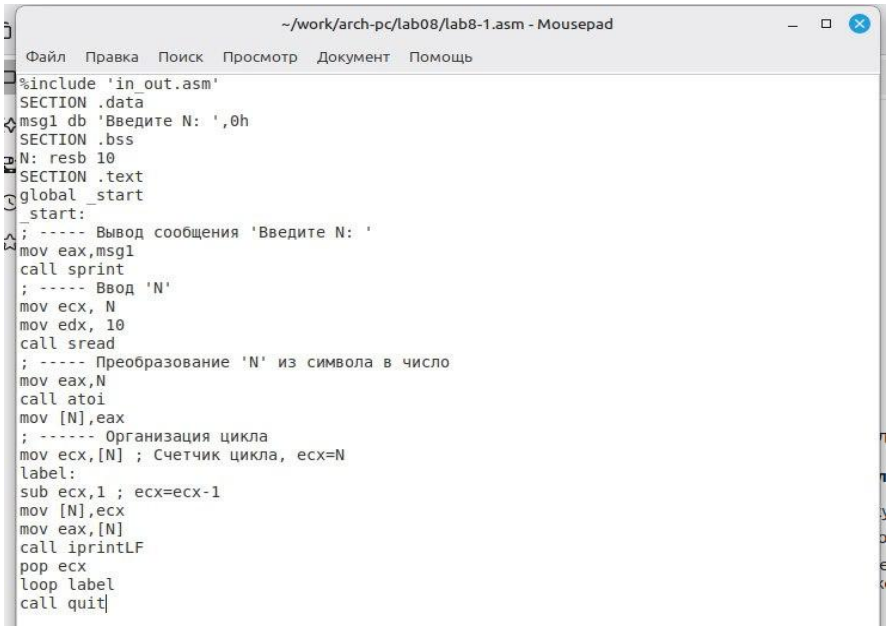

Рис.6. Изменённая программа lab8-1.asm.

Теперь количество итераций совпадает с введённым N, но произошло смещение вводимых чисел на -1 (рис.7).

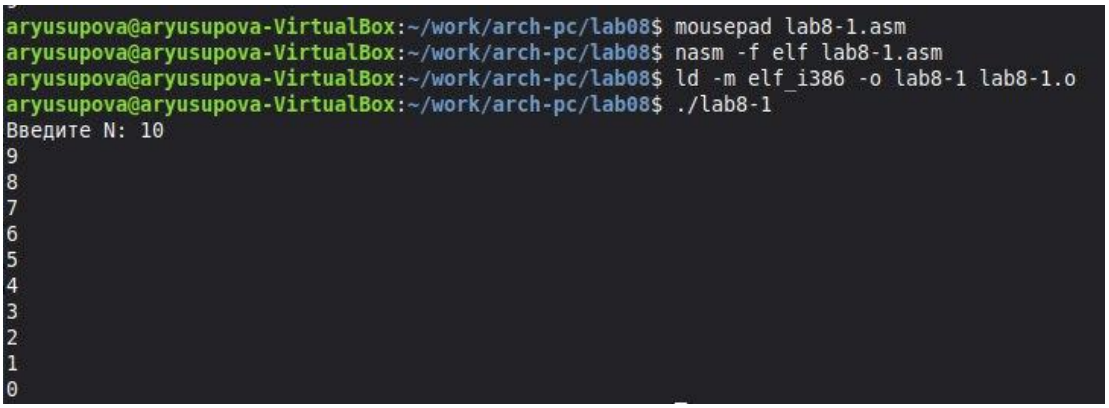

Рис.7. Запуск изменённой программы lab8-1.asm.

4.2. Обработка аргументов командной строки.

Создаю новый файл lab8-2.asm для программы и копирую в него код из листинга 8.2 (рис. 8-9).

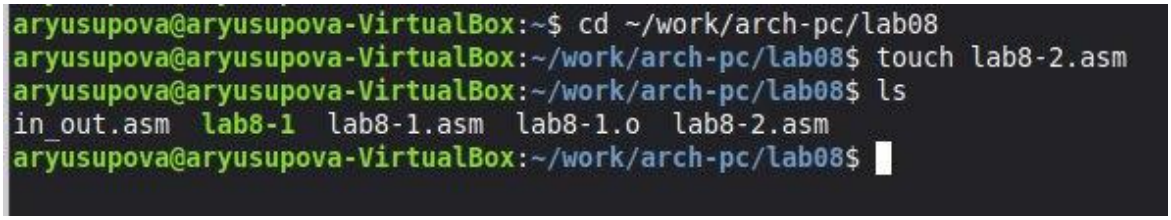

Рис.8. Создание файла lab8-2.asm.

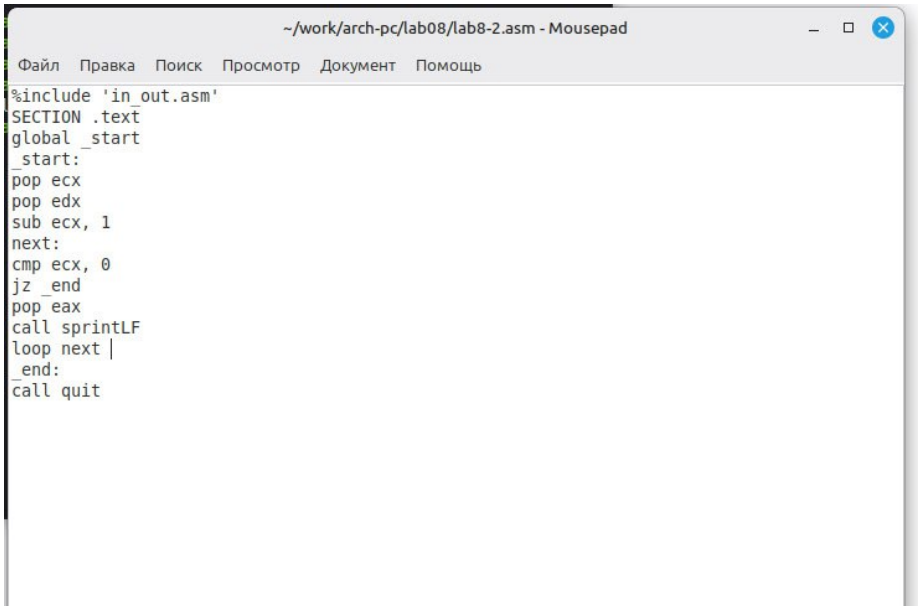

Рис.9. Программа lab8-2.asm.

Далее компилирую и запускаю программу lab8-2.asm, указав аргументы. Программой было обработано то же количество аргументов, что и было введено (рис.10).

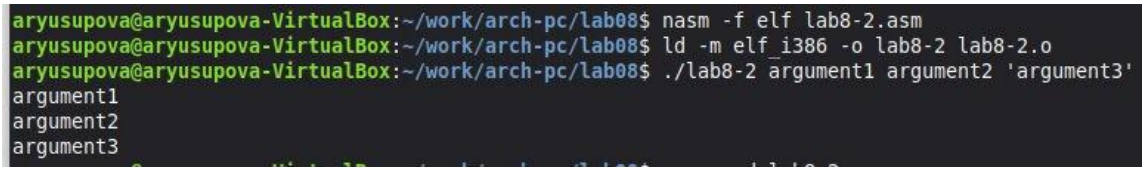

Рис.10. Запуск программы lab8-2.asm.

Создаю новый файл lab8-3.asm и копирую в него код из листинга 8.3 (рис.11-12).

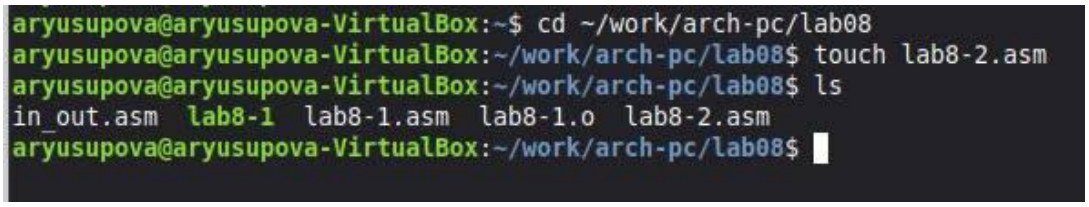

Рис.11. Создание файла lab8-3.asm.

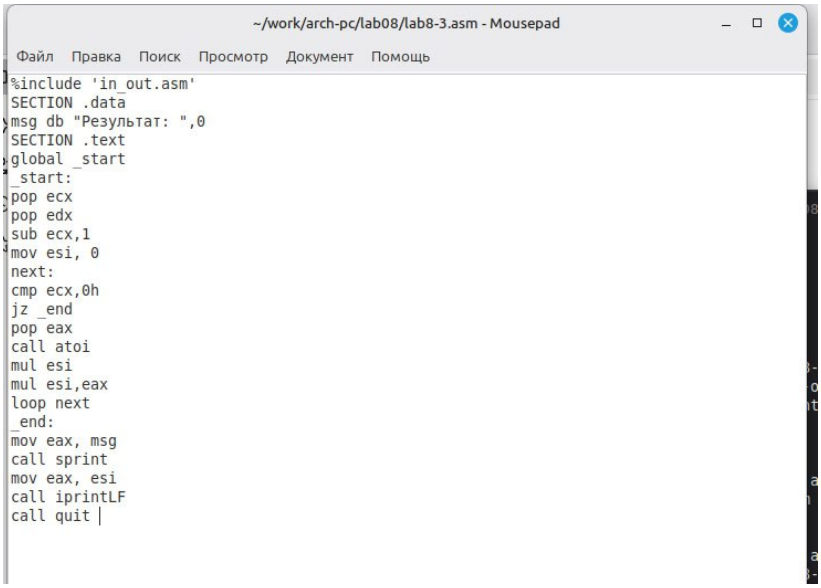

Рис.12. Копирование программы из листинга 8.3.

Компилирую программу и запускаю, указав в качестве аргументов некоторые числа, программа их складывает (рис.13).

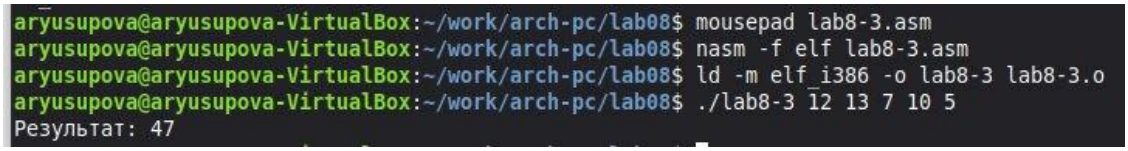

Рис.13. Запуск программы lab8-3.asm.

Изменяю поведение программы так, чтобы указанные аргументы она умножала, а не складывала (рис. 14).

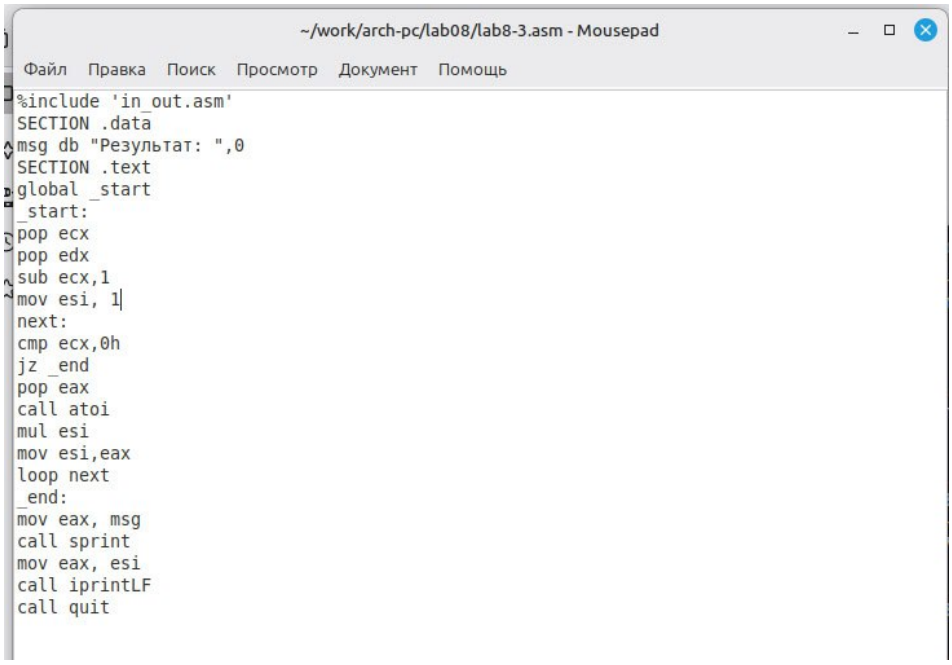

Рис. 14. Копирование изменённой программы.

Далее запускаю изменённую программу lab8-3.asm. Она действительно умножает введённые числа (рис.15).

Рис.15. Запуск изменённой программы.

4.3. Задание для самостоятельной работы.

Создаю файл lab8-4.asm в каталоге lab08 для написания моей программы (рис. 16).

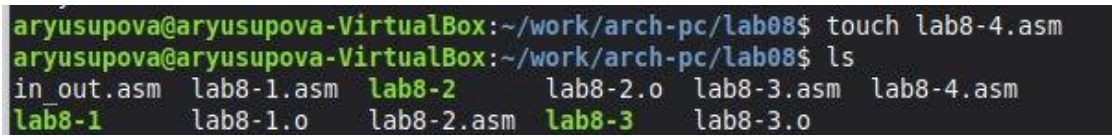

Рис.16. Создание файла lab8-4.asm.

Пишу программу, которая будет находить сумма значений для функции f(x)=10x-4, которая совпадает с моим девятым вариантом (рис. 17).

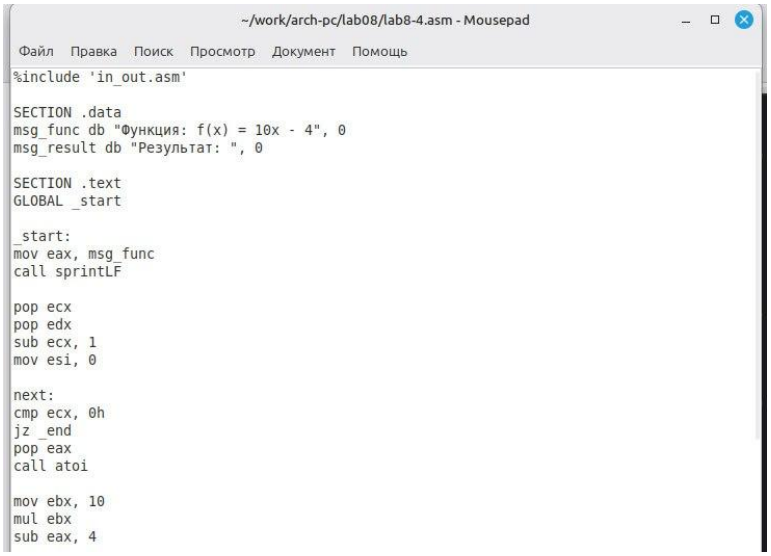

Рис.17. Написание программы для самостоятельной работы

Код программы:

```
%include 'in_out.asm'
```

```
SECTION .data
msg_func db "Функция: f(x) = 10x - 4", 0
msg_result 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

mov ebx, 10
mul ebx
sub eax, 4

add esi, eax

loop next

_end:
mov eax, msg_result
```

```
call sprint
mov eax, esi
call iprintLF
call quit
```

Проверяю работу программы, указав в качестве аргумента несколько чисел (рис. 18).

```
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ nasm -f elf lab8-4.asm
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ld -m elf_i386 -o lab8-4 lab8-4.o
aryusupova@aryusupova-VirtualBox:~/work/arch-pc/lab08$ ./lab8-4 1 2 3
Функция: f(x) = 10x - 4
Результат: 48
```

Рис. 18. Запуск программы для самостоятельной работы

5 Выводы

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

6 Список литературы.

1. Программирование на языке ассемблера NASM Столяров А. В.
2. https://esystem.rudn.ru/
3. https://metanit.com/assembler/nasm/2.1.php
4. Мой гитхаб: https://github.com/alyusupova/study_2025-2026_arh-pc.git