Отчёт по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки

Новикова Анастасия Андреевна

Содержание

# 1 Цель работы

Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

# 2 Задание

1. Реализация циклов в NASM
2. Обработка аргументов командной строки
3. Задание для самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Реализация циклов в NASM

Создаю директорию, в которой буду выполнять лабораторную работу, перехожу в созданный каталог и создаю файл lab8-1.asm, в котором буду выполнять первое задание (рис. 1).

Создание каталога и файла в нём

Рис. 1: Создание каталога и файла в нём

Ввожу текст кода из листинга 8.1 ((рис. 2). Эта программа запрашивает число N и выдает все числа перед N вместе с ним до 0 не включительно.

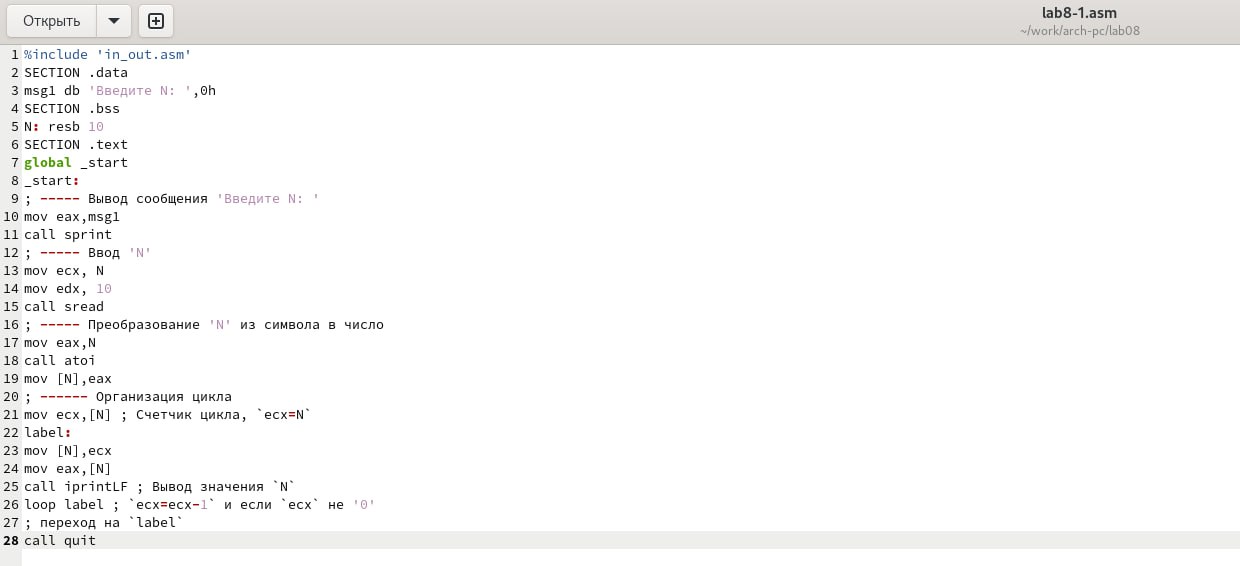


Рис. 2: Редактирование программы

Создаю исполняемый код (рис. 3). После его запуска убеждаюсь, что программа работает успешно.

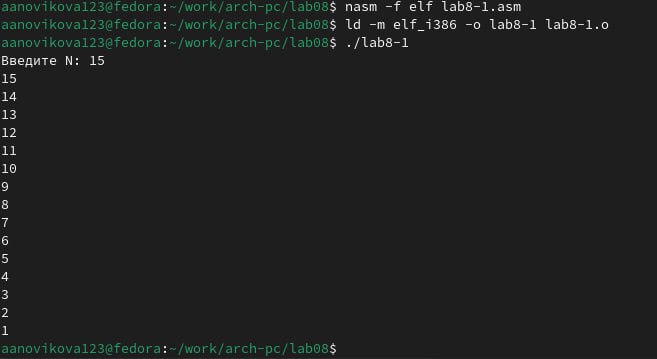


Рис. 3: Запуск программы

Теперь я редактирую код, добавив изменение значение регистра ecx в цикле (рис. 4)



Рис. 4: Редактирование программы

Запускаю программу. Теперь код передаёт значение через 2 числа. Регистр ecx принимает значения 9, 7, 3, 5, 1. Число проходов по циклу не равно N. (рис. 5)

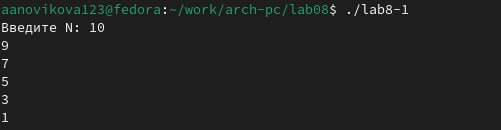


Рис. 5: Запуск программы

Еще раз редактирую код программы, добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop (рис. 6).



Рис. 6: Редактирование программы

Создаю и запускаю исполняемый файл (рис. 7). Теперь программа показывает все предыдущие числа до 0, не включая заданное N. Число проходов по циклу равно N.

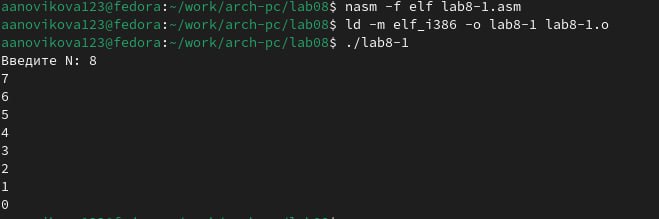


Рис. 7: Запуск программы

## 3.2 Обработка аргументов командной строки

Создаю новый файл lab8-2.asm и ввожу в него код из листинга 8.2 (рис. 8). Данная программа позволяет выводить на экран аргументы командной строки.

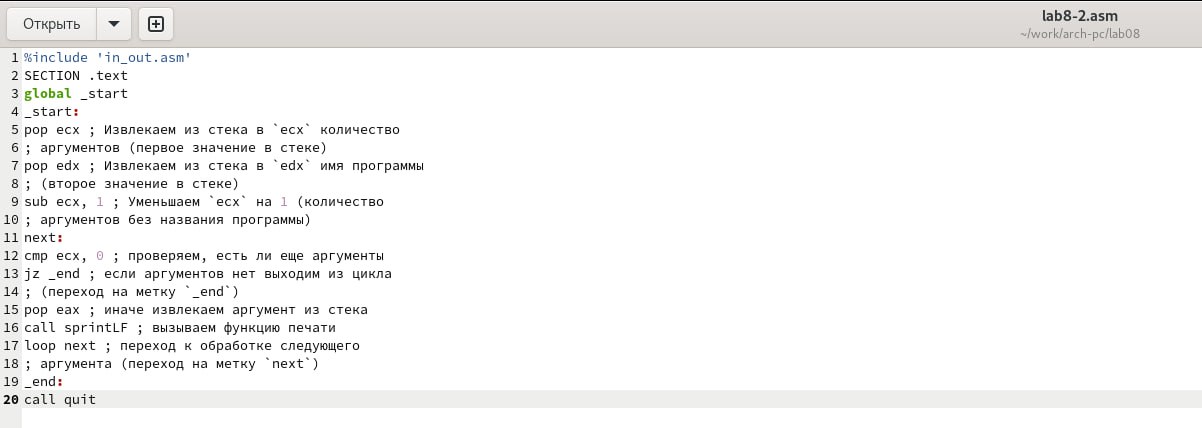


Рис. 8: Редактирование программы

Запускаю исполняемый файл вместе с аргументами (аргумент1 аргумент 2 ‘аргумент 3’) (рис. 9). Программа обработала 4 аргумента: аргумент1 - 1-ый аргумент, аргумент - 2-ой аргумент, 2 - 3-ий аргумент, ‘аргумент 3’ - 4-ый аргумент.



Рис. 9: Запуск программы

Создаю новый файл lab8-3.asm, используя команду touch. Ввожу в него код из листинга 8.3 (рис. 10). Данная программа позволяет выводить на экран сумму аргументов командной строки.

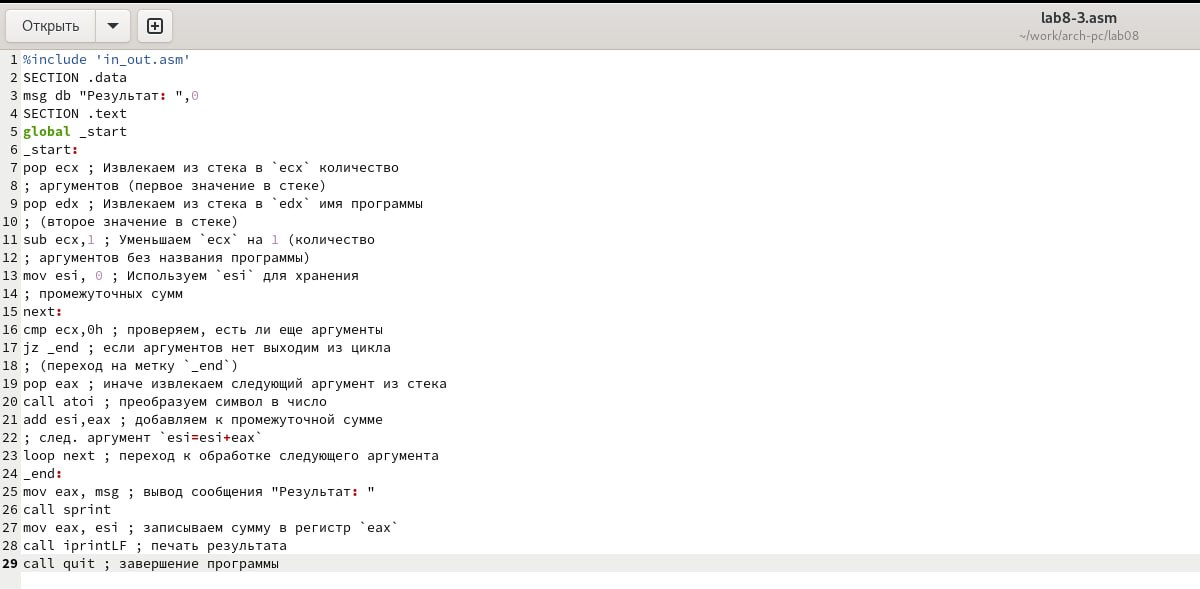


Рис. 10: Редактирование программы

Создаю и запускаю исполняемый файл вместе с аргументами (15, 5, 8, 11, 3) (рис. 11). Программа действительно выдаёт сумму всех аргументов.



Рис. 11: Запуск программы

Теперь редактирую код программы так, чтобы она выводила произведение всех аргументов (рис. 12).

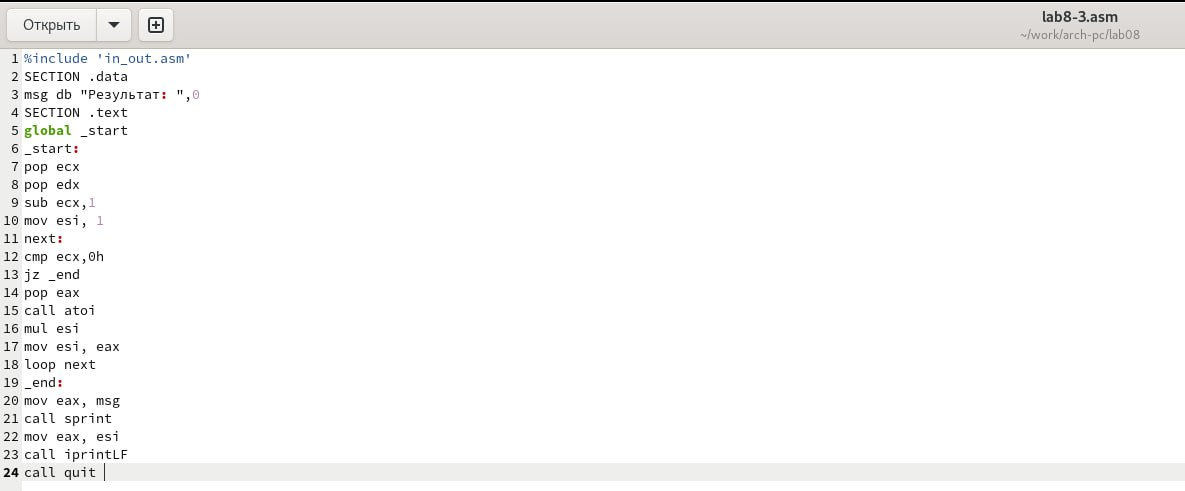


Рис. 12: Редактирование программы

%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  
call sprint  
mov eax, esi   
call iprintLF   
call quit

Запускаю исполняемый файл вместе с аргументами (10, 5, 6, 3, 11) (рис. 13). Программа выдаёт произведение всех аргументов.

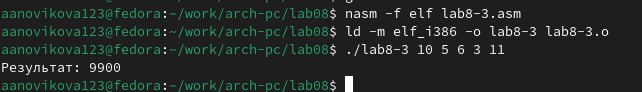


Рис. 13: Запуск программы

## 3.3 Задание для самостоятельной работы

***ВАРИАНТ 20***

Создаю файл lab8-4.asm в котором буду писать код для последней задачи (рис. 14)

Создание файла

Рис. 14: Создание файла

Пишу код программы, который позволяет вывести сумму всех преобразованных аргументов. Преобразования я беру из варианта задания №20 3(10 + x) (рис. 15)

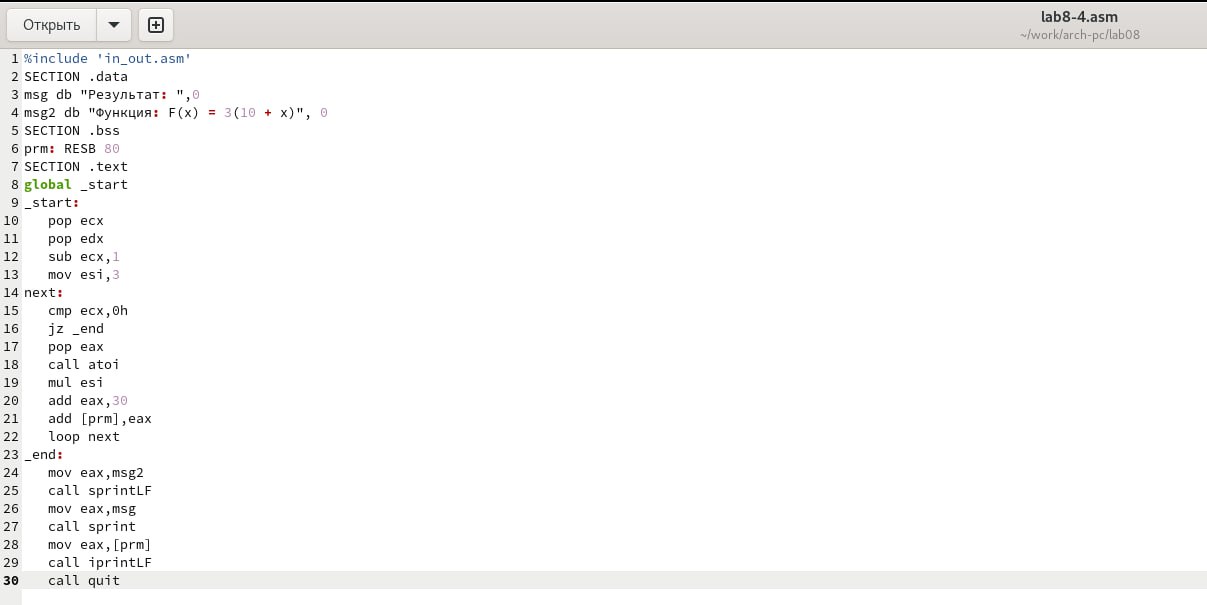


Рис. 15: Редактирование программы

Листинг написанной программы:

%include 'in\_out.asm'  
SECTION .data  
msg db "Результат: ",0  
msg2 db "Функция: F(x) = 3(10 + x)", 0  
SECTION .bss  
prm: RESB 80  
SECTION .text  
global \_start  
\_start:  
 pop ecx  
 pop edx  
 sub ecx,1  
 mov esi,3  
next:  
 cmp ecx,0h  
 jz \_end  
 pop eax  
 call atoi  
 mul esi  
 add eax,30  
 add [prm],eax  
 loop next  
\_end:  
 mov eax,msg2  
 call sprintLF  
 mov eax,msg  
 call sprint  
 mov eax,[prm]  
 call iprintLF  
 call quit

Запускаю исполняемый файл вместе с аргументами (1, 2, 4, 5) (рис. 16). Программа выдаёт верную сумму всех преобразованных аргументов.

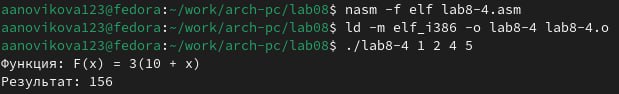


Рис. 16: Запуск программы

Повторно запускаю программу с другими аргументами (2, 5, 10, 11), чтобы убедиться, что всё работает верно (рис. 17). Программа выдает верный ответ.

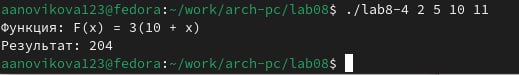


Рис. 17: Повторный запуск программы

# 4 Выводы

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