Отчет по лабораторной работе №8

Дисциплина: архитектура компьютера

Ицков Андрей Станиславович

Содержание

# 1 Цель работы

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

# 2 Задание

1. Реализация циклом 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 и пустой файл lab8-1.asm в нем (рис. -fig. 1).

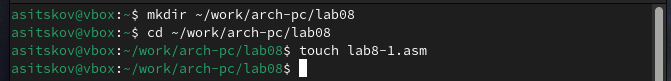


Рис. 1: Создание каталога и файла

Вставляю в файл данную мне программу из листинга (рис. -fig. 2).

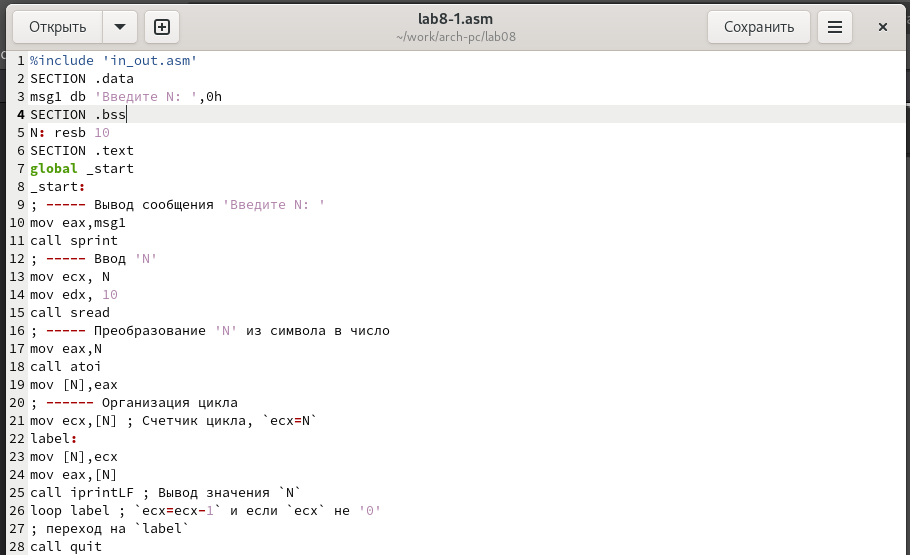


Рис. 2: Копирование программы из листинга

Компилирую и запускаю программу (рис. -fig. 3).

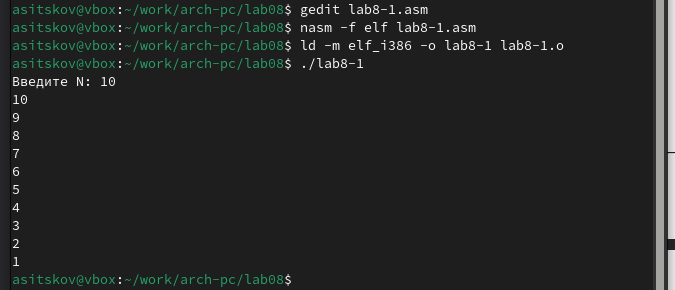


Рис. 3: Запуск программы

Изменяю текст программы, дбавляя изменение значения регистра eax в цикле (рис. -fig. 4).

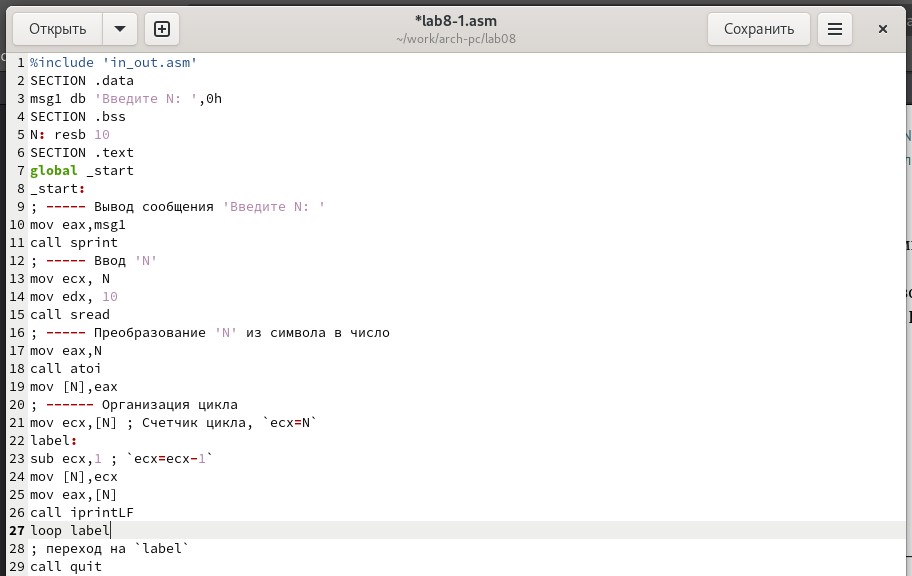


Рис. 4: Изменение программы

Запускаю программу и вижу, что количество проходов уменьшилось вдвое (рис. -fig. 5).

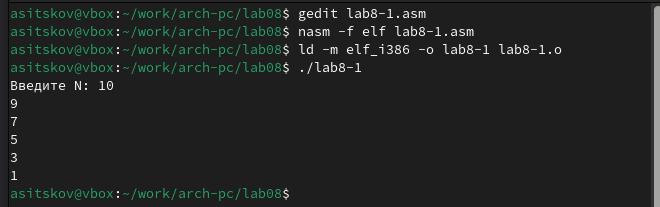


Рис. 5: Запуск программы

Изменяю текст программы, добавляя команды push и pop (рис. -fig. 6).



Рис. 6: Изменение программы

Запускаю программу и вижу, что количество проходов совпадает введенному N, но произошло смещение выводимых чисел на один (рис. -fig. 7).



Рис. 7: Запуск программы

## 4.2 Обработка аргументов командной строки

Создаю новый пустой файл и копирую в него программу из листинга (рис. -fig. 8).

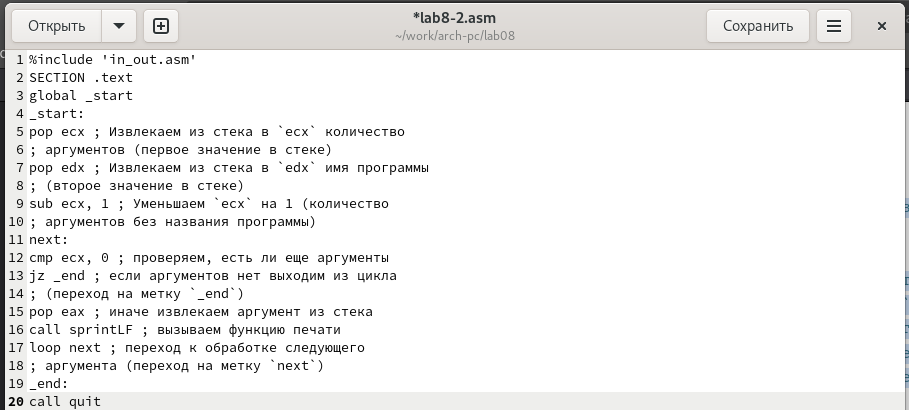


Рис. 8: Копирование программы из листинга

Компилирую и запускаю программу, указав аргументы (рис. -fig. 9).

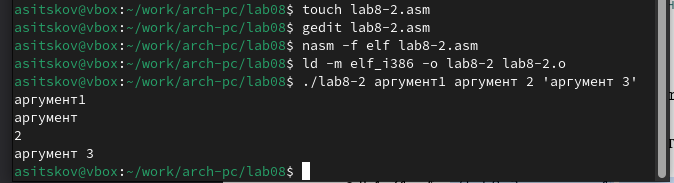


Рис. 9: Запуск программы

Создаю новый пустой файл и копирую в него программу из третьего листинга (рис. -fig. 10).

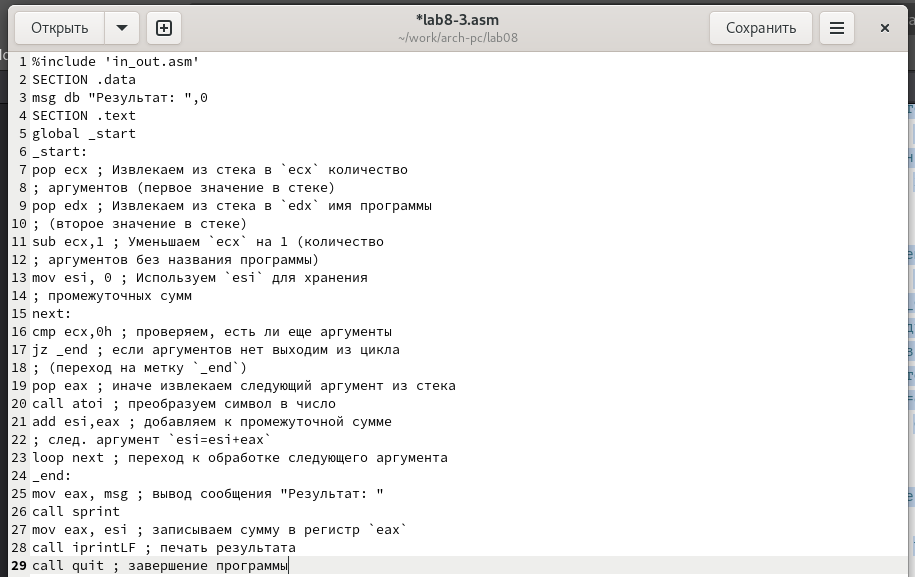


Рис. 10: Копирование программы из третьего листинга

Запускаю программу, добавив в аргументы данные мне числа (рис. -fig. 11).

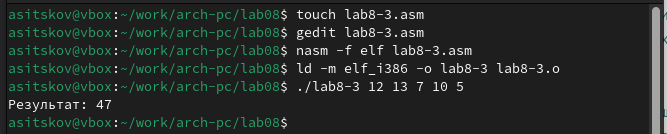


Рис. 11: Запуск программы

Изменяю программу так, чтобы числа из аргументов умножались а не складывались (рис. -fig. 12).

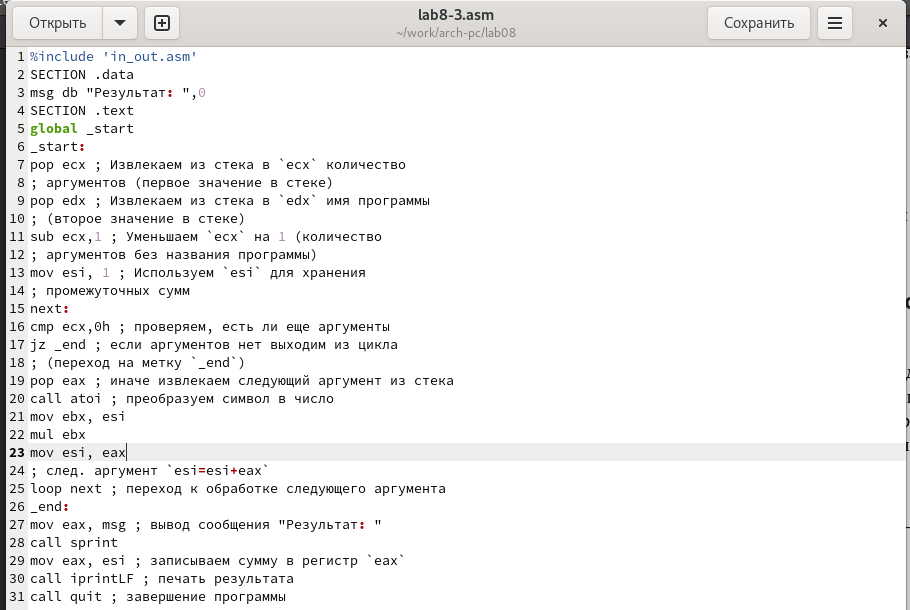


Рис. 12: Изменение программы

Запускаю программу и вижу что числа из аргументов умножились (рис. -fig. 13).

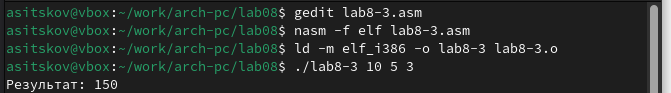


Рис. 13: Запуск программы

## 4.3 Задание для самостоятельной работы

Пишу программму, которая находит сумму значений функции 17 + 5x (данную для моего 18 варианта) (рис. -fig. 14).

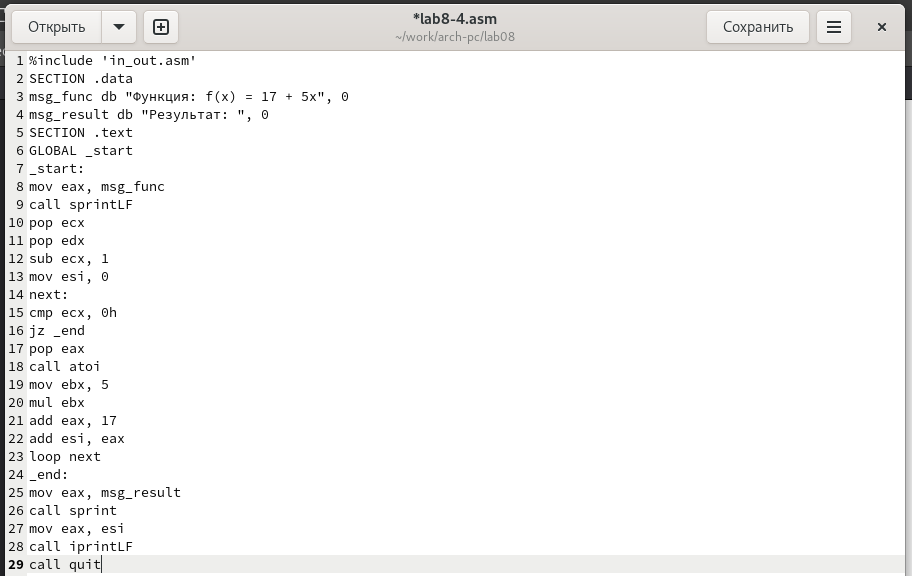


Рис. 14: Текст программы

Текст программы:

%include 'in\_out.asm'  
SECTION .data  
msg\_func db "Функция: f(x) = 17 + 5x", 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, 5  
mul ebx  
add eax, 17  
add esi, eax  
loop next  
\_end:  
mov eax, msg\_result  
call sprint  
mov eax, esi  
call iprintLF  
call quit

Запускаю программу и ввожу несколько чисел, вижу, что программа работает корректно (рис. -fig. 15).

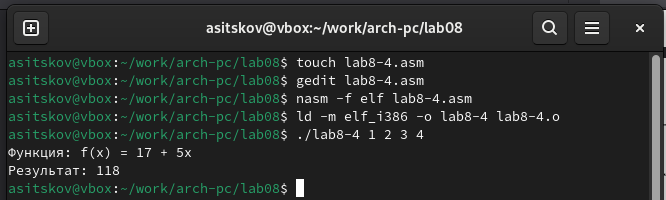


Рис. 15: Запуск программы

# 5 Выводы

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

# 6 Список литературы

1. [Курс на ТУИС](https://esystem.rudn.ru/course/view.php?id=112)
2. [Лабораторная работа №8](https://esystem.rudn.ru/pluginfile.php/2089095/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%968.%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D1%86%D0%B8%D0%BA%D0%BB%D0%B0.%20%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0%20%D0%B0%D1%80%D0%B3%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%20%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9%20%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8.pdf)
3. [Программирование на языке ассемблера NASM Столяров А. В.](https://esystem.rudn.ru/pluginfile.php/2088953/mod_resource/content/2/%D0%A1%D1%82%D0%BE%D0%BB%D1%8F%D1%80%D0%BE%D0%B2%20%D0%90.%20%D0%92.%20-%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D0%BD%D0%B0%20%D1%8F%D0%B7%D1%8B%D0%BA%D0%B5%20%D0%B0%D1%81%D1%81%D0%B5%D0%BC%D0%B1%D0%BB%D0%B5%D1%80%D0%B0%20NASM%20%D0%B4%D0%BB%D1%8F%20%D0%9E%D0%A1%20Unix.pdf)