Отчёта по лабораторной работе №8

Программирование цикла. Обработка аргументов командной строки.

Павлюченков Сергей Витальевич

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

* Реализация циклов в NASM

Создаю каталог для программ лабораторной работы № 8, перехожу в него и создаю файл lab8-1.asm (рис. [1](#fig:001)).

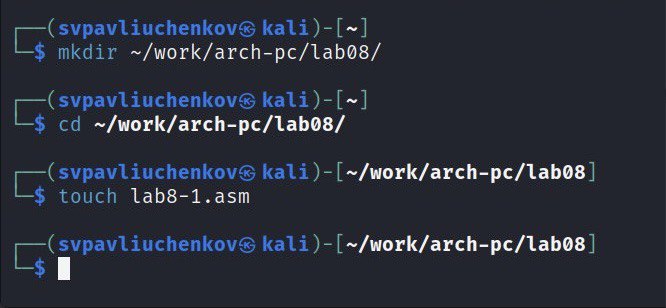


Figure 1: Создание первого файла

Ввожу в файл lab8-1.asm текст программы из листинга 8.1 (рис. [2](#fig:002)).

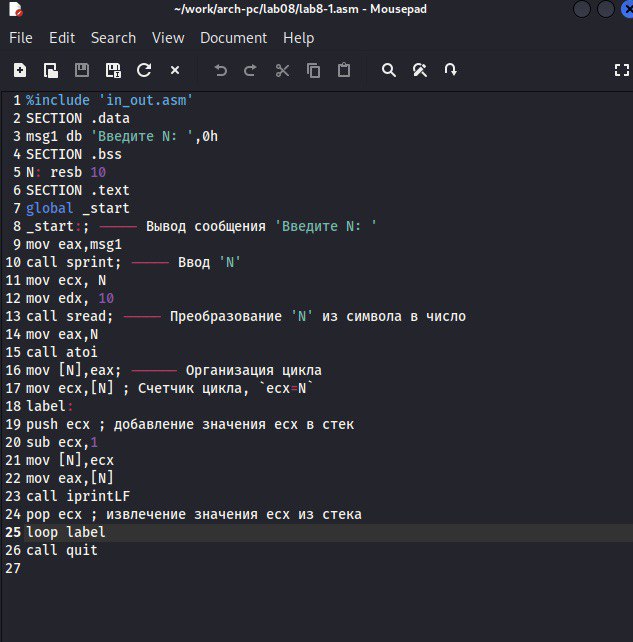


Figure 2: Программа из листинга 8.1

Создаю исполнительный файл и проверяю его работу (рис. [3](#fig:003)).

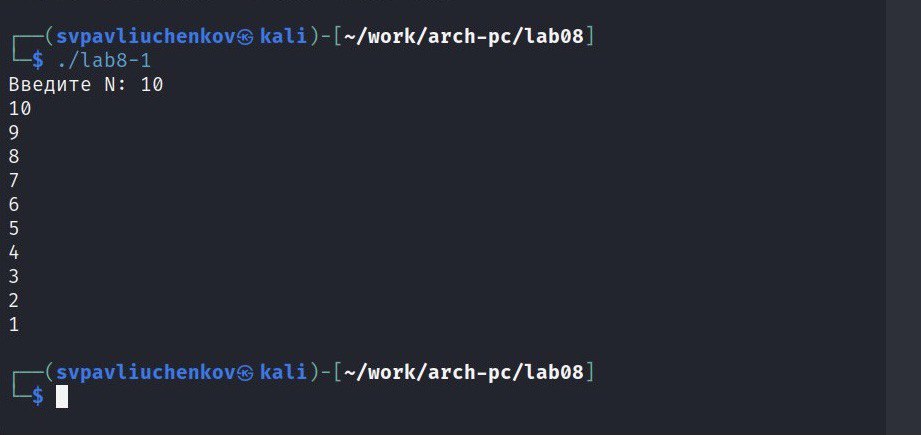


Figure 3: Запуск исполнительного файла

Этот алгоритм выводит все целые числа от N до 1.

Изменяю текст программы добавив изменение значение регистра ecx в цикле (рис. [3](#fig:003)).

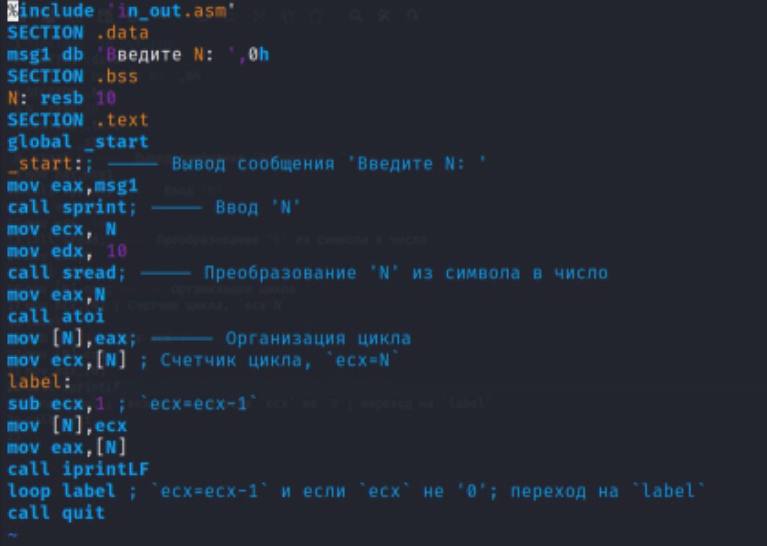


Figure 4: Измененная программа

Создаю исполнительный файл и проверяю его работу (рис. [5](#fig:005)).

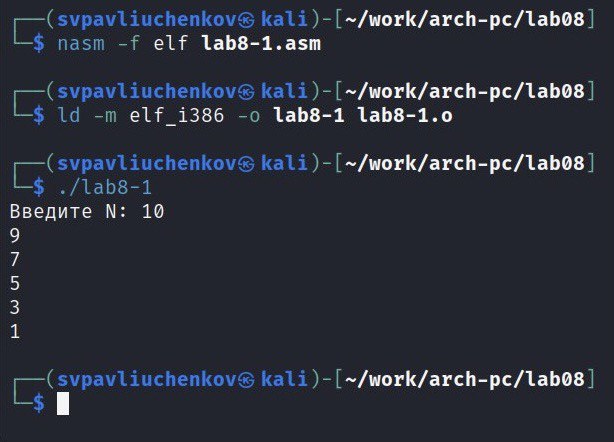


Figure 5: Запуск исполнительного файла

Число проходов цикла не соответствует значению 𝑁 введенному с клавиатуры.

Вношу изменения в текст программы добавив команду push и pop для сохранения значения счетчика цикла loop (рис. [6](#fig:006)).

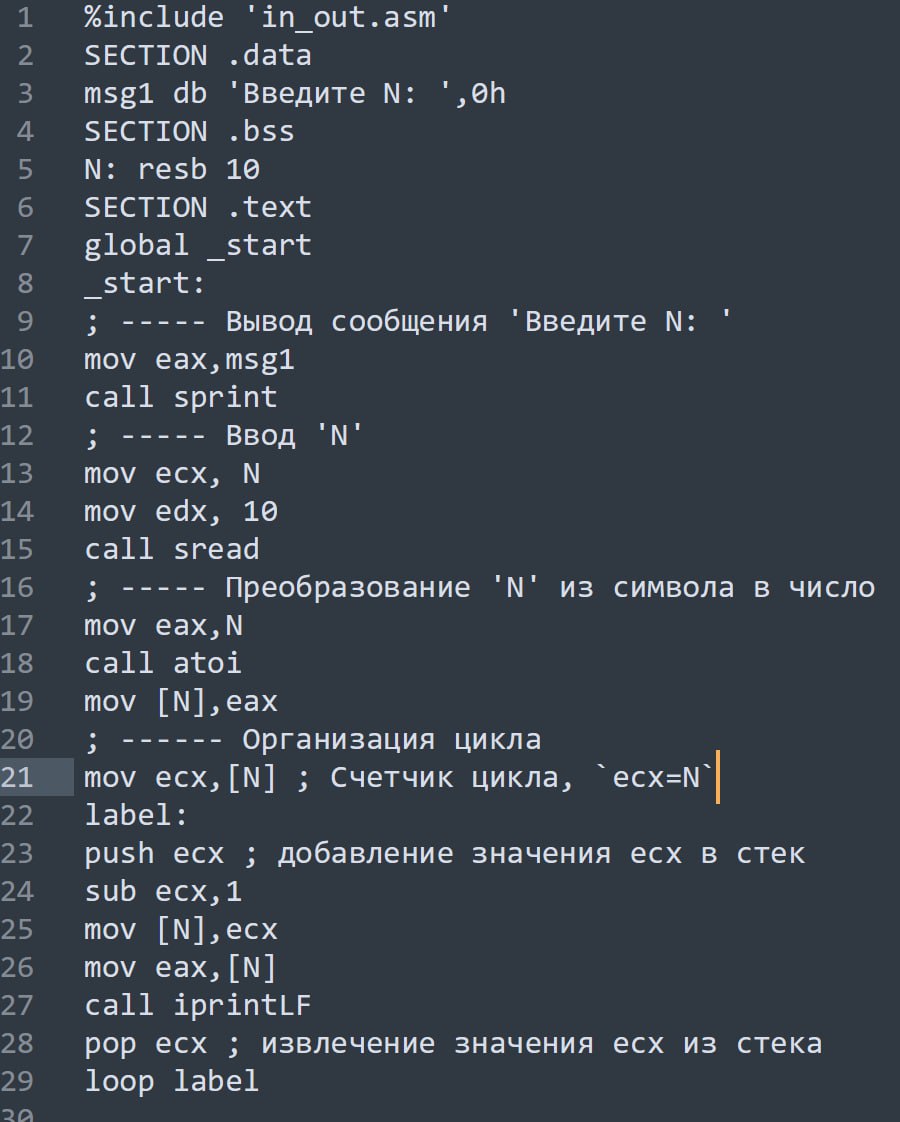


Figure 6: Измененная программа

Создаю исполнительный файл и проверяю его работу (рис. [5](#fig:005)).

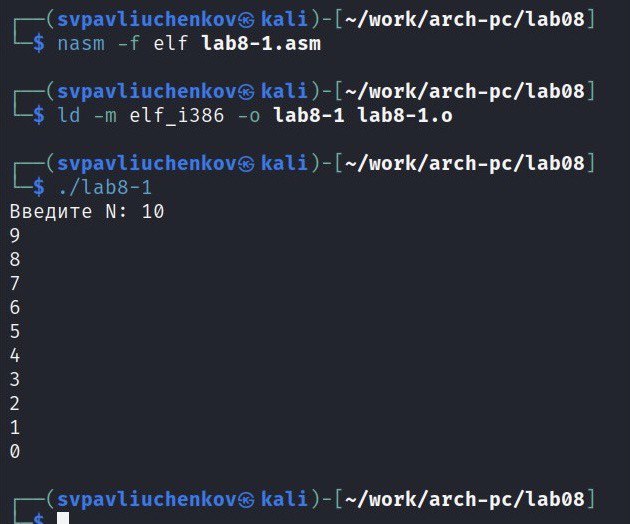


Figure 7: Запуск исполнительного файла

Программа выводит все целые числа от N-1 до 0 (рис. [7](#fig:007)).

Число проходов цикла соответствует значению 𝑁 введенному с клавиатуры.

* Обработка аргументов командной строки

Создаю файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и ввожу в него текст программы из листинга 8.2 (рис. [8](#fig:008)).

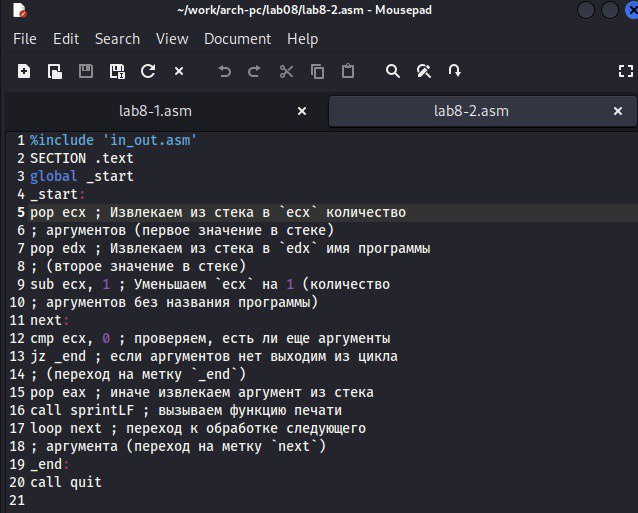


Figure 8: Программа из листинга 8.2

Создаю исполнительный файл и проверяю его работу (рис. [9](#fig:009)).

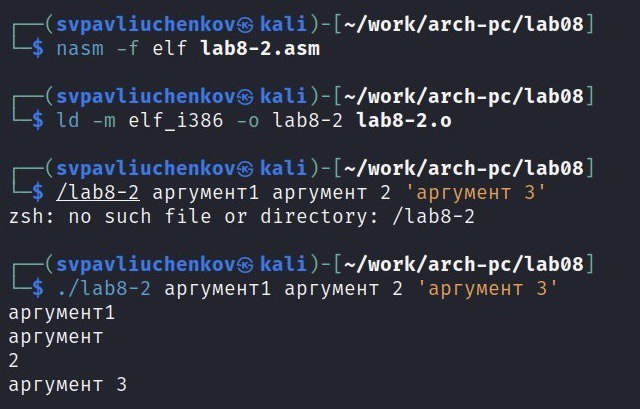


Figure 9: Запуск исполнительного файла

Программой было обработано 4 аргумента, так как 2 не было взято в кавычки, из-за чего засчиталось за отдельный аргумент.

Рассмотрим еще один пример программы которая выводит сумму чисел, которые передаются в программу как аргументы.

Создаю файл lab8-3.asm в каталоге ~/work/archpc/lab08 и ввожу в него текст программы из листинга 8.3(рис. [10](#fig:010)).

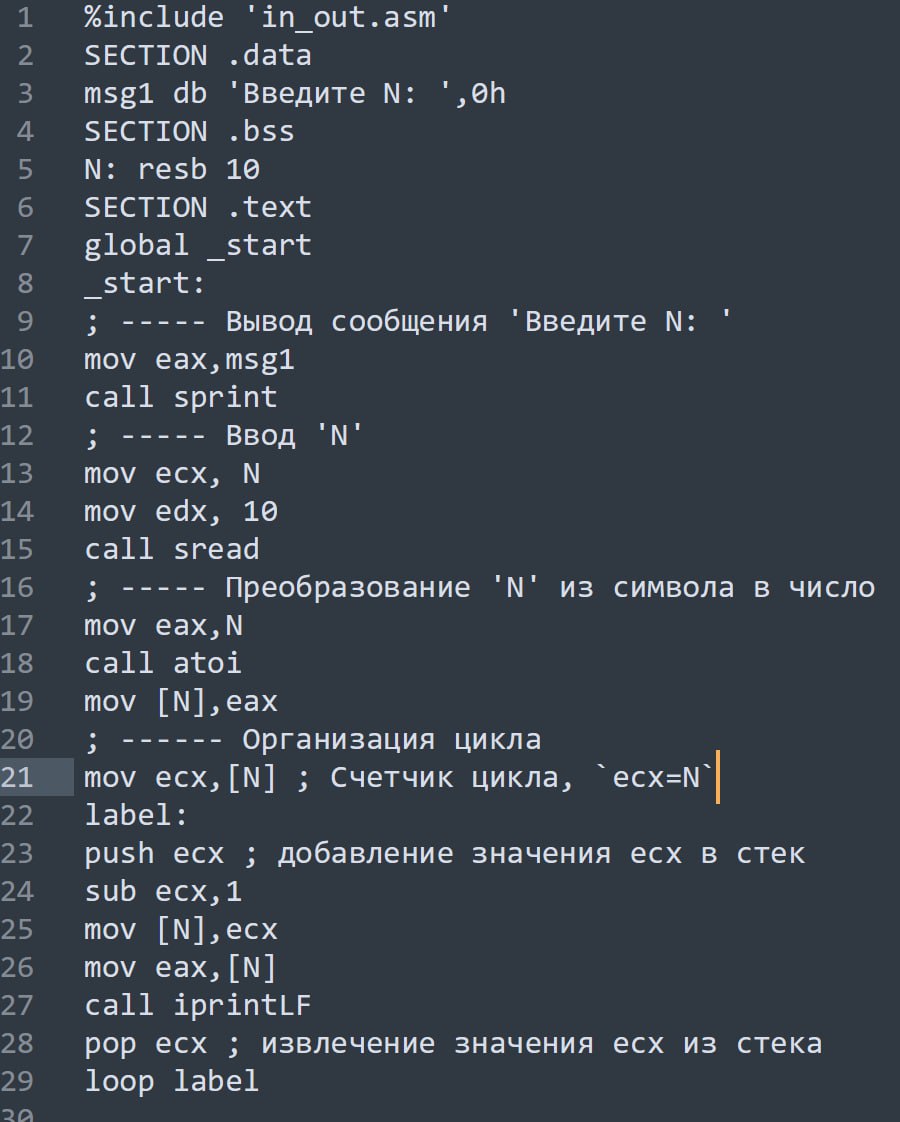


Figure 10: Программа из листинга 8.3

Создаю исполнительный файл и проверяю его работу (рис. [11](#fig:011)).

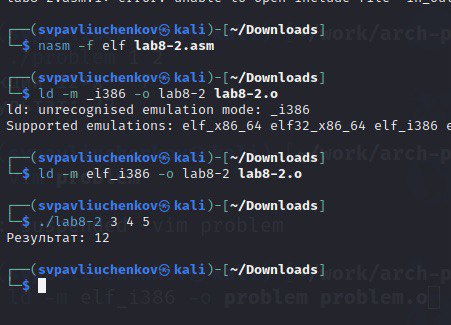


Figure 11: Запуск исполнительного файла

Изменяю текст программы из листинга 8.3 для вычисления произведения аргументов командной строки(рис. [12](#fig:012))

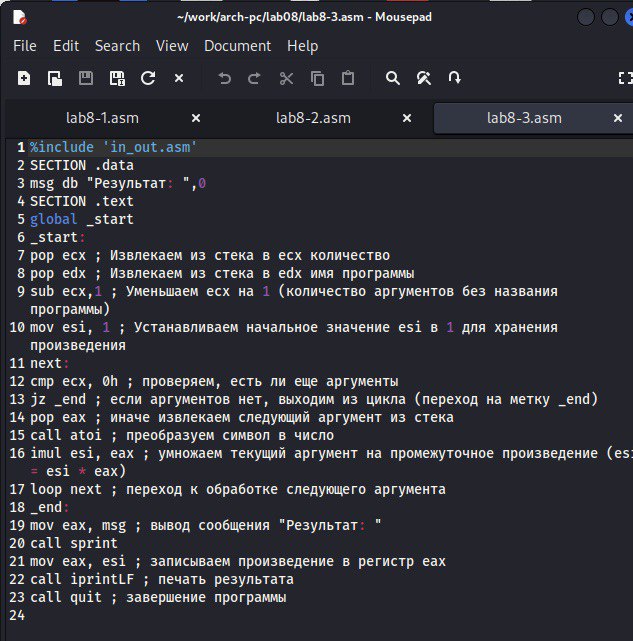


Figure 12: Измененная программа

Создаю исполнительный файл и проверяю его работу (рис. [3](#fig:003)).

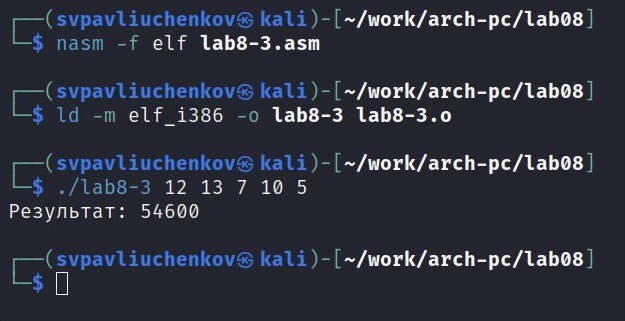


Figure 13: Запуск исполнительного файла

* Задание для самостоятельной работы

Пишу программу, которая находит сумму значений функции (12x - 7) для x=x1 x2, …, xn, т.е. программа должна выводить значение f(x1) + f(x2) +… + f(xn).Значения xi передаются как аргументы.

Создаю файл problem.asm пишу для него программу (рис. [14](#fig:014)).

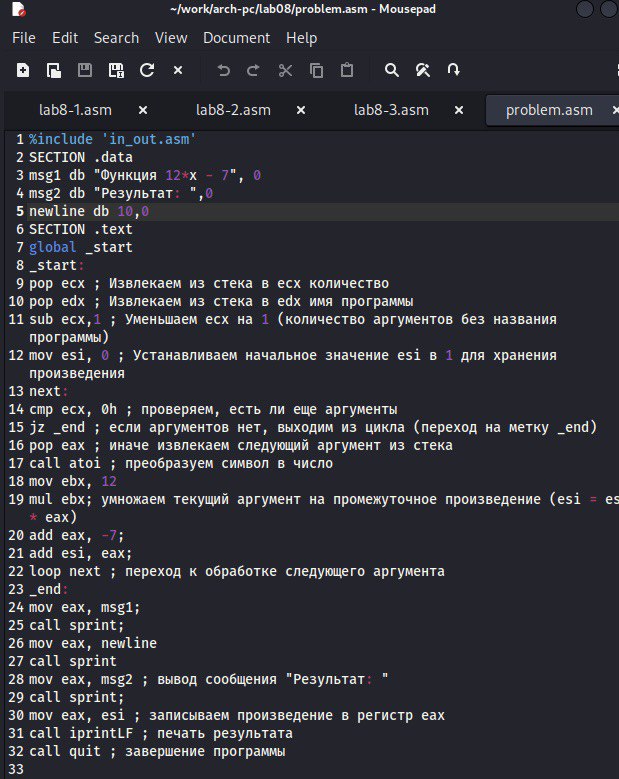


Figure 14: Получившаяся программа

Создаю исполняемый файл и проверяю его работу на нескольких наборах x = x1,x2,…,xn.

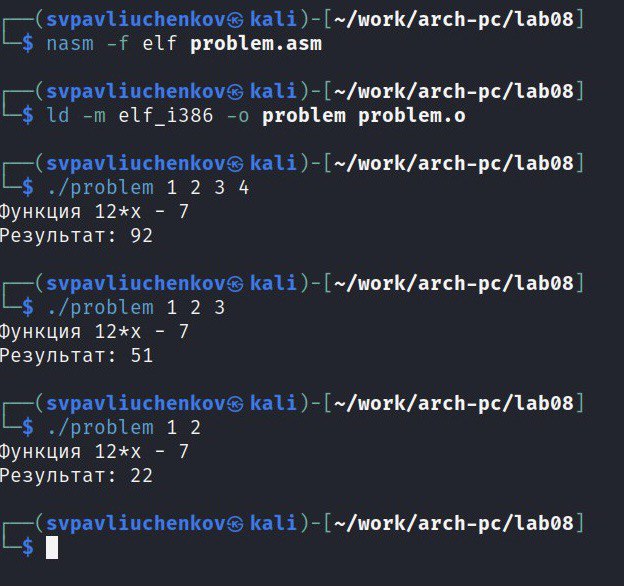


Figure 15: Запуск исполнительного файла

Программа выдает точные значения.

# 4 Выводы

Я научился писать программы с использованием циклов и обработкой аргументов командной строки.

# Список литературы

Лабораторная работа №8

GDB: The GNU Project Debugger. — URL: https://www.gnu.org/software/gdb/

NASM Assembly Language Tutorials. — 2021. — URL: https://asmtutor.com/