Отчёт по лабораторной работе №8

Дисциплина: Архитектура компьютера

Ларина Наталья Денисовна

Содержание

# 1 Цель работы

Целью данной лабораторной работы является 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 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.  
На рис. 8.1 показана схема организации стека в процессоре.  
Стек имеет вершину, адрес последнего добавленного элемента, который хранится в регистре esp (указатель стека). Противоположный конец стека называется дном. Значение, помещённое в стек последним, извлекается первым. При помещении значения в стек указатель стека уменьшается, а при извлечении — увеличивается.  
Для стека существует две основные операции:  
• добавление элемента в вершину стека (push);  
• извлечение элемента из вершины стека (pop).

# 4 Выполнение лабораторной работы

## 4.1 Реализация циклов в NASM

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

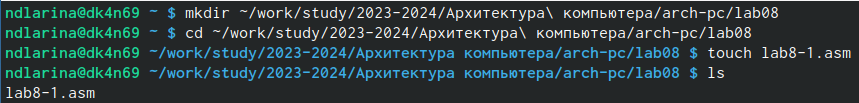


Figure 1: Создание файлов для лабораторной работы

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

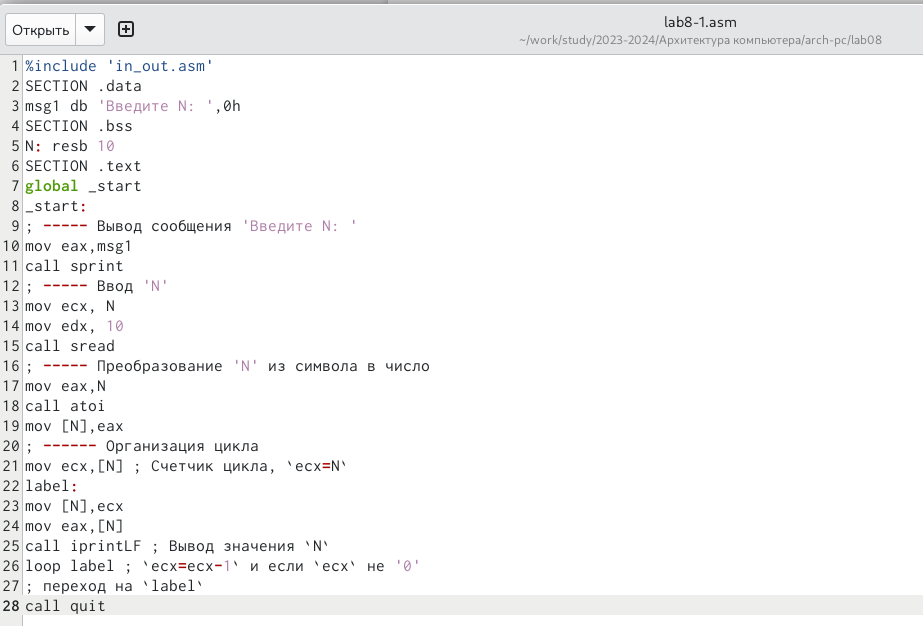


Figure 2: Ввод текста из листинга 8.1

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

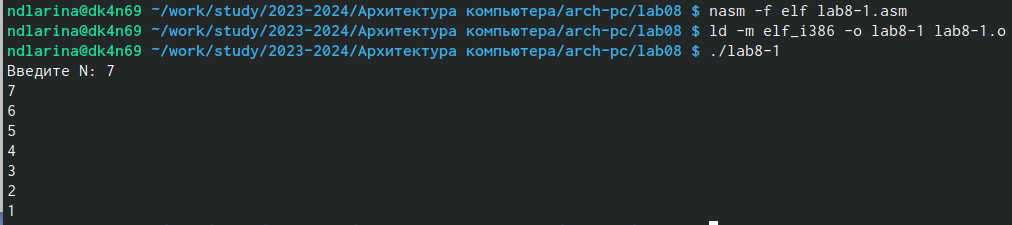


Figure 3: Запуск исполняемого файла

Данная программа выводит числа от N до 1 включительно.

Изменяю текст программы, добавив изменение значения регистра ecx в цикле. (рис. [4](#fig:004)).

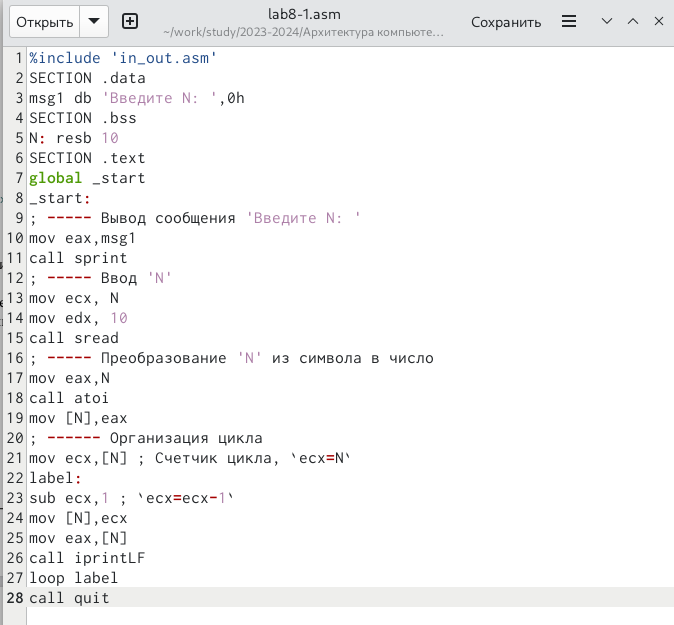


Figure 4: Изменение текста программы

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

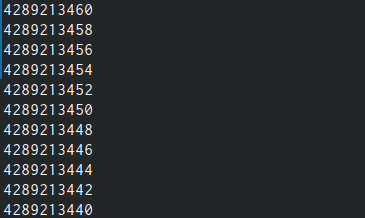


Figure 5: Запуск обновленной программы

В данном случае число проходов цикла не соответствует введенному с клавиатуры значению.

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

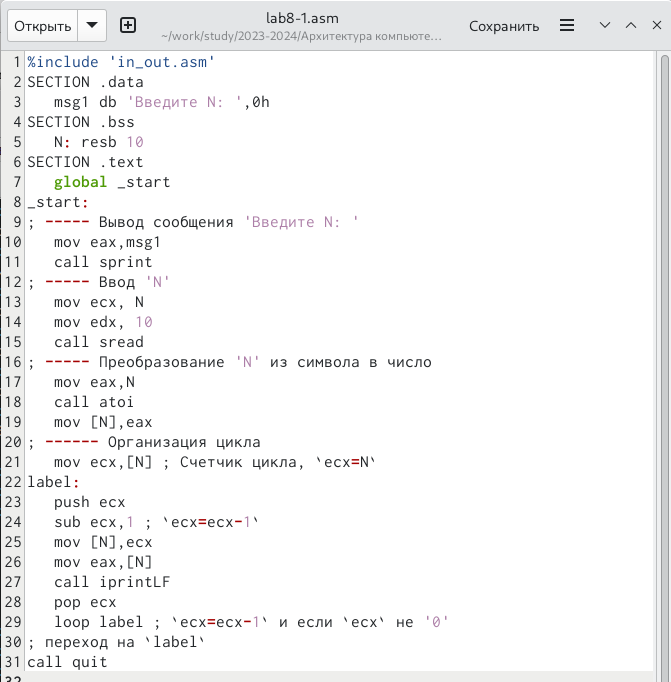


Figure 6: Изменение текста программы

Создаю исполняемый файл и проверяю его работу.(рис. [7](#fig:007)).

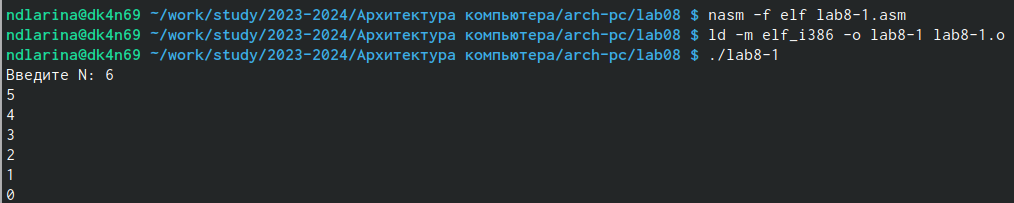


Figure 7: Запуск исполняемого файла

В данном случае число проходов цикла соответствует введенному с клавиатуры значению и выводит числа от N-1 до 0 включительно.

## 4.2 Обработка аргументов командной строки

Создаю файл lab8-2.asm в каталоге ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab08. (рис. [8](#fig:008)).

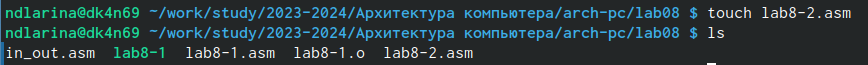


Figure 8: Создание файла lab8-2.asm

Ввожу в него текст программы из листинга 8.2. (рис. [9](#fig:009)).

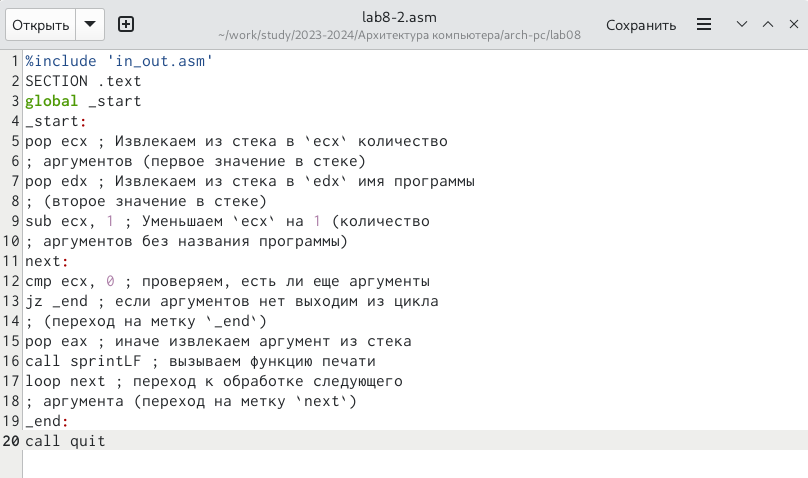


Figure 9: Ввод текста программы из листинга 8.2

Далее создаю исполняемый файл и запускаю его, указав нужные аргументы. (рис. [10](#fig:010)).

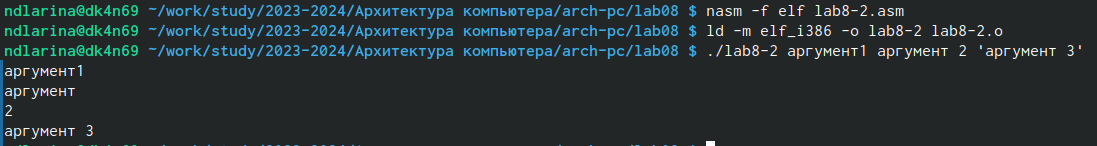


Figure 10: Запуск исполняемого файла

Программа вывела 4 аргумента, так как аргумент 2 не взят в кавычки, в отличии от аргумента 3, поэтому из-за пробела программа считывает “2” как отдельный аргумент.

Рассмотрим пример программы, которая выводит сумму чисел, которые передаются в программу как аргументы. Создаю файл lab8-3.asm в каталоге ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab08. (рис. [11](#fig:011)).

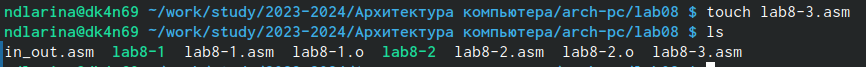


Figure 11: Создание файла lab8-3.asm

Ввожу в него текст программы из листинга 8.3. (рис. [12](#fig:012)).

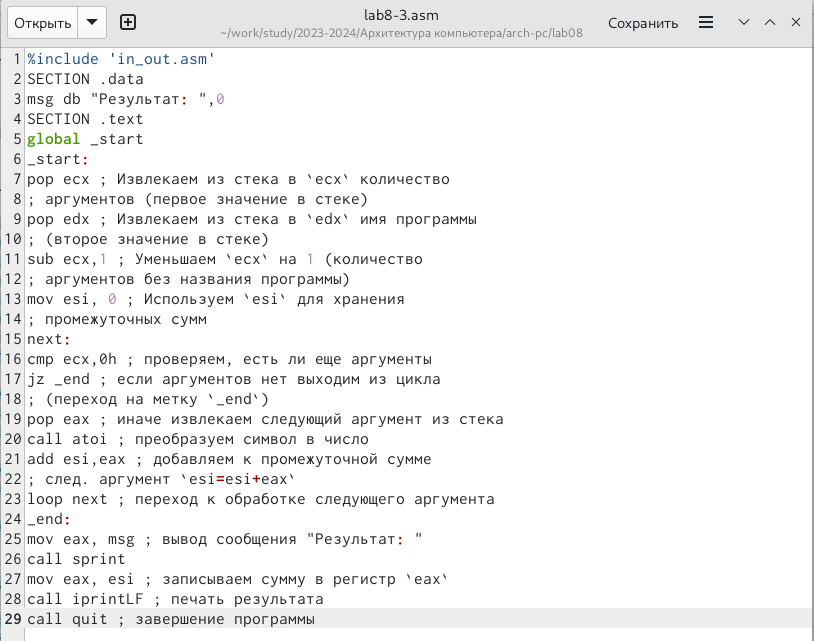


Figure 12: Ввод текста программы из листинга 8.3

Создаю исполняемый файл и запускаю его, указав аргументы. (рис. [13](#fig:013)).

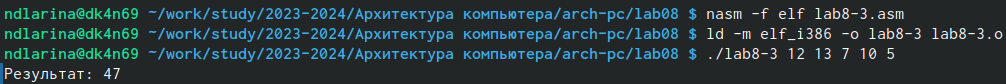


Figure 13: Запуск исполняемого файла

Изменяю текст программы из листинга 8.3 для вычисления произведения аргументов командной строки. (рис. [14](#fig:014)).

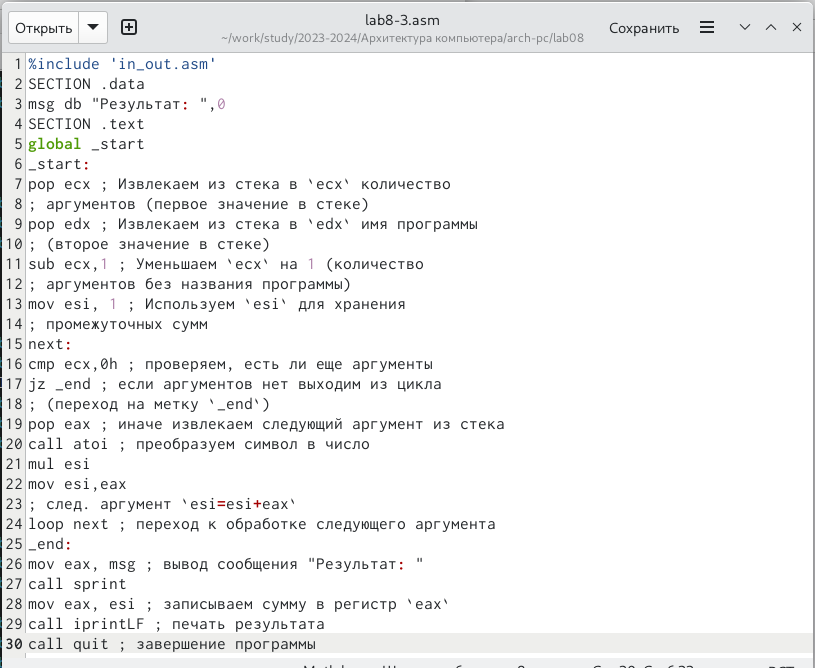


Figure 14: Изменение текста программы

Создаю исполняемый файл и запускаю его, указав аргументы. (рис. [15](#fig:015)).

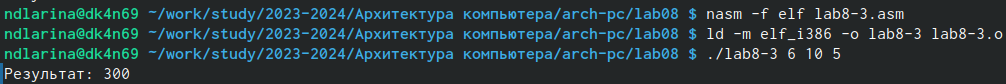


Figure 15: Запуск исполняемого файла

## 4.3 Задание для самостоятельной работы

Создаю файл lab8-task1.asm в каталоге ~/work/study/2023-2024/Архитектура компьютера/arch-pc/lab08. (рис. [16](#fig:016)).

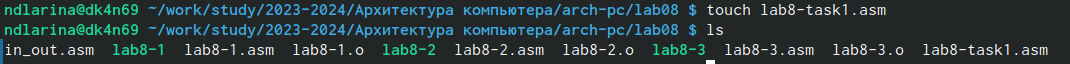


Figure 16: Создание файла lab8-task1.asm

Пишу текст программы, которая находит сумму значений функции f(x) = 4\*x - 3 в соответствии с моим номером варианта (6) для x = x1, x2, …, xn. Значения xi передаются как аргументы. (рис. [17](#fig:017)).

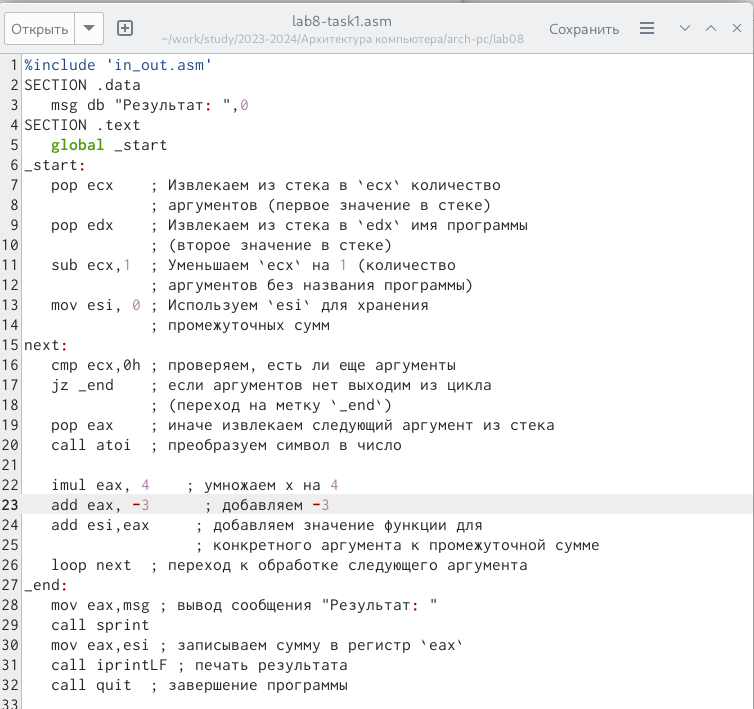


Figure 17: Текст программы

Создаю исполняемый файл и проверьте его работу на нескольких наборах x = x1, x2, …, xn. (рис. [18](#fig:018)).

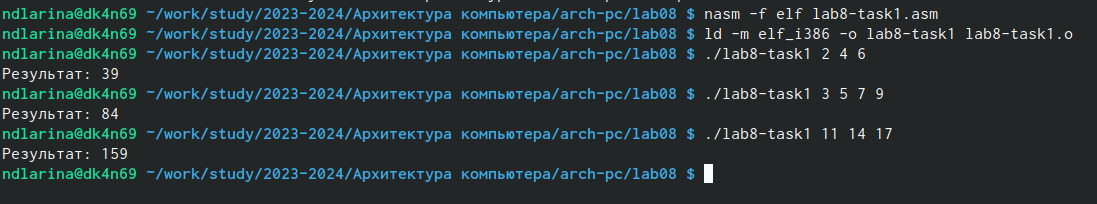


Figure 18: Запуск исполняемого файла и проверка его работы

Программа работает корректно.

# 5 Выводы

В ходе работы над данной лабораторной работой мне удалось приобрести навыки написания программ использованием циклов и обработкой аргументов командной строки, что поможет мне при выполнении последующих лабораторных работ.

# 6 Список литературы

1. [Лабораторная работа №8](https://esystem.rudn.ru/mod/resource/view.php?id=1030556)