Отчет по лабораторной работе №8

Дисциплина: архитектура компьютера

Гомазкова Алина

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 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. 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.

Команда push размещает значение в стеке, т.е. помещает значение в ячейку памяти, на которую указывает регистр esp, после этого значение регистра esp увеличивается на 4. Данная команда имеет один операнд — значение, которое необходимо поместить в стек.

Команда pop извлекает значение из стека, т.е. извлекает значение из ячейки памяти, на которую указывает регистр esp, после этого уменьшает значение регистра esp на 4. У этой команды также один операнд, который может быть регистром или переменной в памяти. Нужно помнить, что извлечённый из стека элемент не стирается из памяти и остаётся как “мусор”, который будет перезаписан при записи нового значения в стек.

Для организации циклов существуют специальные инструкции. Для всех инструкций максимальное количество проходов задаётся в регистре ecx. Наиболее простой является инструкция loop. Она позволяет организовать безусловный цикл.

# 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)

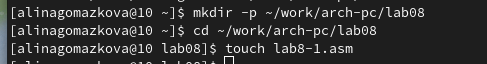


Рис. 1 Создание файлов для лабораторной работы

Ввожу в файл lab8-1.asm текст программы из листинга 8.1 (рис. 2)

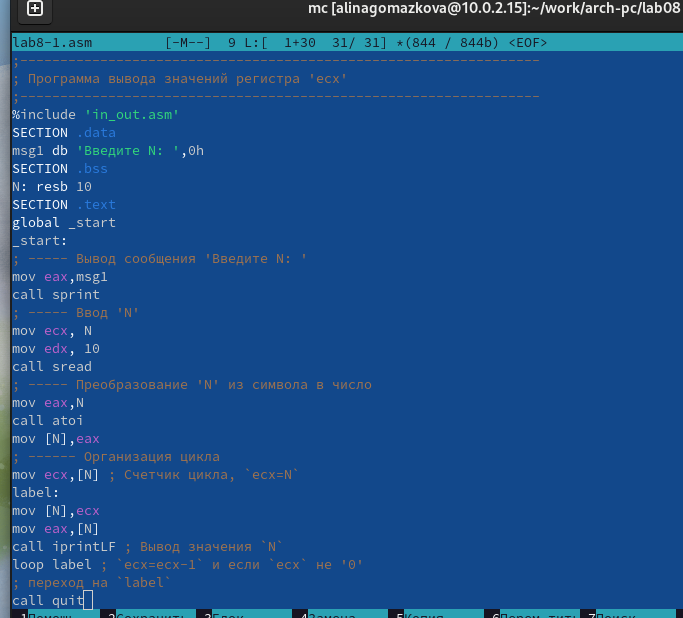


Рис. 2 Ввод текста из листинга 8.1

Создаю исполняемый файл и проверяю его работу (рис. 3)

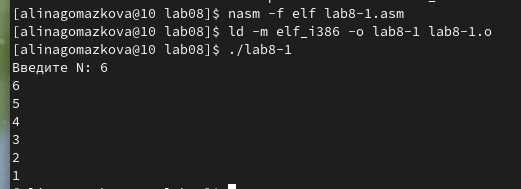


Рис. 3 Запуск исполняемого файла

Данная программа выводит числа от N до 1 включительно. Изменяю текст программы, добавив изменение значения регистра ecx в цикле (рис. 4)

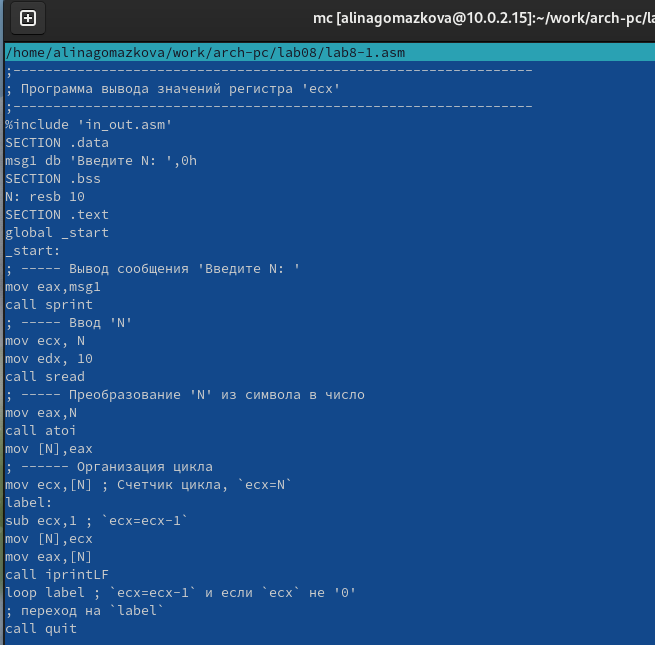


Рис. 4 Изменение текста программы

Создаю исполняемый файл и проверяю его работу (рис. 5)



Рис. 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)

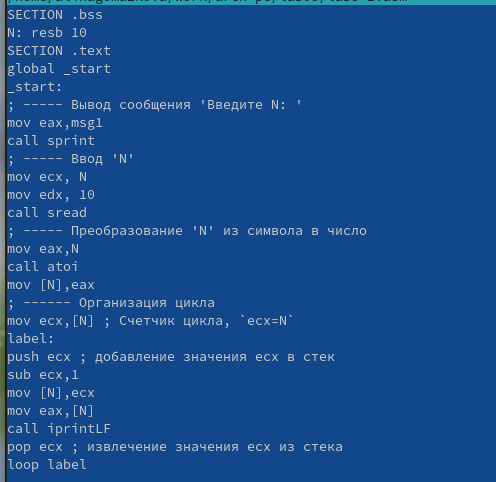


Рис. 6 Изменение текста программы

Создаю исполняемый файл и проверяю его работу (рис. 7)

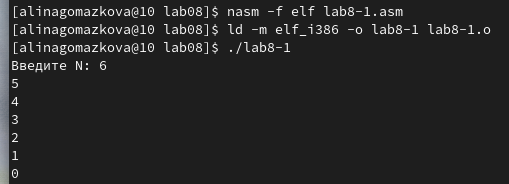


Рис. 7 Запуск исполняемого файла

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 и ввожу в него текст программы из листинга 8.2 (рис. 8)

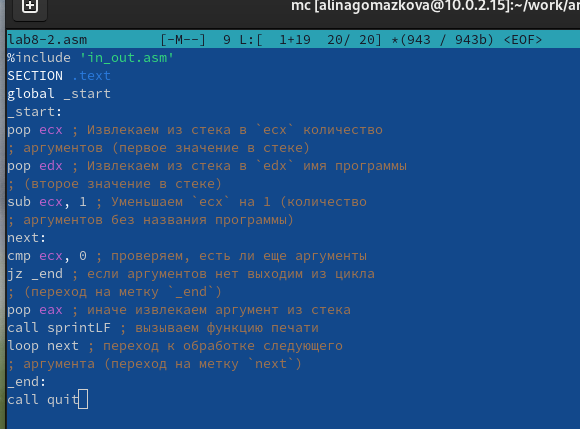


Рис. 8 Ввод текста программы из листинга 8.2

Создаю исполняемый файл и запускаю его, указав нужные аргументы (рис. 9)

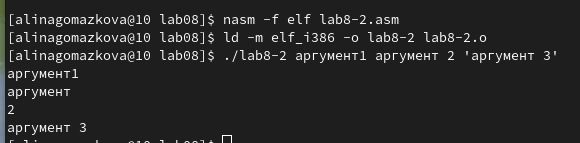


Рис. 9 Запуск исполняемого файла

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 и ввожу в него текст программы из листинга 8.3 (рис. 10)

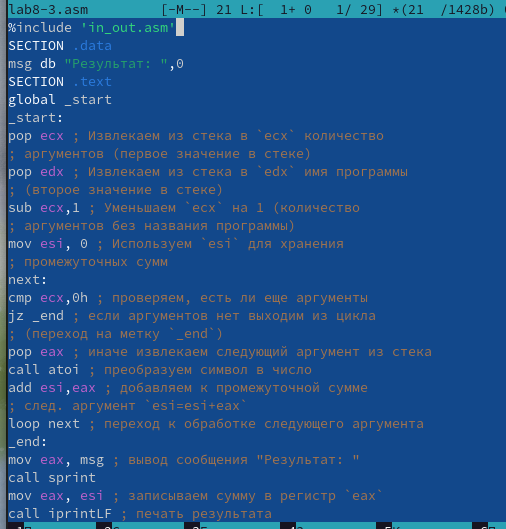


Рис. 10 Ввод текста программы из листинга 8.3

Создаю исполняемый файл и запускаю его, указав аргументы (рис. 11)

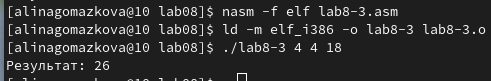


Рис. 11 Запуск исполняемого файла

Изменяю текст программы из листинга 8.3 для вычисления произведения аргументов командной строки (рис. 12)

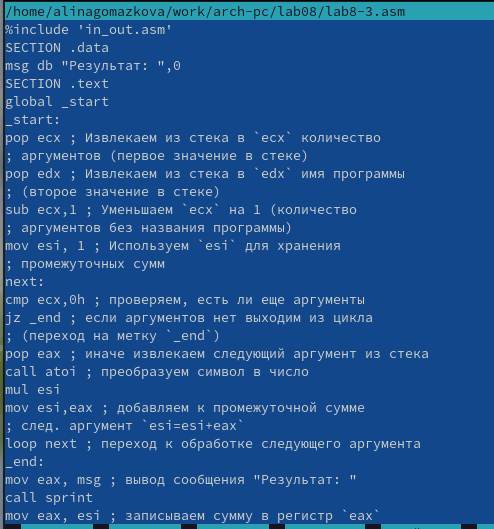


Рис. 12 Изменение текста программы

Создаю исполняемый файл и запускаю его, указав аргументы (рис. 13)

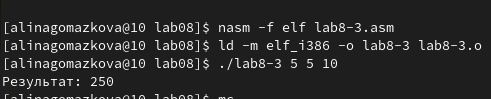


Рис. 13 Запуск исполняемого файла

## 4.3 Задание для самостоятельной работы

Пишу текст программы, которая находит сумму значений функции f(x) = 10x - 4 в соответствии с моим номером варианта (9) для x = x1, x2, …, xn. Значения xi передаются как аргументы (рис. 14)

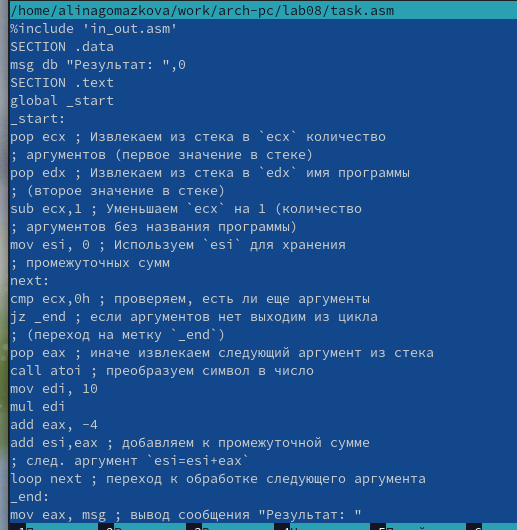


Рис. 14 Текст программы

Создаю исполняемый файл и проверьте его работу на нескольких наборах x = x1, x2, …, xn (рис. 15)

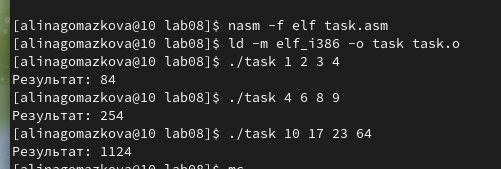


Рис. 15 Запуск исполняемого файла и проверка его работы

# 5 Выводы

Благодаря данной лабораторной работе я приобрела навыки написания программ использованием циклов и обработкой аргументов командной строки, что поможет мне при выполнении последующих лабораторных работ.

# Список литературы

1.[Архитектура ЭВМ](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)