Отчёт по лабораторной работе №8

Дисциплина: архитектура компьютера

Байдина Елизавета Дмитриевна

Содержание

# 1 Цель работы

Приобрести навыки 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.

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.

Иструкция loop выполняется в два этапа. Сначала из регистра ecx вычитается единица и его значение сравнивается с нулём. Если регистр не равен нулю, то выполняется переход к указанной метке. Иначе переход не выполняется и управление передаётся команде, которая следует сразу после команды loop.

При разработке программ иногда встает необходимость указывать аргументы, которые будут использоваться в программе, непосредственно из командной строки при запуске программы.

При запуске программы в NASM аргументы командной строки загружаются в стек в обратном порядке, кроме того в стек записывается имя программы и общее количество аргументов. Последние два элемента стека для программы, скомпилированной NASM, – это всегда имя программы и количество переданных аргументов.

Таким образом, для того чтобы использовать аргументы в программе, их просто нужно извлечь из стека. Обработку аргументов нужно проводить в цикле. Т.е. сначала нужно извлечь из стека количество аргументов, а затем циклично для каждого аргумента выполнить логику программы.

# 4 Выполнение лабораторной работы

## 4.1 Реализация циклов в NASM

Создаю каталог для программам лабораторной работы № 8 (рис. 1)

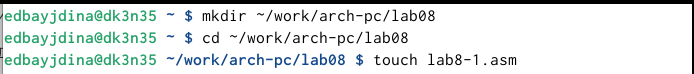


Рис. 1: Создание каталога, переход в него и создание файла

Копирую файл in\_out.asm из загрузок в соответствующую папку для дальнейшей работы (рис. 2)

Рис. 2: Копирование файла

Рис. 2: Копирование файла

Ввожу в lab8-1.asm текст программы из листинга 8.1 (рис. 3)

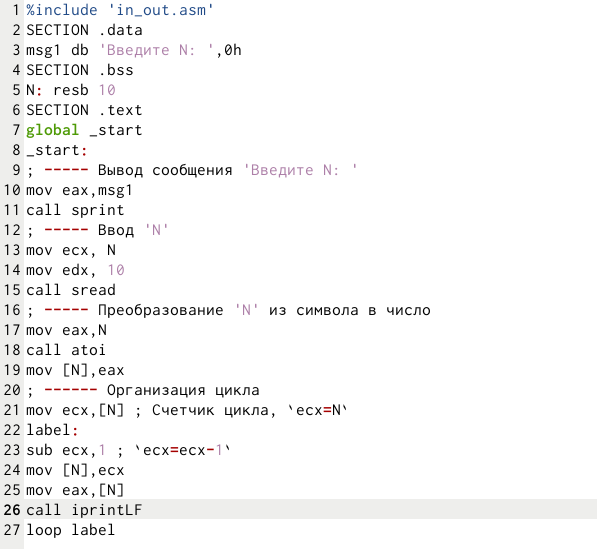


Рис. 3: Текст программы листинга 8.1

Создаю исполняемый файл и запускаю его (рис. 4)

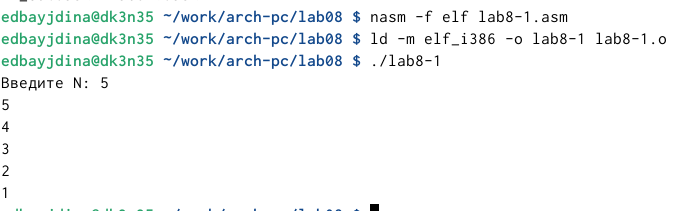


Рис. 4: Создание и запуск файла

Изменение текста программы из листинга 8.1 (рис. 5)

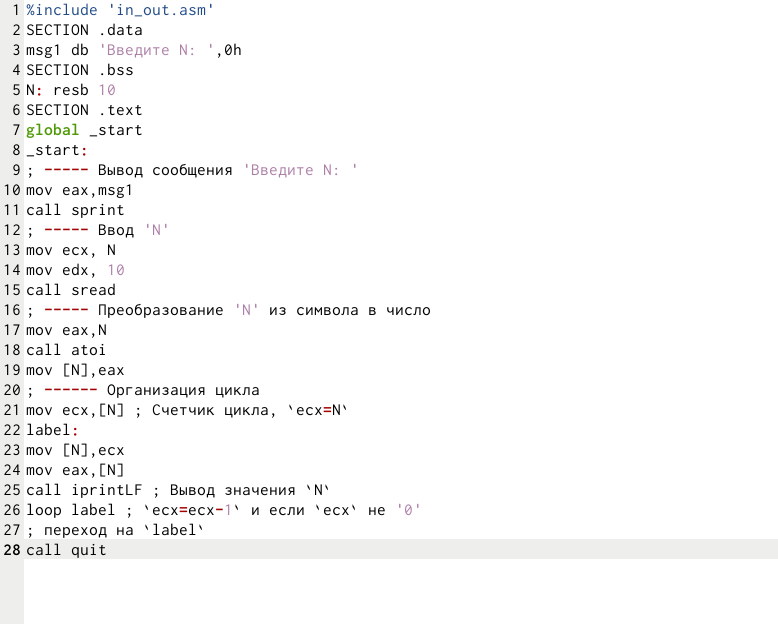


Рис. 5: Изменение текста программы из листинга 8.1

Создаю изменённый исполняемый файл и запускаю его (рис. 6)

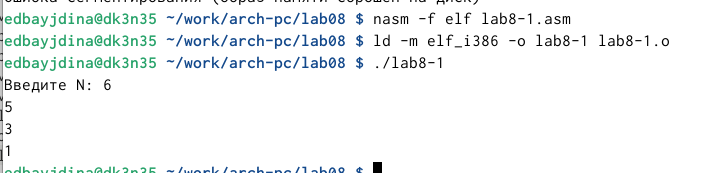


Рис. 6: Запуск изменённого файла

Изменяю текст программы так, чтобы программа работала корректно (рис. 7)

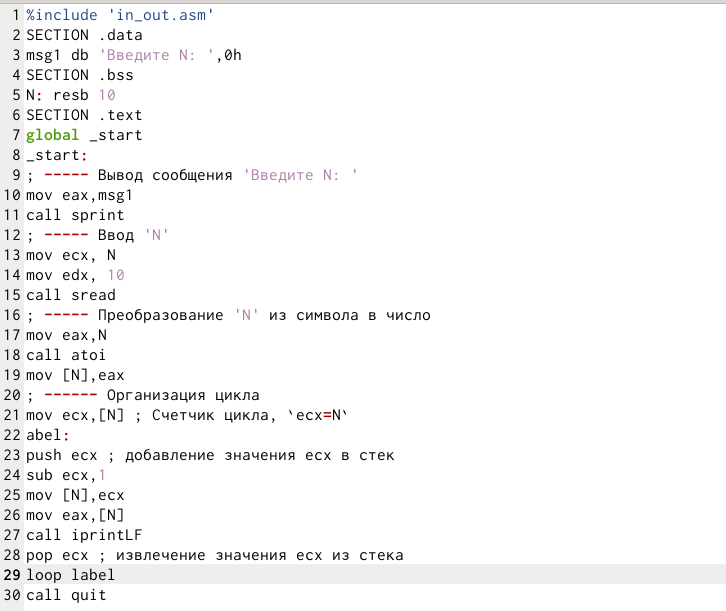


Рис. 7: Изменение текста программы

Создаю исполняемый файл и запускаю его (рис. 8)

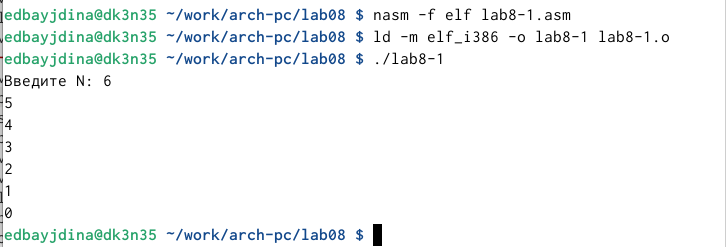


Рис. 8: Создание файла и запуск

## 4.2 Обработка аргументов командной строки

Создаю файл lab8-2.asm (рис. 9)

Рис. 9: Создание файла

Рис. 9: Создание файла

Ввожу текст программы из листинга 8.2 в файл (рис. 10)

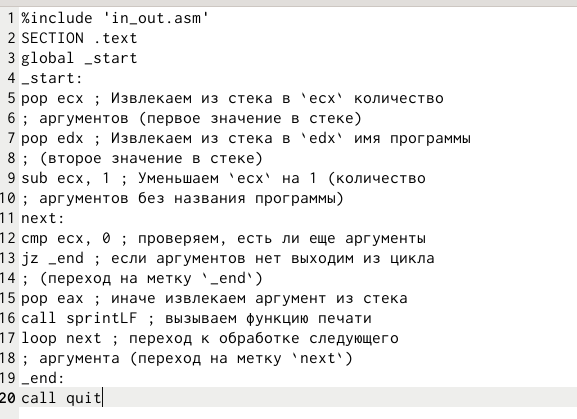


Рис. 10: Текст программы листинга 8.2

Создаю исполняемый файл и запускаю его (рис. 11)

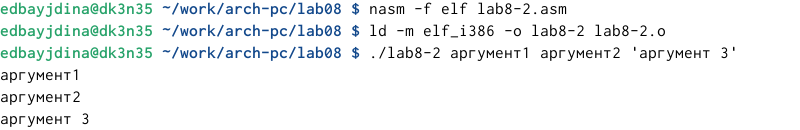


Рис. 11: Создание файла и запуск

Создаю файл lab8-3.asm(рис. 12)

Рис. 12: Создание файла

Рис. 12: Создание файла

Ввожу текст программы из листинга 8.3 в файл (рис. 13)

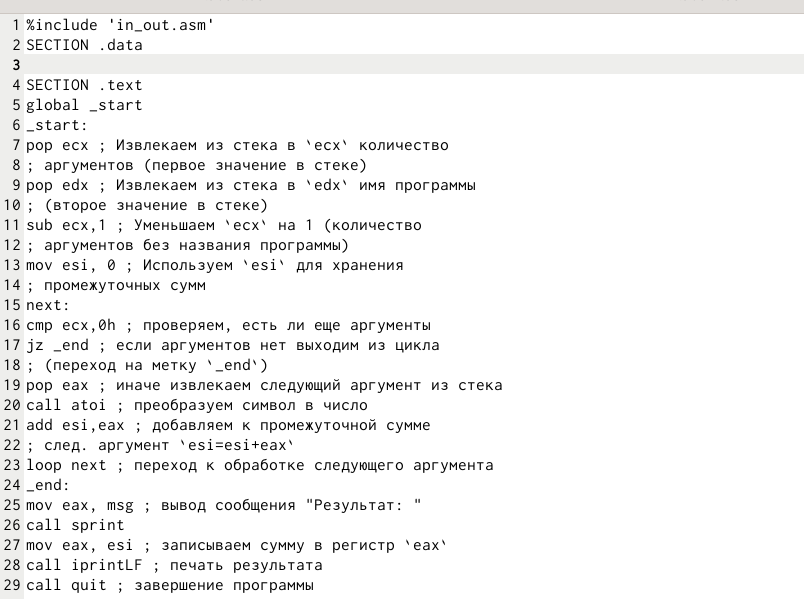


Рис. 13: Текст пограммы из листинга 8.3

Создаю исполняемый файл и запускаю его.И вывожу сумма аргументов (рис. 14)

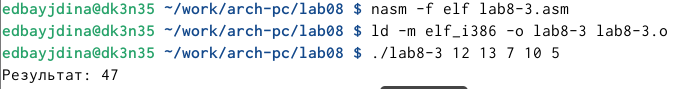


Рис. 14: Создание файла и запуск

Изменяю текст рограммы так,чтобы она выводла произвдение аргументов (рис. 15)

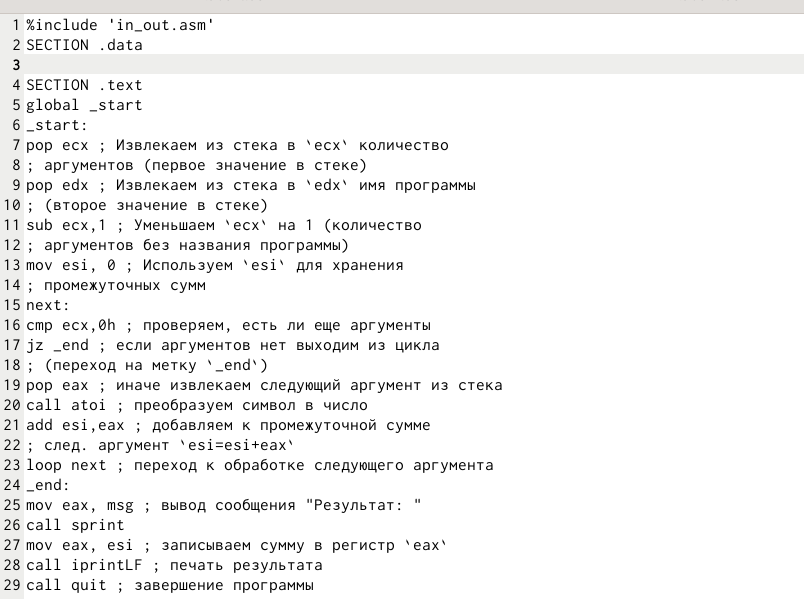


Рис. 15: Изменение программы

Создаю исполняемый файл и запускаю его (рис. 16)

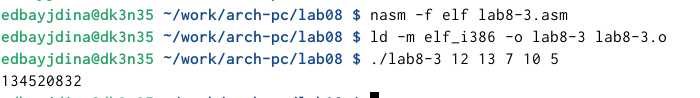


Рис. 16: Создание файла и запуск

## 4.3 Задание для самостоятельной работы

Создаю файл, для программы в рамках смостоятельной работы (рис. 17)

Рис. 17: Создание файла

Рис. 17: Создание файла

Пишу программу для своего тринадцатого варианта (рис. 18)

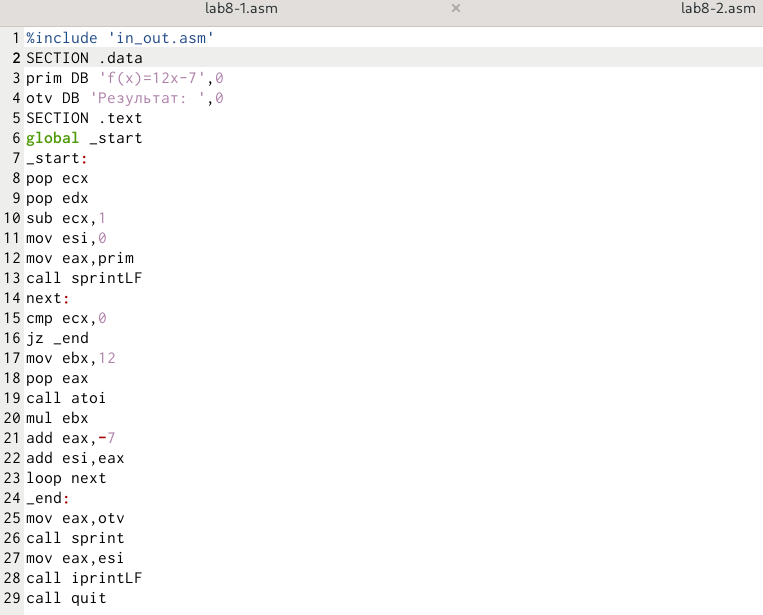


Рис. 18: Написание программы

Запускаю исполняемый файл (рис. 19)

Рис. 19: Запуск файла

Рис. 19: Запуск файла

Запускаю исполняемый файл (рис. 20)

Рис. 20: Запуск файла

Рис. 20: Запуск файла

# 5 Выводы

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