Отчёт по лабораторной работе 8

Архитектура компьютеров и операционные системы

Плетяго Кирилл НММбд-03-23

Содержание

# 1 Цель работы

Целью работы является приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки..

# 2 Выполнение лабораторной работы

Создал каталог для программам лабораторной работы № 8 и файл lab8-1.asm

При реализации циклов в NASM с использованием инструкции loop необходимо помнить о том, что эта инструкция использует регистр ecx в качестве счетчика и на каждом шаге уменьшает его значение на единицу. В качестве примера рассмотрим программу, которая выводит значение регистра ecx.

Написал в файл lab8-1.asm текст программы из листинга 8.1. (рис. [[1](#fig:001)]) Создал исполняемый файл и проверил его работу. (рис. [[2](#fig:002)])

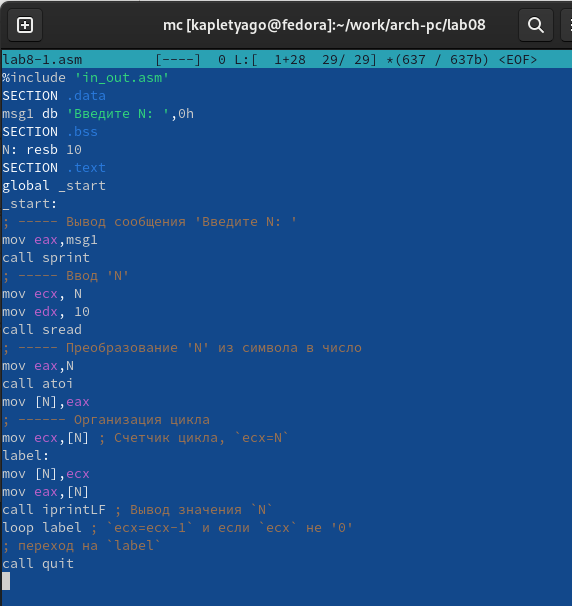


Figure 1: Программа в файле lab8-1.asm



Figure 2: Запуск программы lab8-1.asm

Данный пример показывает, что использование регистра ecx в теле цилка loop может привести к некорректной работе программы. Изменил текст программы добавив изменение значение регистра ecx в цикле. (рис. [[3](#fig:003)])

Программа запускает бесконечный цикл при нечетном N и выводит только нечетные числа при четном N. (рис. [[4](#fig:004)])

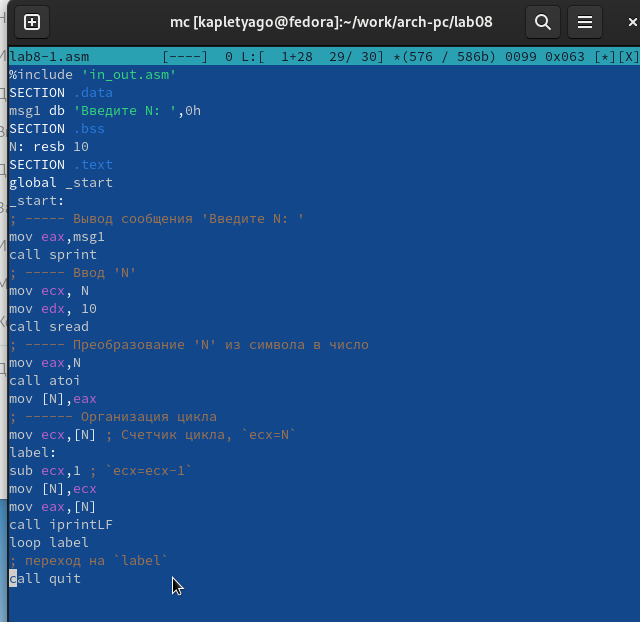


Figure 3: Программа в файле lab8-1.asm

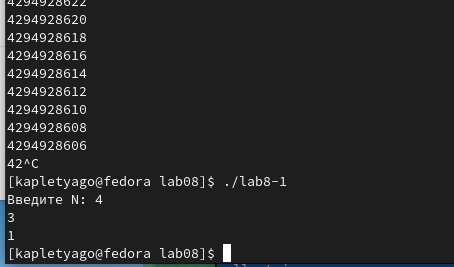


Figure 4: Запуск программы lab8-1.asm

Для использования регистра ecx в цикле и сохранения корректности работы программы можно использовать стек. Внес изменения в текст программы добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop. (рис. [[5](#fig:005)])

Создал исполняемый файл и проверьте его работу. Программа выводит числа от N-1 до 0, число проходов цикла соответсвует N.(рис. [[6](#fig:006)])

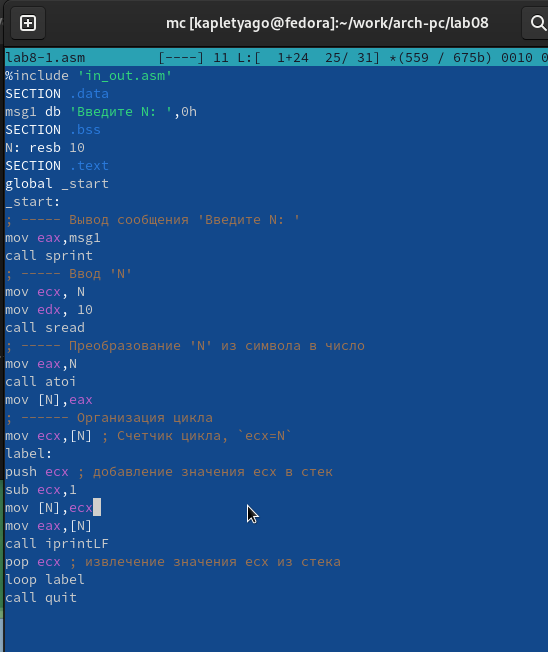


Figure 5: Программа в файле lab8-1.asm

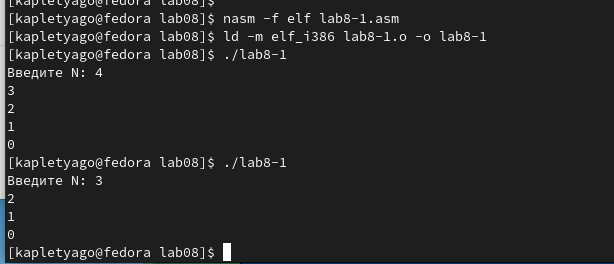


Figure 6: Запуск программы lab8-1.asm

Создал файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и ввел в него текст программы из листинга 8.2. (рис. [[7](#fig:007)])

Создал исполняемый файл и запустил его, указав аргументы. Программа обработала 5 аргументов. Аргументами считаются слова/числа, разделенные пробелом. (рис. [[8](#fig:008)])

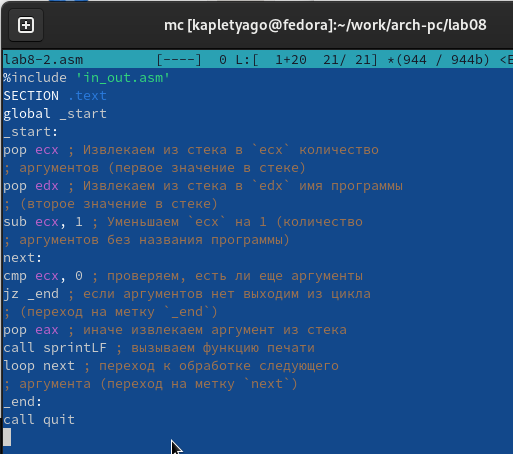


Figure 7: Программа в файле lab8-2.asm

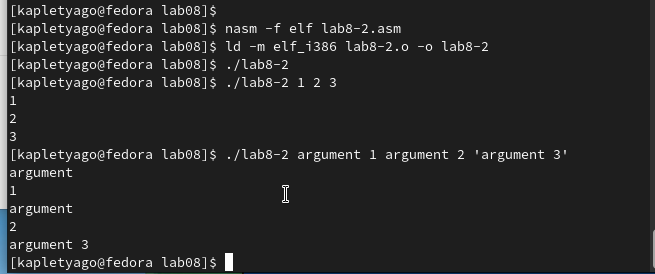


Figure 8: Запуск программы lab8-2.asm

Рассмотрим еще один пример программы которая выводит сумму чисел, которые передаются в программу как аргументы. (рис. [[9](#fig:009)]) (рис. [[10](#fig:010)])

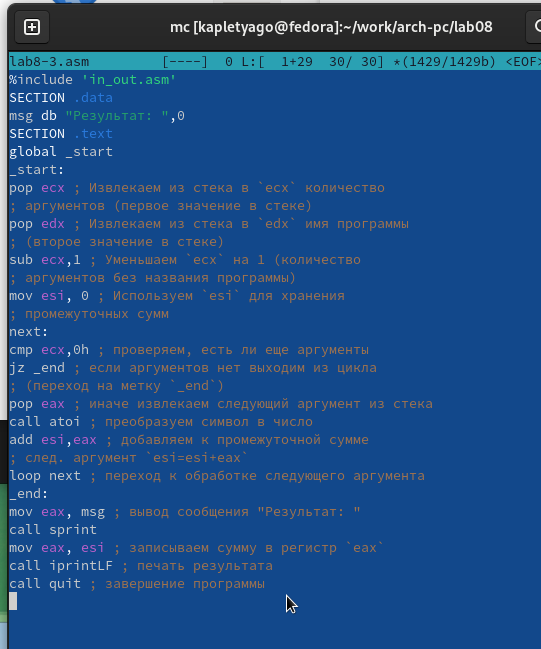


Figure 9: Программа в файле lab8-3.asm

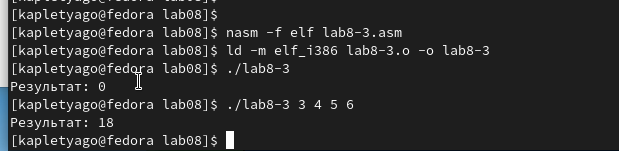


Figure 10: Запуск программы lab8-3.asm

Изменл текст программы из листинга 8.3 для вычисления произведения аргументов командной строки. (рис. [[11](#fig:011)]) (рис. [[12](#fig:012)])

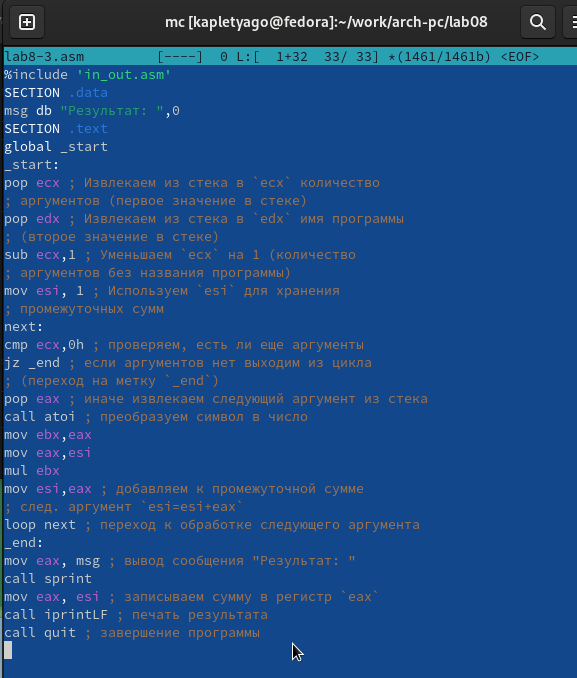


Figure 11: Программа в файле lab8-3.asm

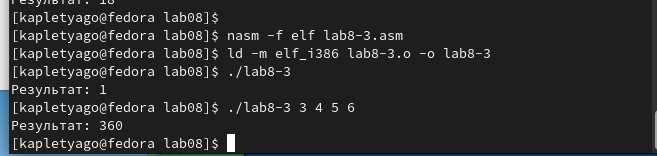


Figure 12: Запуск программы lab8-3.asm

## 2.1 Самостоятельное задание

Напишите программу, которая находит сумму значений функции для , т.е. программа должна выводить значение . Значения передаются как аргументы. Вид функции выбрать из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах .(рис. [[13](#fig:013)]) (рис. [[14](#fig:014)])

для варианта 8

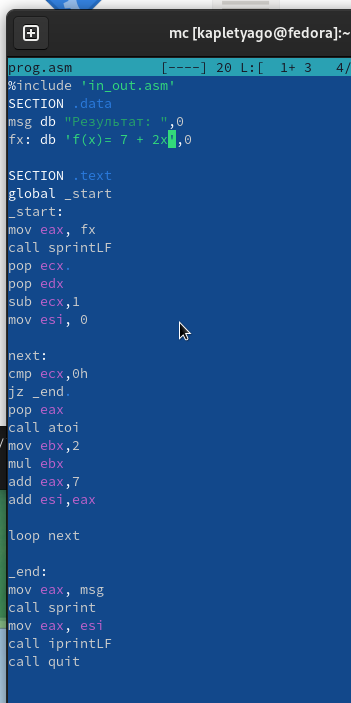


Figure 13: Программа в файле prog.asm

Для проверки я запустил сначала с одним аргументом. Так, при подстановке

Затем подал несколько аргументов и получил сумму значений функции.

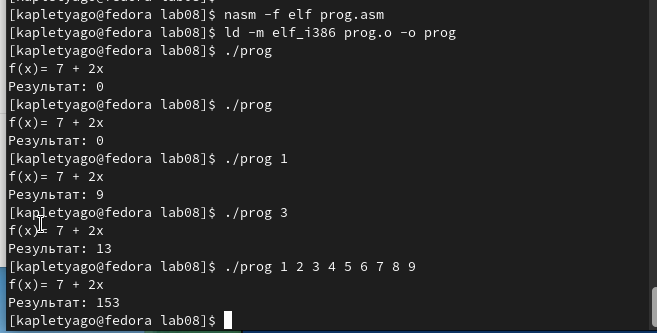


Figure 14: Запуск программы prog.asm

# 3 Выводы

Освоили работы со стеком, циклом и аргументами на ассемблере nasm.