Отчёт по лабораторной работе 8

Архитектура компьютеров

Шамес Эддин Хамза НКА-06-24

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

## 2.1 Реализация циклов в NASM

Создал каталог для программ лабораторной работы №8 и файл lab8-1.asm (рис. 1).

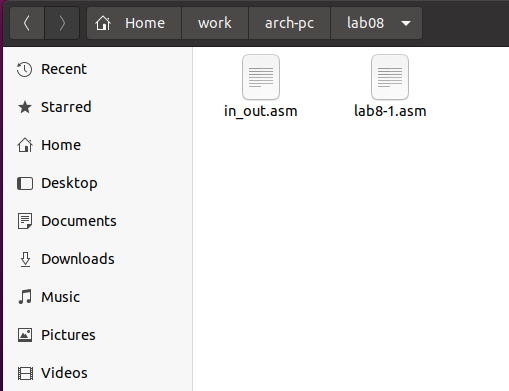


Рис. 1: Создан каталог

При реализации циклов в NASM с использованием инструкции loop необходимо помнить, 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.

Записал в файл lab8-1.asm текст программы из листинга 8.1 (рис. 2). Создал исполняемый файл и проверил его работу (рис. 3).

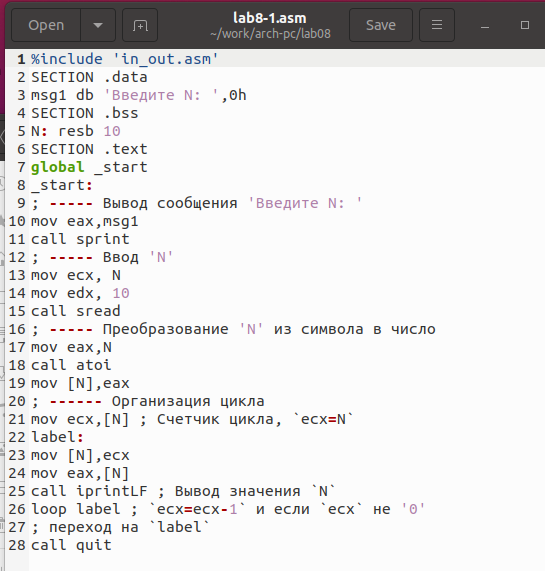


Рис. 2: Программа lab8-1.asm

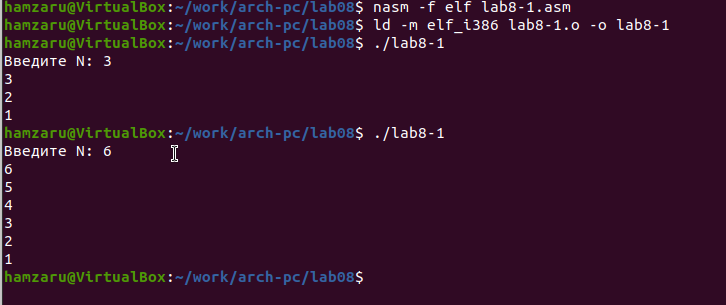


Рис. 3: Запуск программы lab8-1.asm

Данный пример демонстрирует, что изменение значения регистра ecx в теле цикла loop может привести к некорректной работе программы. Изменил текст программы, добавив изменение значения регистра ecx в цикле (рис. 4). Программа запускает бесконечный цикл при нечетном N и выводит только нечетные числа при четном N (рис. 5).

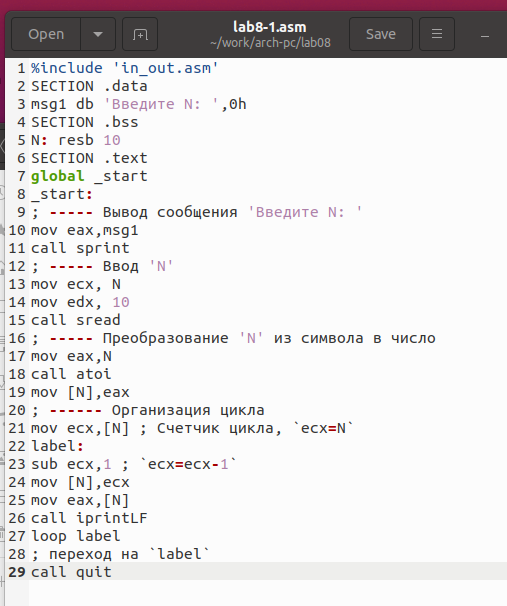


Рис. 4: Программа lab8-1.asm

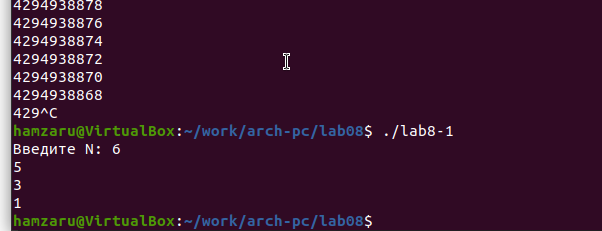


Рис. 5: Запуск программы lab8-1.asm

Для корректного использования регистра ecx в цикле можно использовать стек. Внес изменения в текст программы, добавив команды push и pop для сохранения и восстановления значения счетчика цикла loop (рис. 6). Создал исполняемый файл и проверил его работу (рис. 7). Программа корректно выводит числа от N-1 до 0, при этом число проходов цикла соответствует N.

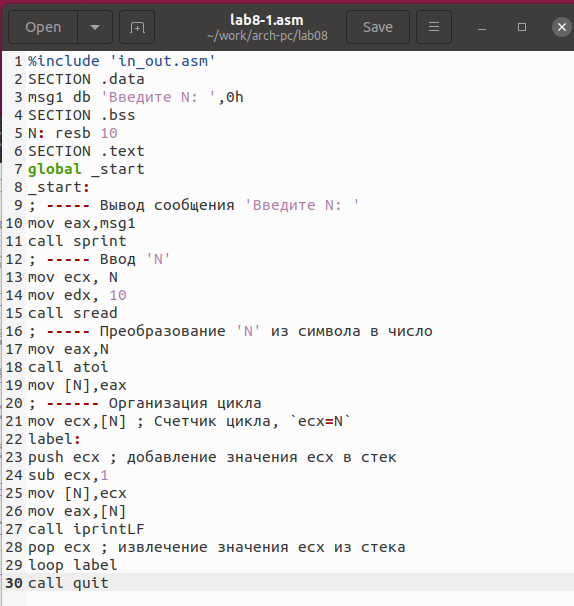


Рис. 6: Программа lab8-1.asm

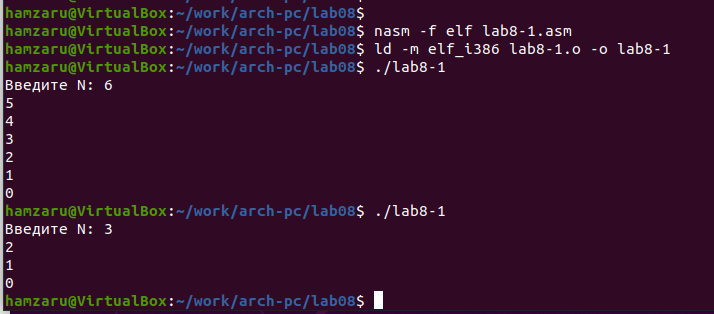


Рис. 7: Запуск программы lab8-1.asm

Создал файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и записал в него текст программы из листинга 8.2 (рис. 8). Скомпилировал исполняемый файл и запустил его с указанием аргументов. Программа обработала 4 аргумента — слова или числа, разделенные пробелом (рис. 9).

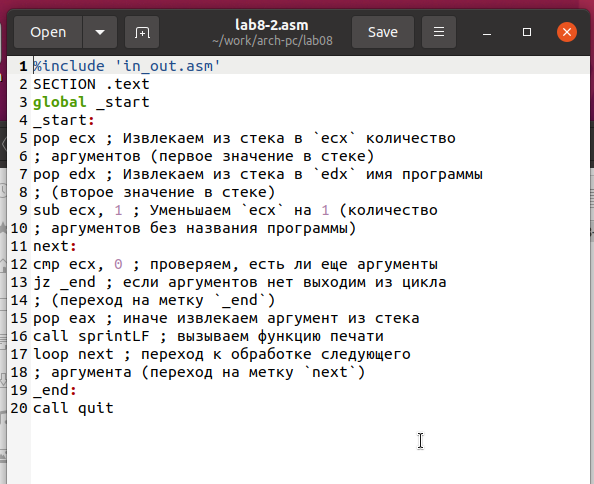


Рис. 8: Программа lab8-2.asm

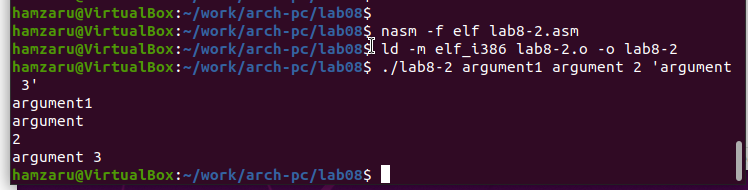


Рис. 9: Запуск программы lab8-2.asm

Рассмотрел пример программы, которая вычисляет сумму чисел, переданных в программу в качестве аргументов (рис. 10, рис. 11).

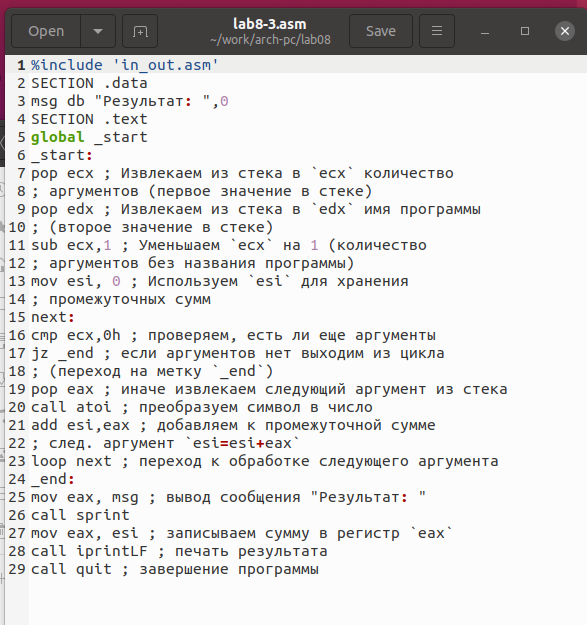


Рис. 10: Программа lab8-3.asm

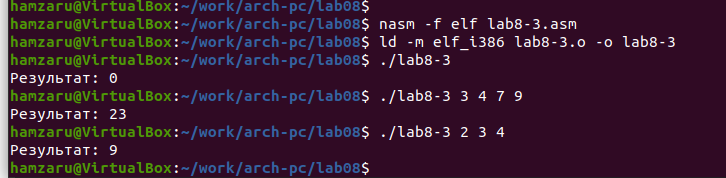


Рис. 11: Запуск программы lab8-3.asm

Изменил текст программы из листинга 8.3 для вычисления произведения аргументов командной строки (рис. 12, рис. 13).

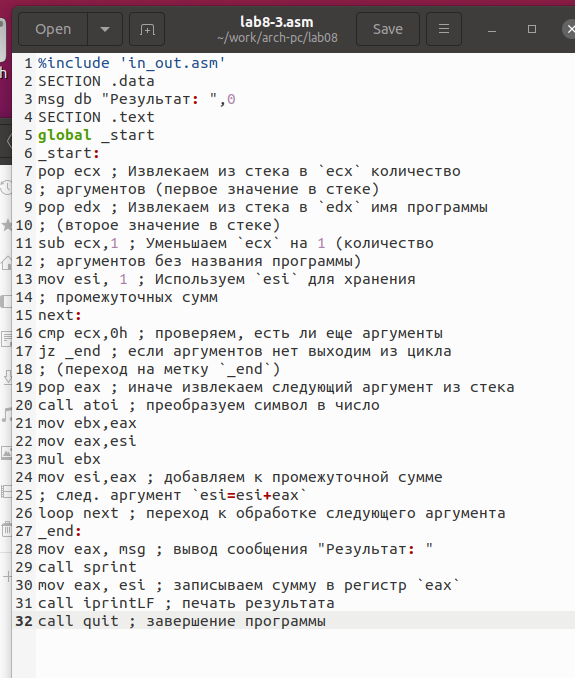


Рис. 12: Программа lab8-3.asm

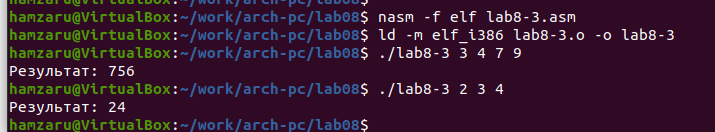


Рис. 13: Запуск программы lab8-3.asm

## 2.2 Самостоятельное задание

Написал программу, которая вычисляет сумму значений функции для , где значения передаются как аргументы. Функция выбрана из таблицы 8.1 в соответствии с вариантом 6:

Программа корректно работает, выводя сумму значений . Создал исполняемый файл и проверил его работу на нескольких наборах (рис. 14, рис. 15).

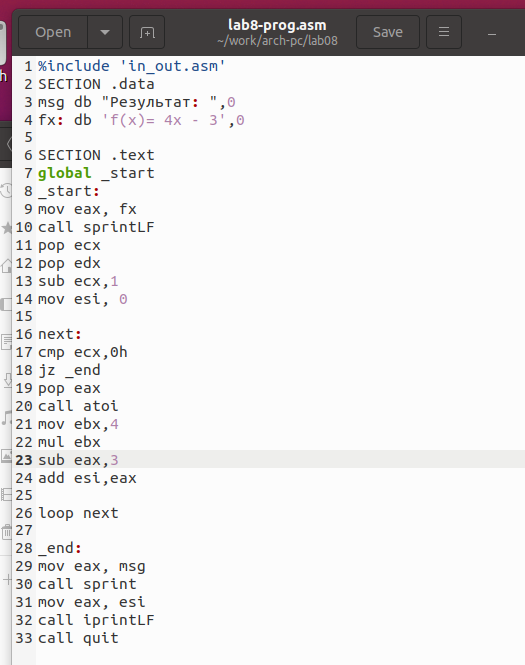


Рис. 14: Программа lab8-prog.asm

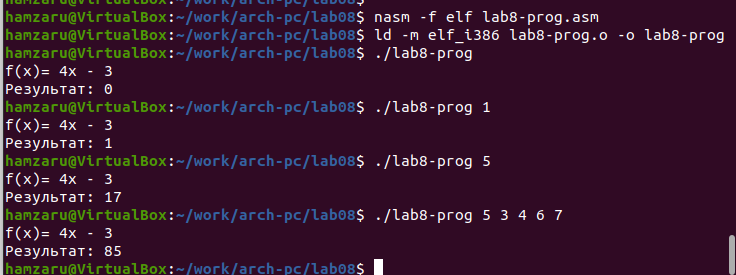


Рис. 15: Запуск программы lab8-prog.asm

Программа правильно считает, например, , .

# 3 Выводы

В ходе работы освоил использование стека, инструкции loop и работу с аргументами командной строки в языке ассемблера NASM.