Отчёт по лабораторной работе 8

Архитектура компьютера

Эргешов Байрам НКАбд-02-23

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

ыл создан каталог для проведения лабораторной работы N8, а также был создан файл с именем lab8-1.asm.

При использовании инструкции loop в NASM для организации циклов, нужно помнить следующее: эта инструкция применяет регистр ecx в качестве счётчика и на каждой итерации уменьшает его значение на единицу. Для лучшего понимания этого процесса давайте рассмотрим пример программы, которая выводит значение регистра ecx.

Я написал в файл lab8-1.asm текст программы из листинга 8.1. Затем создал исполняемый файл и проверил его работу.

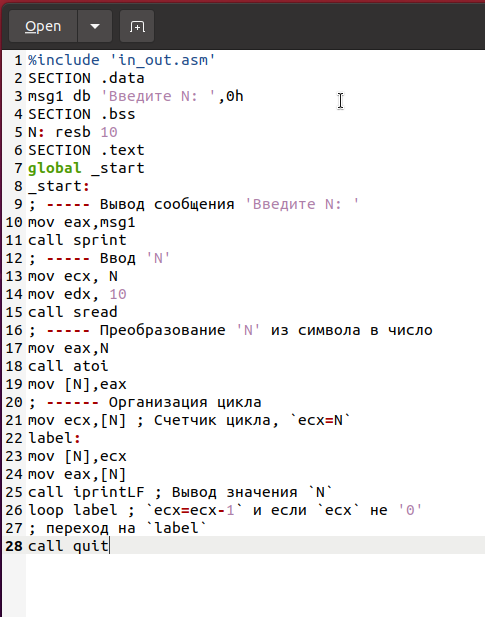


Figure 1: Код программы lab8-1.asm

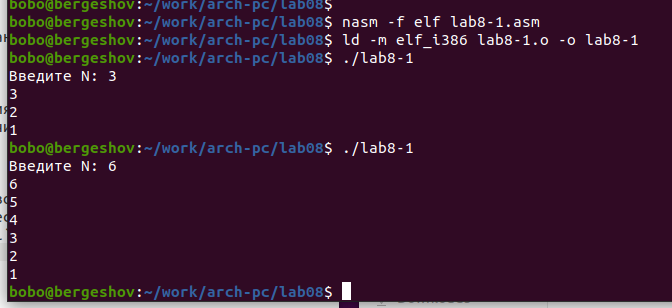


Figure 2: Компиляция и запуск программы lab8-1.asm

В этом примере показано, что применение регистра ecx в инструкции loop может вызвать неправильную работу программы. В текст программы я внёс изменения, заключающиеся в модификации значения регистра ecx внутри цикла.

Теперь эта программа запускает бесконечный цикл при нечётном значении N и выводит только нечётные числа при чётном значении N.

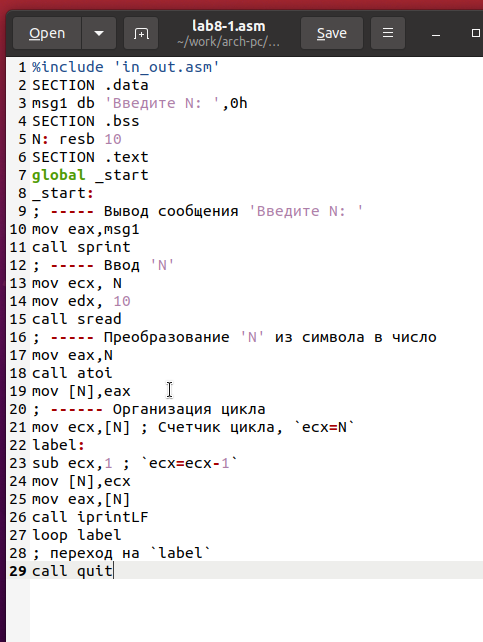


Figure 3: Код программы lab8-1.asm

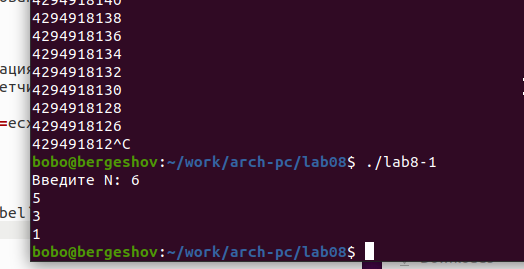


Figure 4: Компиляция и запуск программы lab8-1.asm

Для того, чтобы применить регистр ecx в цикле и обеспечить правильность работы программы, можно использовать стек. Я внёс изменения в текст программы, добавив команды push и pop для сохранения значения счётчика цикла loop в стеке.

Был создан исполняемый файл и проверена его работа. Программа выводит числа от N-1 до 0, где количество проходов цикла соответствует значению N.

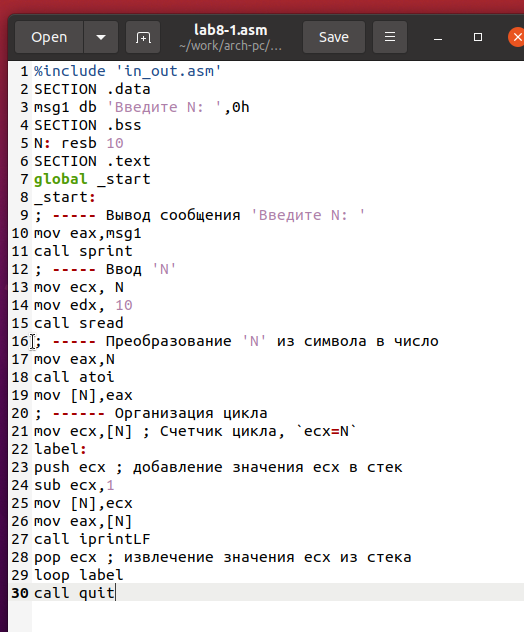


Figure 5: Код программы lab8-1.asm



Figure 6: Компиляция и 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.

Затем создал исполняемый файл и запустил его, указав аргументы. Программа обработала 5 аргументов. Аргументами считаются слова/числа, разделённые пробелом.

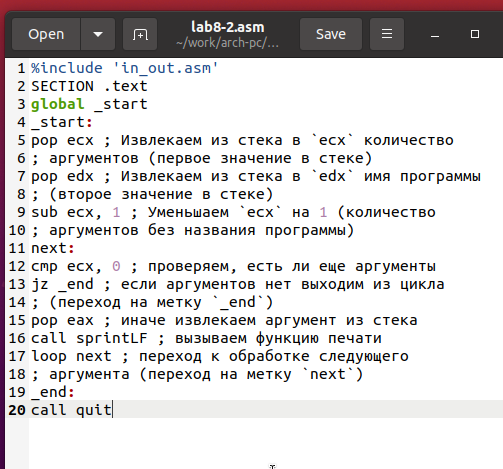


Figure 7: Код программы lab8-2.asm

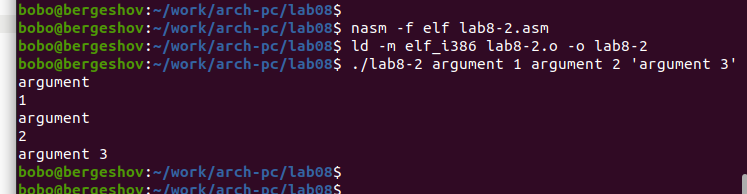


Figure 8: Компиляция и запуск программы lab8-2.asm

Рассмотрим еще один пример программы которая выводит сумму чисел, которые передаются в программу как аргументы.

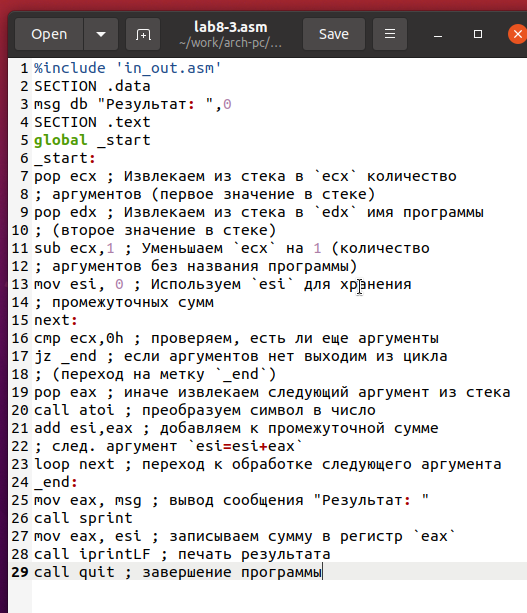


Figure 9: Код программы lab8-3.asm

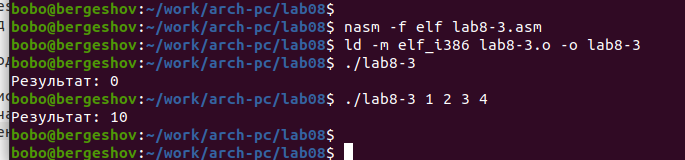


Figure 10: Компиляция и запуск программы lab8-3.asm

Изменл текст программы из листинга 8.3 для вычисления произведения аргументов командной строки.

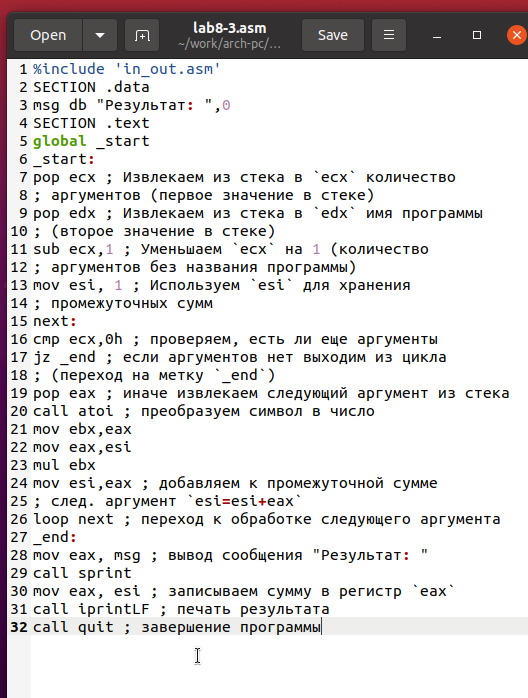


Figure 11: Код программы lab8-3.asm

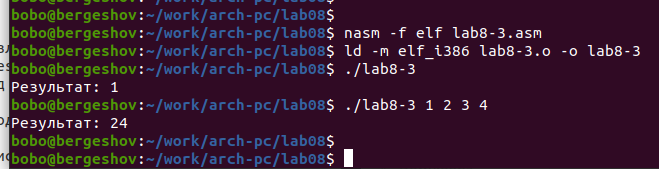


Figure 12: Компиляция и запуск программы lab8-3.asm

## 2.2 Задание для самостоятельной работы

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 .

Мой вариант 1:

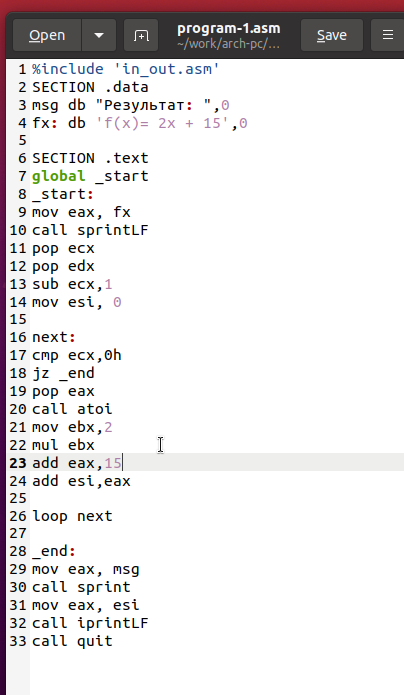


Figure 13: Код программы program-1.asm

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

Затем подал несколько аргументов и получил сумму значений функции.

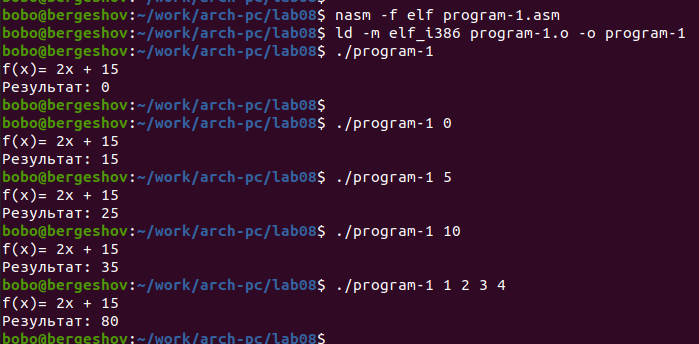


Figure 14: Компиляция и запуск программы program-1.asm

# 3 Выводы

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