Отчёт по лабораторной работе 8

Архитектура компьютера

Диденко Дмитрий Владимирович НПИбд-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

## 2.1 Реализация циклов в NASM

Был создан каталог для выполнения лабораторной работы № 8, а также создан файл с именем lab8-1.asm.

При использовании инструкции loop в NASM для реализации циклов, необходимо помнить о следующем: данная инструкция использует регистр ecx в качестве счетчика и на каждой итерации уменьшает его значение на единицу.

Для лучшего понимания этого процесса, рассмотрим пример программы, 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)])

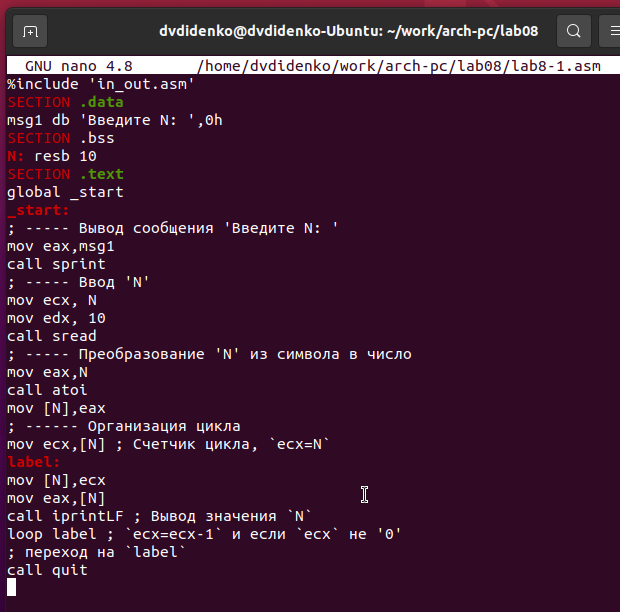


Figure 1: Код программы lab8-1.asm

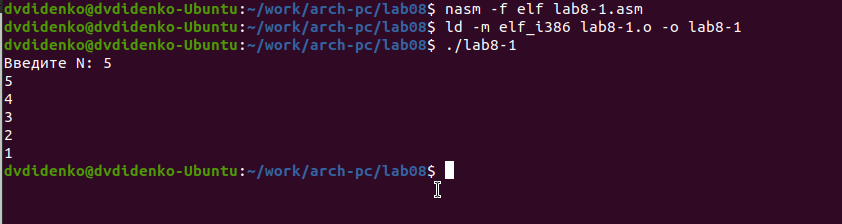


Figure 2: Компиляция и запуск программы lab8-1.asm

В данном примере демонстрируется, что использование регистра ecx в инструкции loop может привести к неправильной работе программы. В тексте программы были внесены изменения, которые включают изменение значения регистра ecx внутри цикла. (рис. [[3](#fig:003)])

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

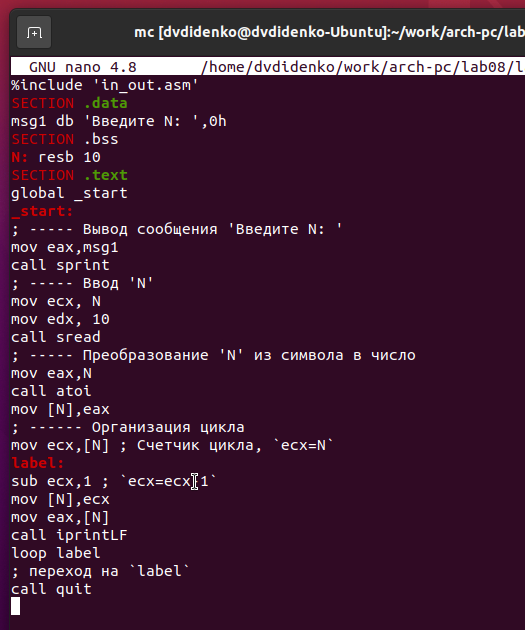


Figure 3: Код программы lab8-1.asm

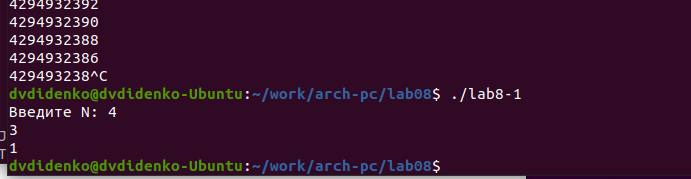


Figure 4: Компиляция и запуск программы lab8-1.asm

Для того чтобы использовать регистр ecx в цикле и обеспечить корректность работы программы, можно применить стек. Внесены изменения в текст программы, добавив команды push и pop для сохранения значения счетчика цикла loop в стеке. (рис. [[5](#fig:005)])

Был создан исполняемый файл и проверена его работа. Программа выводит числа от N-1 до 0, где количество проходов цикла соответствует значению N.(рис. [[6](#fig:006)])

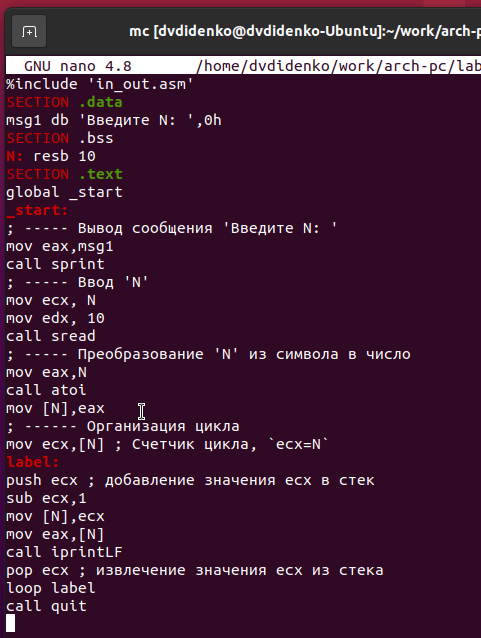


Figure 5: Код программы lab8-1.asm

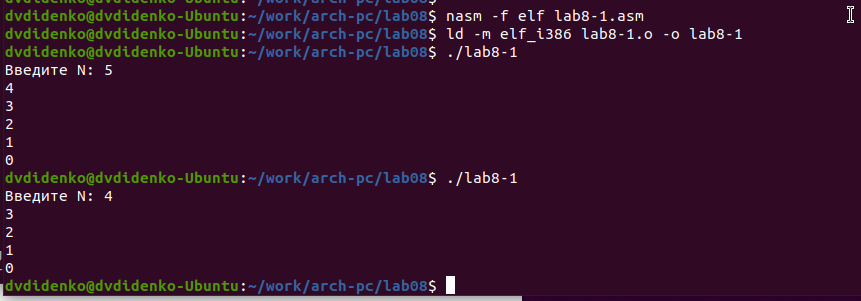


Figure 6: Компиляция и запуск программы lab8-1.asm

## 2.2 Обработка аргументов командной строки

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)])

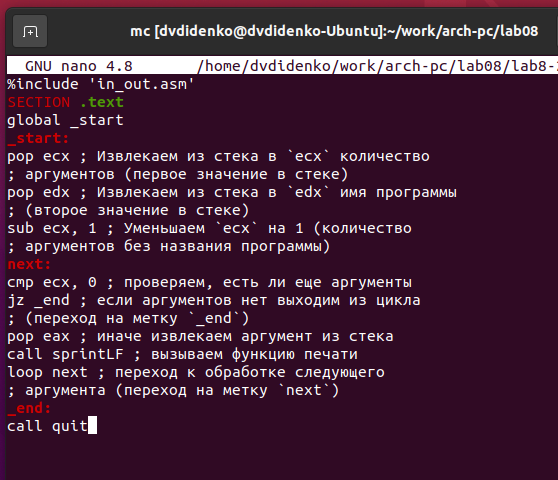


Figure 7: Код программы lab8-2.asm

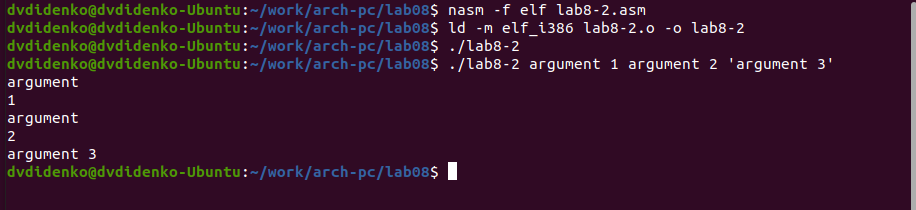


Figure 8: Компиляция и запуск программы lab8-2.asm

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

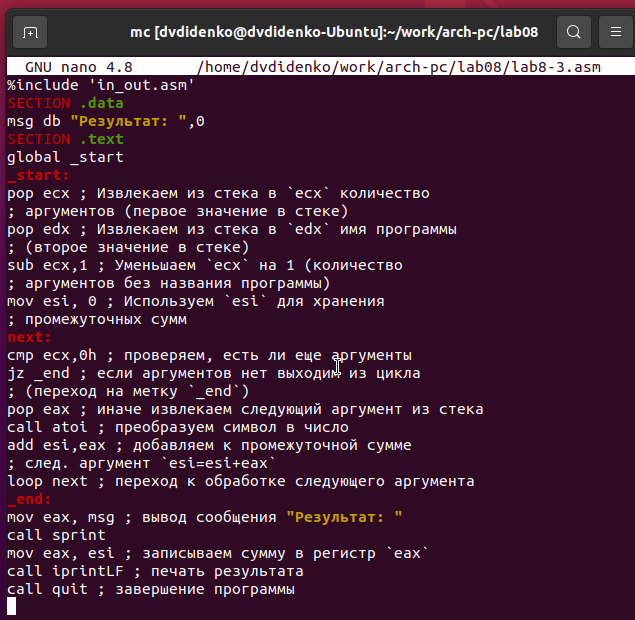


Figure 9: Код программы lab8-3.asm

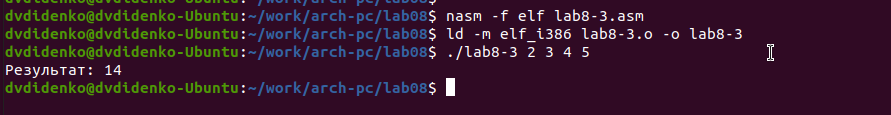


Figure 10: Компиляция и запуск программы lab8-3.asm

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

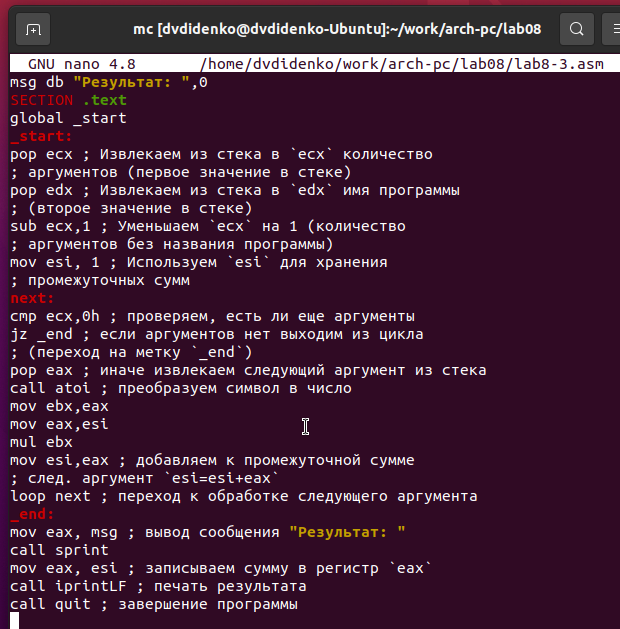


Figure 11: Код программы lab8-3.asm

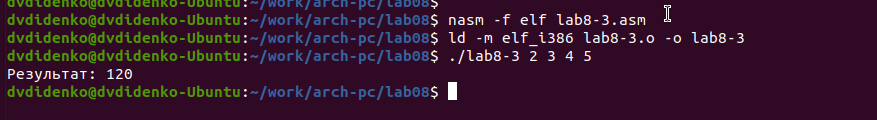


Figure 12: Компиляция и запуск программы lab8-3.asm

## 2.3 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 .(рис. [[13](#fig:013)]) (рис. [[14](#fig:014)])

Мой вариант 13:

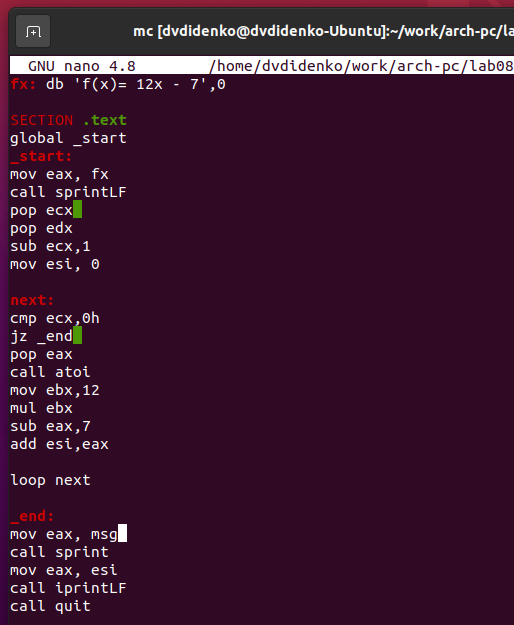


Figure 13: Код программы prog.asm

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

Затем подал несколько аргументов и получил сумму значений функции.

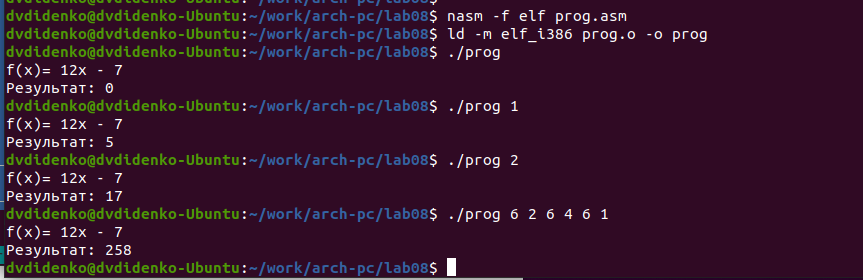


Figure 14: Компиляция и запуск программы prog.asm

# 3 Выводы

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