Отчёт по лабораторной работе №9

Программирование цикла. Обработка аргументов командной строки

Аскеров Александр Эдуардович

Содержание

[1 Цель работы 1](#__RefHeading___Toc237_4251648076)

[2 Выполнение лабораторной работы 1](#__RefHeading___Toc239_4251648076)

[2.1 Реализация циклов в NASM 1](#__RefHeading___Toc241_4251648076)

[2.2 Обработка аргументов командной строки 4](#__RefHeading___Toc243_4251648076)

[2.3 Задание для самостоятельной работы 7](#__RefHeading___Toc245_4251648076)

[3 Выводы 7](#__RefHeading___Toc247_4251648076)

# 1 Цель работы

Приобрести навыки написания программ с использованием циклов и обработкой аргументов командной строки.

# 2 Выполнение лабораторной работы

## 2.1 Реализация циклов в NASM

Создадим каталог для программ лабораторной работы №9, перейдём в него и создадим файл lab9-1.asm.

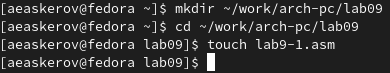


Рис. 1: Создание каталога lab09 и файла lab9-1.asm

При реализации циклов в NASM с использованием инструкции loop необходимо помнить о том, что эта инструкция использует регистр ecx в качестве счётчика и на каждом шаге уменьшает его значение на единицу. В качестве примера рассмотрим программу, которая выводит значение регистра ecx. Внимательно изучим текст программы (листинг 9.1).

Введём в файл lab9-1.asm текст программы из листинга 9.1. Создадим исполняемый файл и проверим его работу.

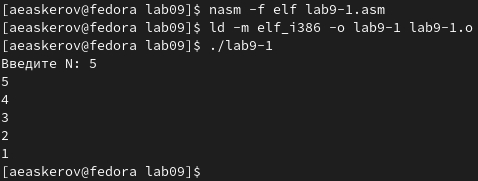


Рис. 2: Проверка работы программы lab9-1.asm

Данный пример показывает, что использование регистра ecx в теле цикла loop может привести к некорректной работе программы. Изменим текст программы, добавив изменение значения регистра ecx в цикле.

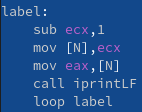


Рис. 3: Изменённый текст программы lab9-1.asm в цикле

Создадим исполняемый файл и проверим его работу.

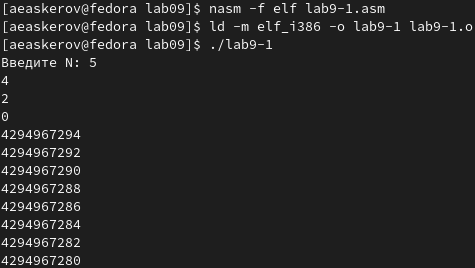


Рис. 4: Проверка работы программы lab9-1.asm

Значения ecx: 4, 2, 0, 4294967294, … Значения уменьшаются на два. Число проходов цикла не соответствует значению N введённому с клавиатуры.

Для использования регистра ecx в цикле и сохранения корректности работы программы можно использовать стек. Внесём изменения в текст программы, добавив команды push и pop (добавление в стек и извлечение из стека) для сохранения значения счётчика цикла loop.

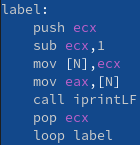


Рис. 5: Изменённый текст программы lab9-1.asm в цикле

Создадим исполняемый файл и проверим его работу.

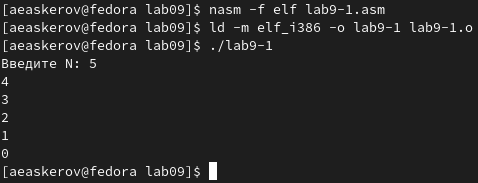


Рис. 6: Проверка работы программы lab9-1.asm

В данном случае число проходов цикла соответствует значению N введённому с клавиатуры.

## 2.2 Обработка аргументов командной строки

При разработке программ иногда встаёт 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. В качестве примера рассмотрим программу, которая выводит на экран аргументы командной строки. Внимательно изучим текст программы (листинг 9.2).

Создадим файл lab9-2.asm в каталоге ~/work/arch-pc/lab09 и введём в него текст программы из листинга 9.2.

Рис. 7: Создание файла lab9-2.asm

Рис. 7: Создание файла lab9-2.asm

Создадим исполняемый файл и запустим его, указав аргументы 1, 2, 3.

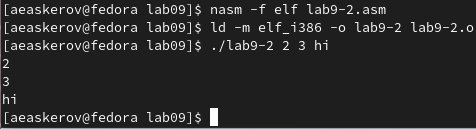


Рис. 8: Проверка работы программы lab9-2.asm

Программой были обработаны 3 аргумента.

Рассмотрим ещё один пример программы, которая выводит сумму чисел, которые передаются в программу как аргументы. Создадим файл lab9-3.asm в каталоге ~/work/arch-pc/lab09 и введём в него текст программы из листинга 9.3.

Создадим исполняемый файл и запустим его, указав аргументы.

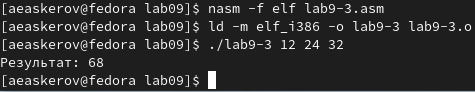


Рис. 9: Проверка работы программы lab9-3.asm

Изменим текст программы из листинга 9.3 для вычисления произведения аргументов командной строки.

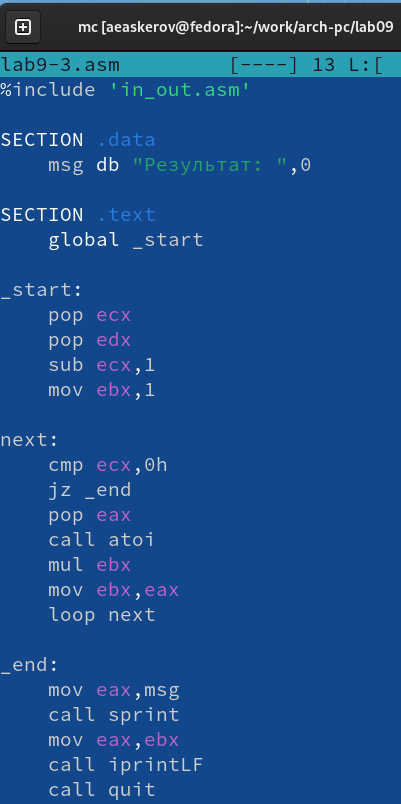


Рис. 10: Изменённый текст программы lab9-3.asm в цикле

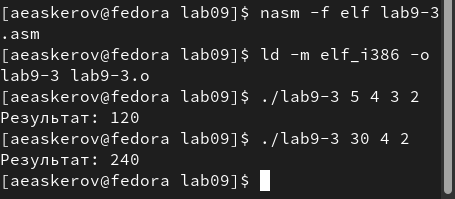


Рис. 11: Проверка работы программы lab9-3.asm

## 2.3 Задание для самостоятельной работы

Напишем программу, которая находит сумму значений функции f(x) для x = x1, x2, …, xn, т.е. программа должна выводить значение f(x1) + f(x2)+…+f(xn). Значения xi передаются как аргументы. Вид функции f(x) выберем из таблицы 9.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы №7. Создадим исполняемый файл и проверим его работу на нескольких наборах x = x1, x2, …, xn.

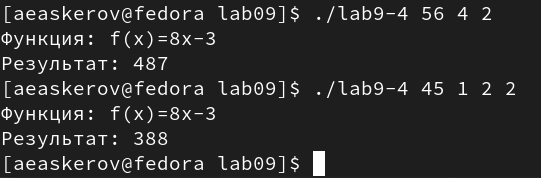


Рис. 12: Пример работы задания для самостоятельной работы (вариант 19)

# 3 Выводы

Приобретены навыки написания программ с использованием циклов и обработкой аргументов командной строки.