Лабораторная работа №8

Программирование цикла. Обработка аргументов командной строки.

Седохин Даниил Алексеевич

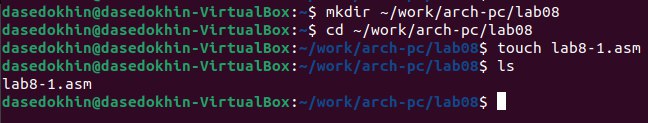
Содержание

# 1 Цель работы

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

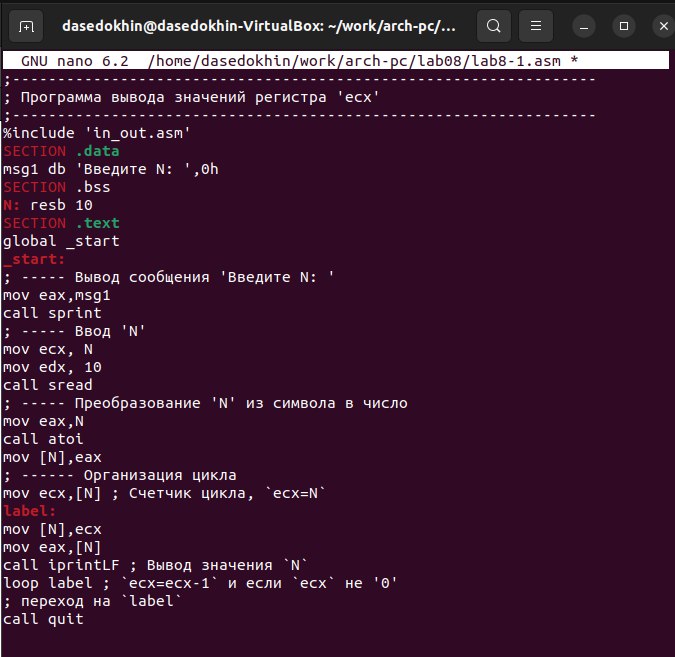
# 2 Выполнение лабораторной работы

1. Создадим каталог для программам лабораторной работы № 8, перейдём в него и создадим файл lab8-1.asm: (рис. ??).



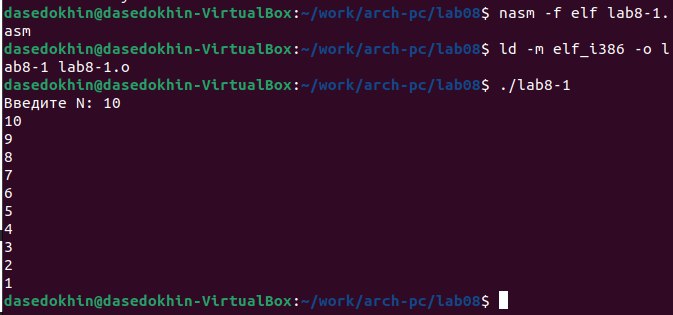
Создание каталога для лабораторной работы №8 и файла lab8-1.asm

1. 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. Введём в файл lab8-1.asm текст программы из листинга 8.1. (рис. ??).



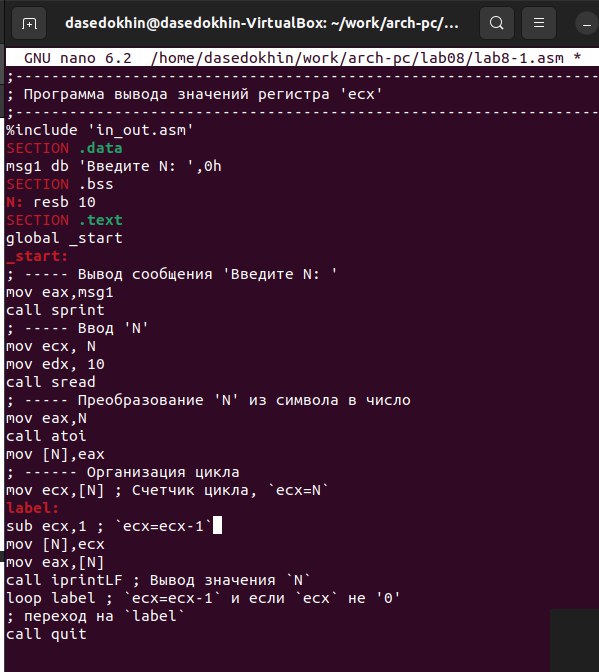
Ввод в файл lab8-1.asm текст программы из листинга 8.1

Создадим исполняемый файл и проверим его работу. (рис. ??).



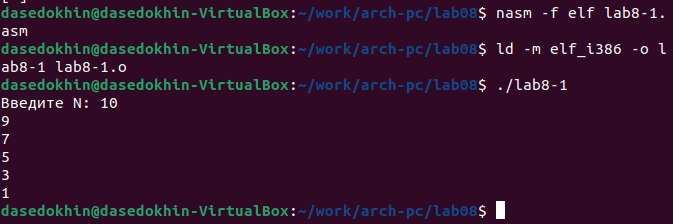
Создание и проверка исполняемого файла lab8-1.asm

Данный пример показывает, что использование регистра ecx в теле цилка loop может привести к некорректной работе программы.  
Изменим текст программы добавив изменение значение регистра ecx в цикле: (рис. ??).



Изменение текста программы, изменив значение регистра ecx

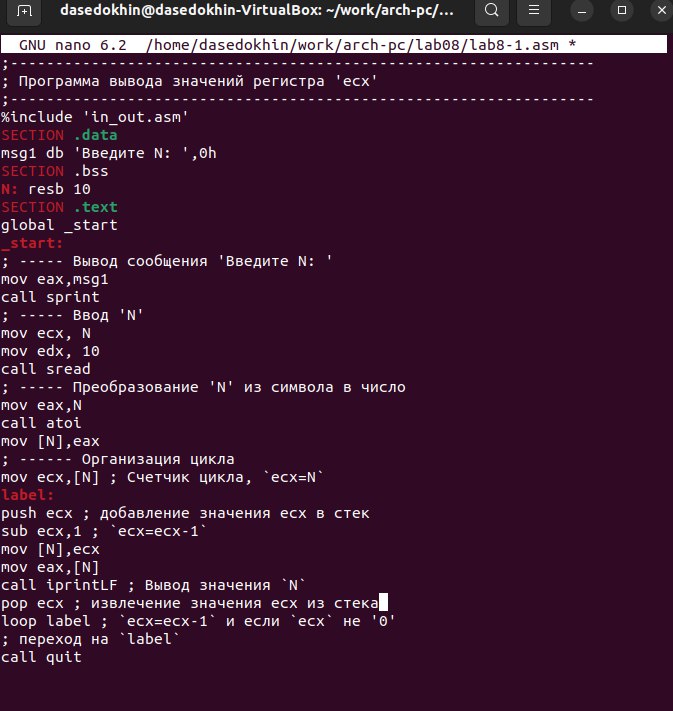
Создадим исполняемый файл и проверим его работу. (рис. ??).



Создание и проверка исполняемого файла

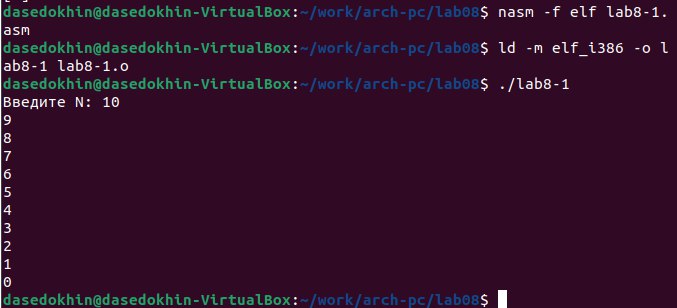
Число проходов цикла не соответствуют введённому значению с клавиатуры. Программа обрабатывает только нечетные числа.

Для использования регистра ecx в цикле и сохранения корректности работы программы можно использовать стек.  
Внесём изменения в текст программы добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop (рис. ??).



Изменение текста программы, добавив команды push и pop

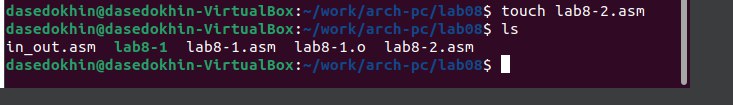
Создадим исполняемый файл и проверим его. (рис. ??).



Создание и проверка исполняемого файла

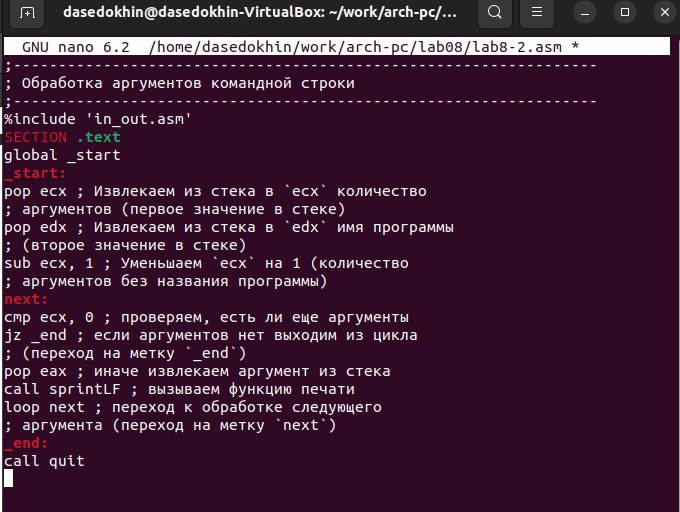
В данном случае число переходов соответствует числу введённому с клавиатуры.

При разработке программ иногда встает 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 обрат- ном 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 из- влечь 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.  
Создадим файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 (рис. ??).



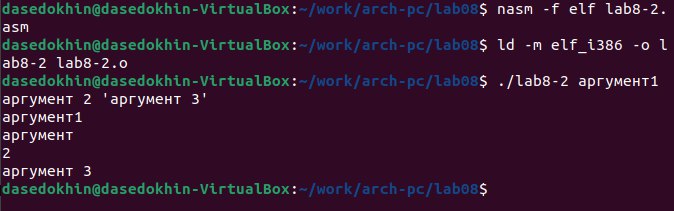
Создание файла lab8-2.asm

Введём в него текст программы из листинга 8.2. (рис. ??).



Ввод текст программы в файл lab7-2.asm из листинга 8.2.

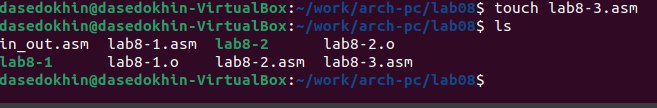
Создание исполняемого файла и его запуск, с указанием аргументов. (рис. ??).



Создание исполняемого файла и его проверка с указанием аргументов

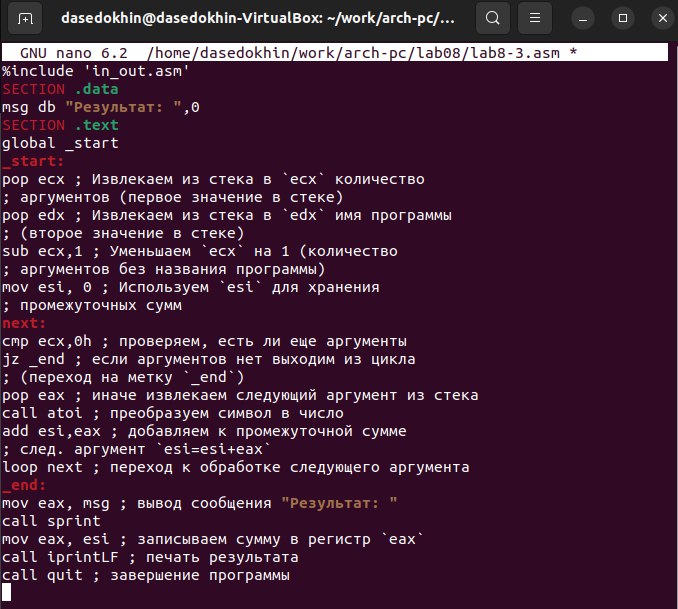
Программой было обработаны все 3 аргумента

Рассмотрим еще один пример программы которая выводит сумму чисел, которые пере- даются в программу как аргументы. Создадим файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 (рис. ??).



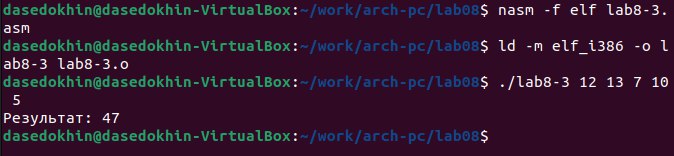
Создание файла lab8-3.asm в каталоге ~/work/arch-pc/lab08

Введём в него текст программы из листинга 8.3. (рис. ??).



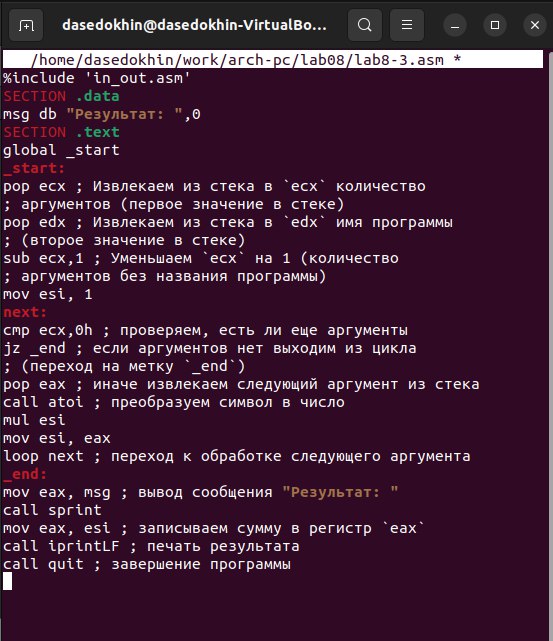
Ввод текста программы в файл lab8-3.asm из листинга 8.3.

Создадим исполняемый файл и запустим его, указав аргументы. (рис. ??).



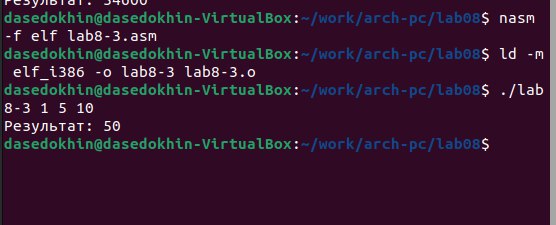
Создание и проверка исполняемого файла lab8-3.asm

Изменим текст программы из листинга 8.3 для вычисления произведения аргументов командной строки. (рис. ??).



Изменение текста программы lab8-3.asm для вычисления произведения аргументов командной строки

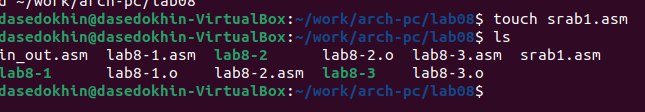
Создадим исполняемый файл и проверим его, указав аргументы. (рис. ??).



Создание и проверка исполняемого изменённого файла lab8-3.asm

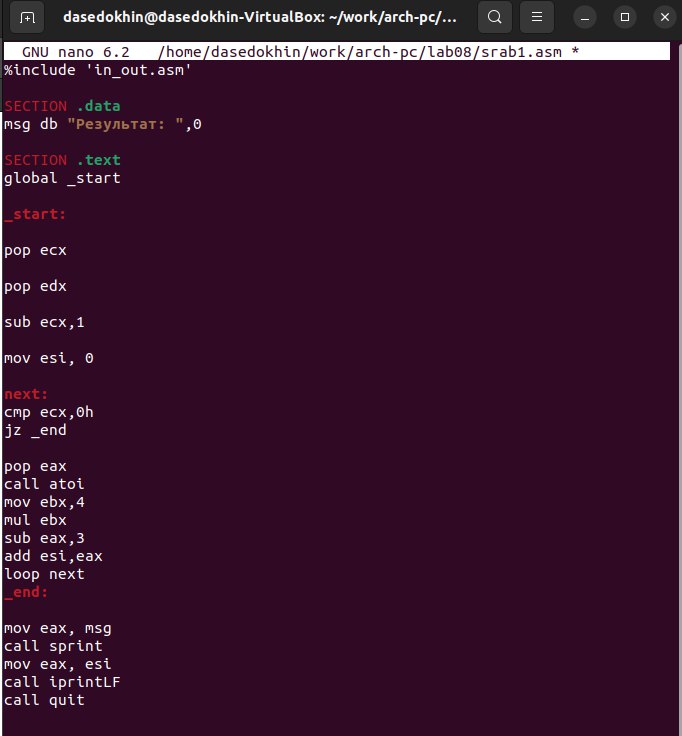
# 3 Задание для самостоятельной работы

Создадим файл srab1.asm в каталоге ~/work/arch-pc/lab08 (рис. ??).



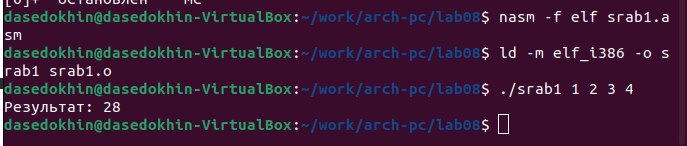
Создание файла srab1.asm

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) возьмём из таблицы 8.1 вариантов заданий в соответствии с 6 вариантом. Моя функция будет следующей: f(x) = 4x - 3 (рис. ??).



Пишем код программы для 6-го варианта

Создадим исполняемый файл и проверим его работу на нескольких наборах x = x1, x2, …, xn (рис. ??).



Создание и проверка исполняемого файла

# 4 Выводы

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