Лабораторная работа №8. Программирование цикла. Обработка аргументов командной строки

Простейший вариант

Диана Садова Алексеевна

Содержание

# 1 Цель работы

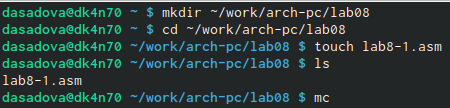
Приобретение навыков написания программ с использованием циклов и обработкой аргументов командной строки.

# 2 Задание

## 2.1 Порядок выполнения лабораторной работы

### 2.1.1 Реализация циклов в NASM

Создайте каталог для программам лабораторной работы № 8, перейдите в него и создайте файл lab8-1.asm:(рис. ??).

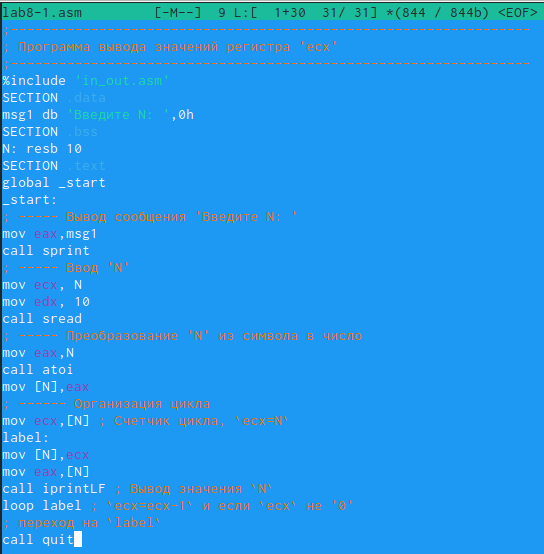


Создаем каталог и проверяем его наличие

Проверили, что файл создался правильно. Приступаем к следующему пункту лабораторной работы

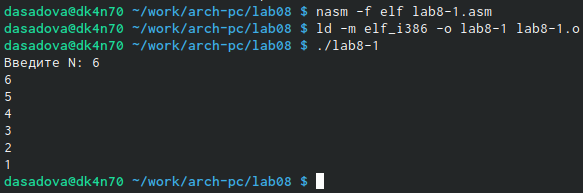
При реализации циклов в NASM с использованием инструкции loop необходимо помнить о том, что эта инструкция использует регистр ecx в качестве счетчика и на каждом шаге уменьшает его значение на единицу. В качестве примера рассмотрим программу, которая выводит значение регистра ecx. Внимательно изучите текст программы (Листинг 8.1),(рис. ??)

Листинг 8.1. Программа вывода значений регистра ecx.



Вводим код

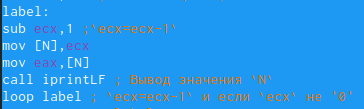
Введите в файл lab8-1.asm текст программы из листинга 8.1. Создайте исполняемый файл и проверьте его работу.(рис. ??).



Создаем исполняемый файл

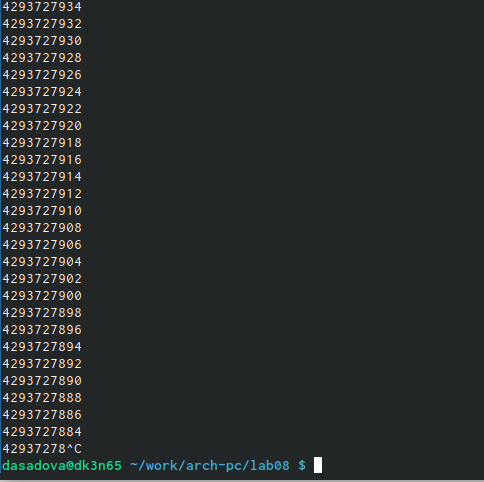
Видим, что программы работает корректно

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

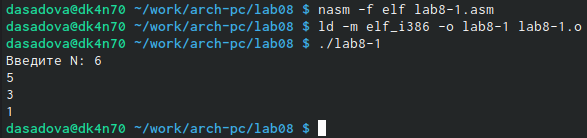


Изменяем текст программы

Создайте исполняемый файл и проверьте его работу. Какие значения принимает регистр ecx в цикле? Соответствует ли число проходов цикла значению N введенному с клавиатуры?(рис. ??),(рис. ??).



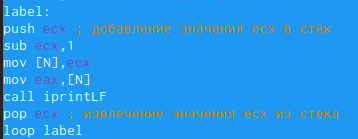
Вывод если число не четное



Вывод если число четное

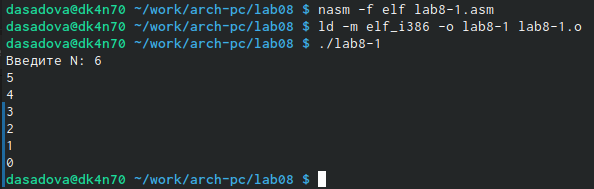
Если мы вводим не четное значение, то программа выводит не корректное значение. Если мы вводим четное число, программа выводит верные значения. В том и другом случае число N не соответствует числу проходов.

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



Вносим изменения в текс программы

Создайте исполняемый файл и проверьте его работу. Соответствует ли в данном случае число проходов цикла значению 𝑁 введенному с клавиатуры?(рис. ??).



Создаем исполняемый файл и проверяем его работу

Проверяем, что программа выполняется корректно

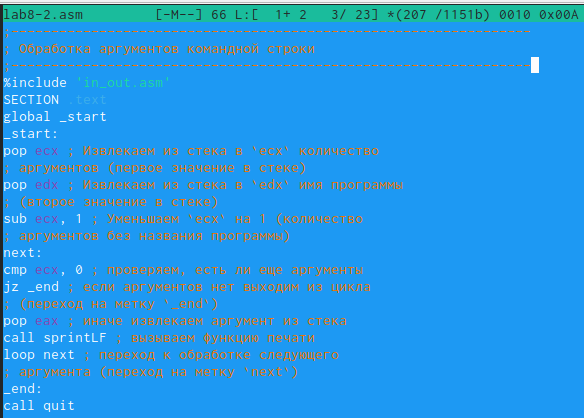
### 2.1.2 Обработка аргументов командной строки

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 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.

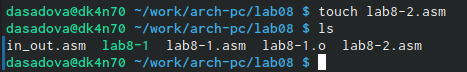
Таким образом, для того чтобы использовать аргументы в программе, их просто нужно извлечь из стека. Обработку аргументов нужно проводить в цикле. Т.е. сначала нужно извлечь из стека количество аргументов, а затем циклично для каждого аргумента выполнить логику программы. В качестве примера рассмотрим программу, которая выводит на экран аргументы командной строки. Внимательно изучите текст программы (Листинг 8.2).

Листинг 8.2. Программа выводящая на экран аргументы командной строки(рис. ??)



Вводим код

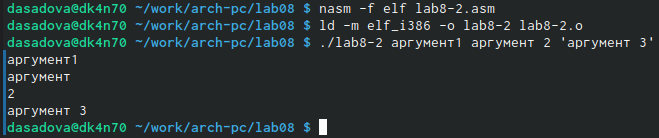
Создайте файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и введите в него текст программы из листинга 8.2.(рис. ??).



Создаем каталог и проверяем его наличие

Проверяем наличия файла в каталоге можем продолжать лабораторную работу

Создайте исполняемый файл и запустите его, указав аргументы:(рис. ??).

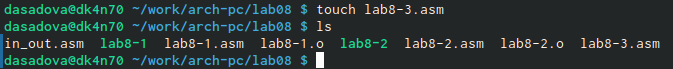


Создаем исполняемый файл и запускаем его с аргументами

Сколько аргументов было обработано программой?

Программой было выведено 4 аргумента: 1 строка - значение “аргумента1” был вписано без пробела, вывилось на экран как один аргумент, строка 2 и 3 - значение “аргумент 2” было вписано с пробелом, вывелось как два аргумента, 4 строка - значение “‘агрумент3’” изначально вводилось в скобках и программой распознавался как один аргумент.

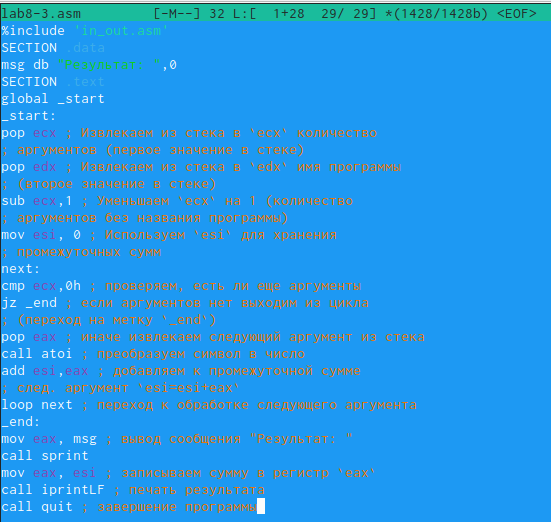
Рассмотрим еще один пример программы которая выводит сумму чисел, которые пере даются в программу как аргументы. Создайте файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 и введите в него текст программы из листинга 8.3.(рис. ??).



Создаем каталог и проверяем его наличие

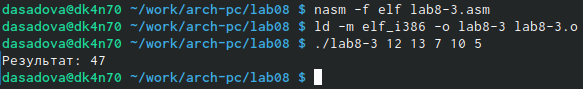
После проверки наличия файла в каталоге можем продолжать лабораторную работу

Листинг 8.3. Программа вычисления суммы аргументов командной строки(рис. ??).



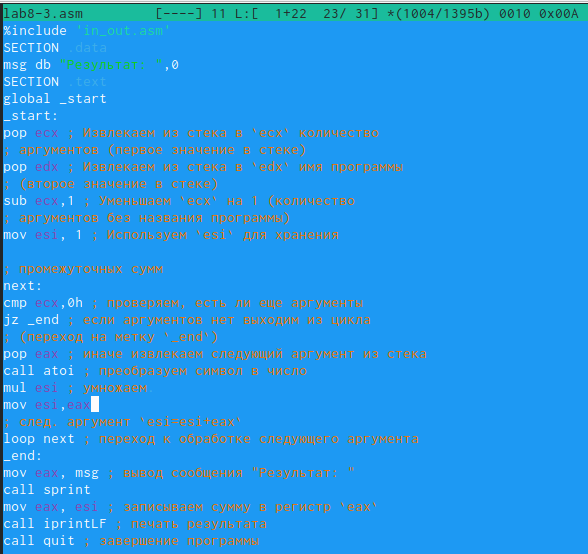
Вводим код

Создайте исполняемый файл и запустите его, указав аргументы. Пример результата работы программы:(рис. ??).

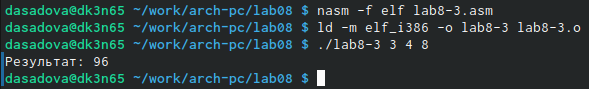


Вводим код

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



Изменяем текст программы



Создаем исполняемый файл и запускаем

Видим, что программа выполняется корректно

# 3 Теоретическое введение

# 4 Выполнение лабораторной работы

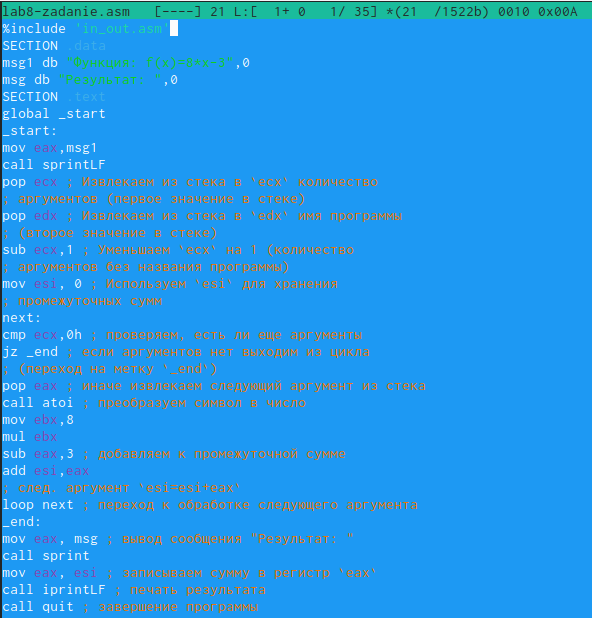
## 4.1 Напишите программу, которая находит сумму значений функции f(x) для

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) 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 x = x1, x2, …, xn. Пример работы программы для функции f(x) = x + 2 и набора x1 = 1, x2 = 2, x3 = 3, x4 = 4:

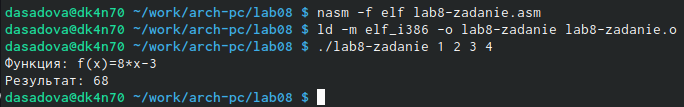
user@dk4n31:~$ ./main 1 2 3 4 Функция: f(x)=x+2 Результат: 18 user@dk4n31:~$

(рис. ??).(рис. ??).

Мой вариант 19. При решении этого задания используем полученные навыки при решении лабораторной работы и данные 19 варианта.



Создаем программу



Создаем исполняемый файл и запускаем его

# 5 Выводы

Мы приобрели навыки написания программ с использованием циклов и обработки аргументов командной строки.

# Список литературы