Отчёта по лабораторной работе 9

Программирование цикла. Обработка аргументов командной строки.

Дев Авинаш

НКАбд-05-22

Содержание

[1 Цель работы 1](#_Toc128529537)

[2 Задание 1](#_Toc128529538)

[3 Теоретическое введение 1](#_Toc128529539)

[4 Выполнение лабораторной работы 2](#_Toc128529540)

[5 Выводы 10](#_Toc128529541)

[Список литературы 11](#_Toc128529542)

# 1 Цель работы

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

# 2 Задание

1. Изучите примеры программ
2. 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 x передаются как аргументы. Вид функции f(x) выбрать из таблицы 9.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах x.
3. Загрузите файлы на GitHub.

# 3 Теоретическое введение

Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией стека является функция сохранения адресов возврата и передачи аргументов при вызове процедур. Кроме того, в нём выделяется память для локальных переменных и могут временно храниться значения регистров.

Для организации циклов существуют специальные инструкции. Для всех инструкций максимальное количество проходов задаётся в регистре ecx. Наиболее простой является инструкция loop. Иструкция loop выполняется в два этапа. Сначала из регистра ecx вычитается единица и его значение сравнивается с нулём. Если регистр не равен нулю, то выполняется переход к указанной метке. Иначе переход не выполняется и управление передаётся команде, которая следует сразу после команды loop.

# 4 Выполнение лабораторной работы

1. Создайте каталог для программам лабораторной работы № 9, перейдите в него и создайте файл lab9-1.asm
2. Введите в файл lab9-1.asm текст программы из листинга 9.1. Создайте исполняемый файл и проверьте его работу. (рис. 1, 2)

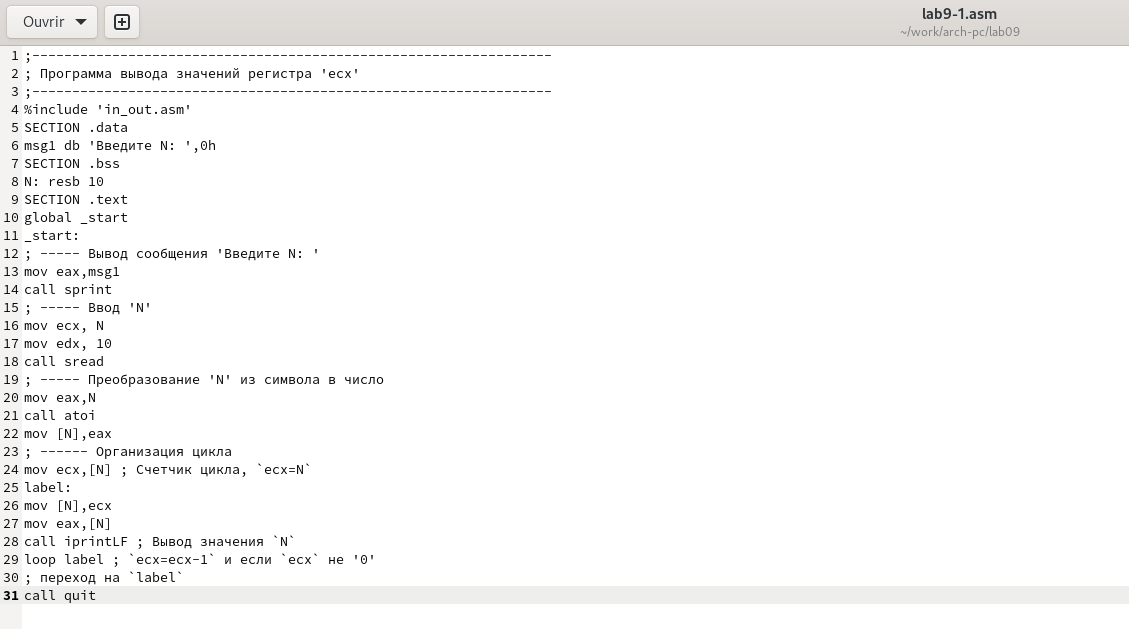


Рис. 1: Файл lab9-1.asm

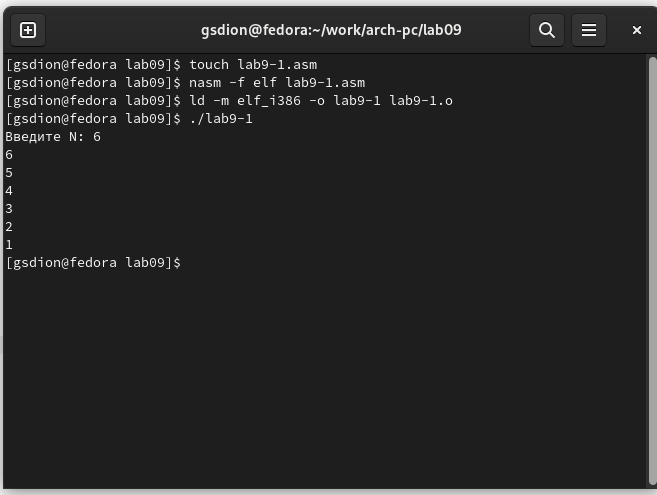


Рис. 2: Работа программы lab9-1.asm

1. Данный пример показывает, что использование регистра ecx в теле цилка loop может привести к некорректной работе программы. Измените текст программы добавив изменение значение регистра ecx в цикле: 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. 3, 4)

Программа запускает бесконечный цикл при нечетном N и выводит только нечетные числа при четном N.

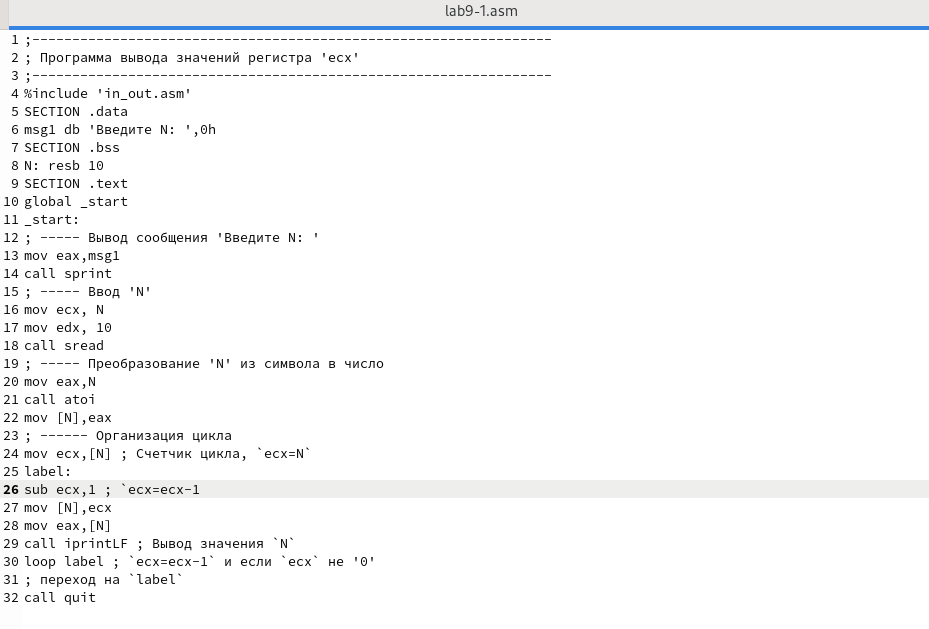


Рис. 3: Файл lab9-1.asm



Рис. 4: Работа программы lab9-1.asm

1. 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 loop. Создайте исполняемый файл и проверьте его работу. Соответствует ли в данном случае число проходов цикла значению N введенному с клавиатуры? (рис. 5, 6)

Программа выводит числа от N-1 до 0, число проходов цикла соответсвует N.

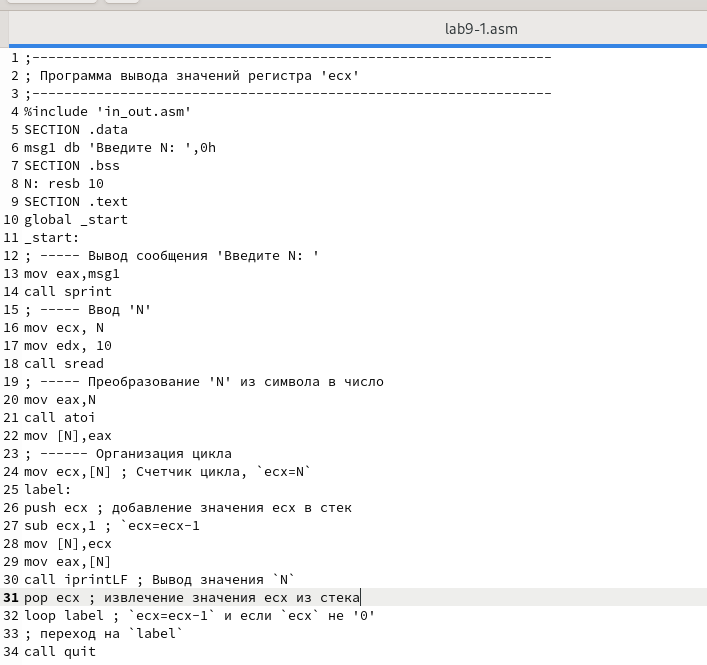


Рис. 5: Файл lab9-1.asm

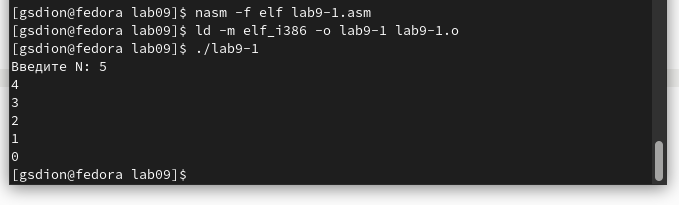


Рис. 6: Работа программы lab9-1.asm

1. Создайте файл lab9-2.asm в каталоге ~/work/arch-pc/lab09 и введите в него текст программы из листинга 9.2. Создайте исполняемый файл и запустите его, указав аргументы. (рис. 7, 8) Сколько аргументов было обработано программой?

Программа обработала 5 аргументов.

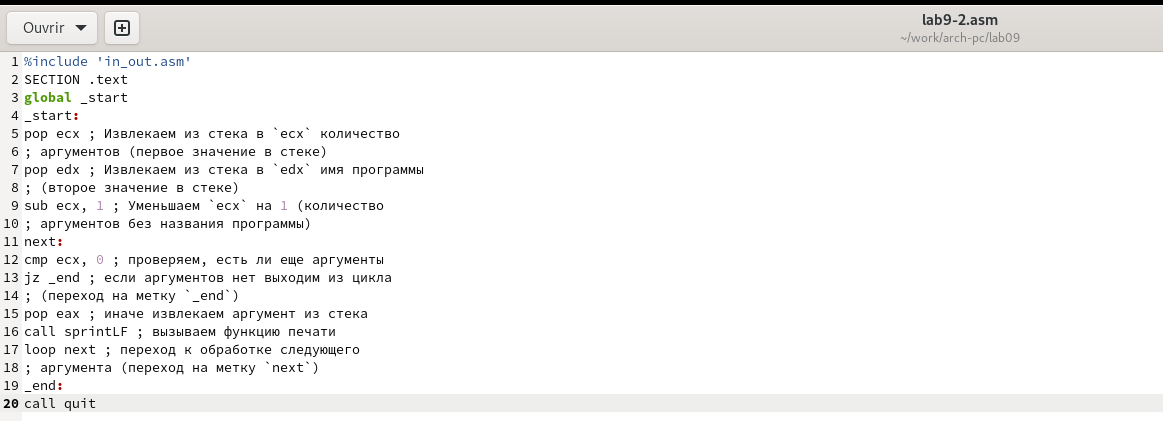


Рис. 7: Файл lab9-2.asm

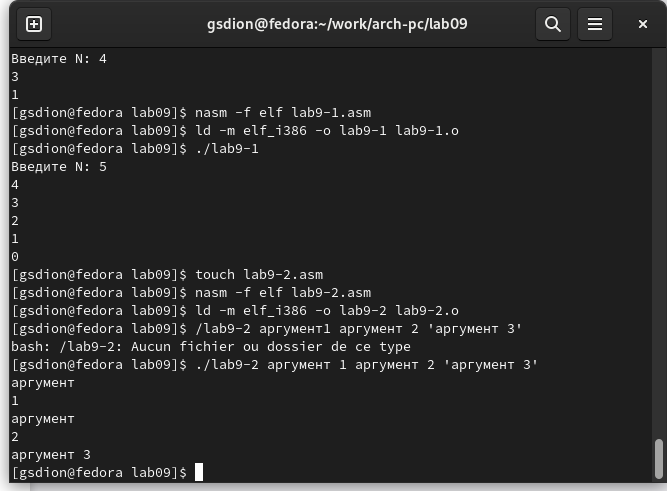


Рис. 8: Работа программы lab9-2.asm

1. Рассмотрим еще один пример программы которая выводит сумму чисел, которые передаются в программу как аргументы. (рис. 9, 10)

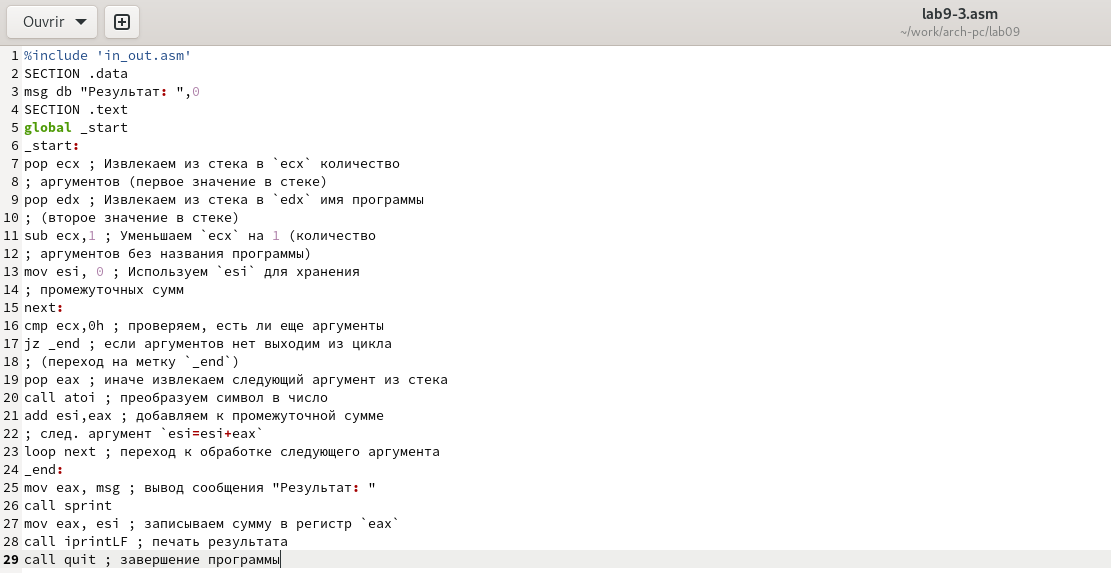


Рис. 9: Файл lab9-3.asm

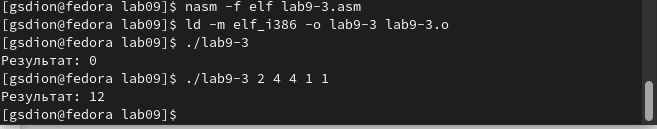


Рис. 10: Работа программы lab9-3.asm

1. Измените текст программы из листинга 9.3 для вычисления произведения аргументов командной строки. (рис. 11, 12)

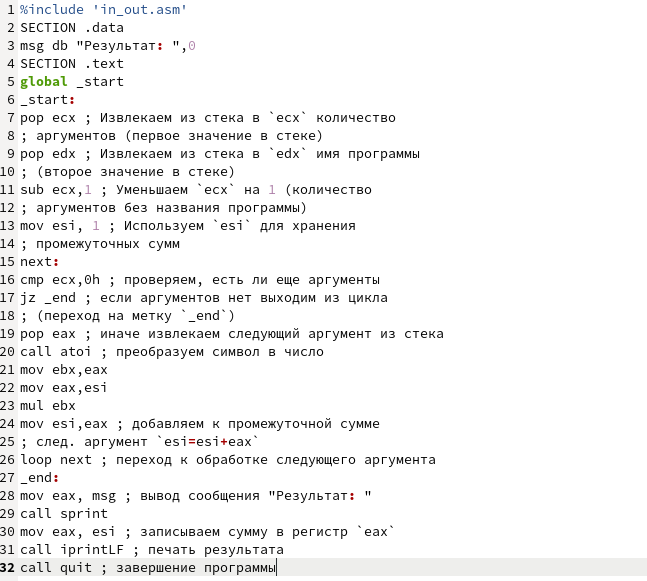


Рис. 11: Файл lab9-3.asm

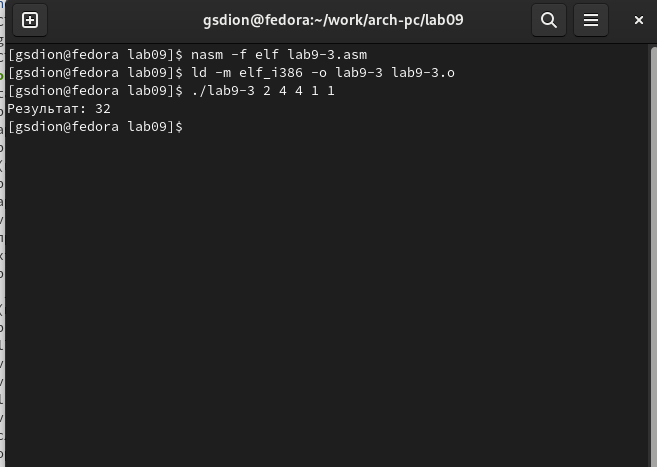


Рис. 12: Работа программы lab9-3.asm

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 x передаются как аргументы. Вид функции f(x) выбрать из таблицы 9.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу на нескольких наборах x. (рис. 13, 14)

для варивнта 10 f(x) = 5(2+x)

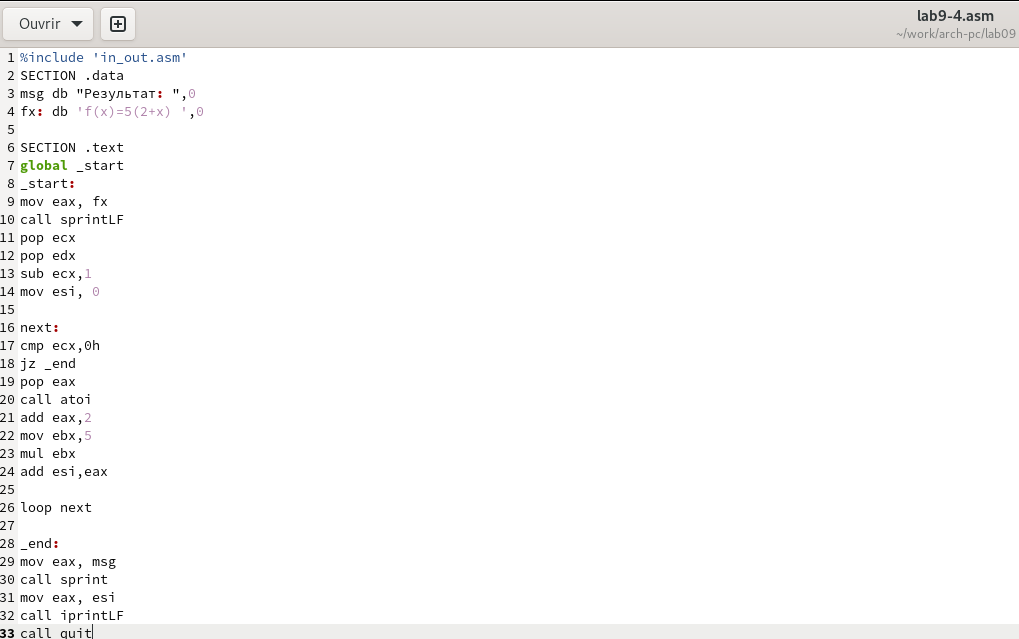


Рис. 13: Файл lab9-4.asm

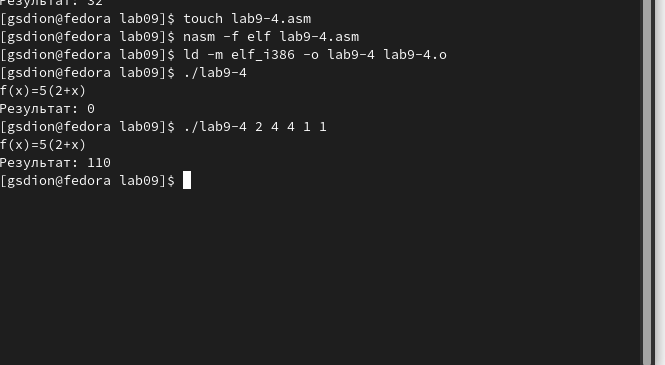


Рис. 14: Работа программы lab9-4.asm

# 5 Выводы

В заключение, это позволило нам освоить работу со стеком, циклом и аргументами в ассемблере Nasm.

# Список литературы

1. [Расширенный ассемблер: NASM](https://www.opennet.ru/docs/RUS/nasm/)