Лабораторная работа №8.

Программирование цикла.Обработка аргументов командной строки.

Швед Карина Дмитриевна

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

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

# 2 Ход 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.

Создала каталог для программ лабораторной работы № 8, перешла в него и создала файл lab8-1.asm. Далее я внимательно изучила текст программы из листинга 8.1 и ввела в файл lab8-1.asm. (рис. 1). Создала исполняемый файл и проверила его работу.(рис. 2).

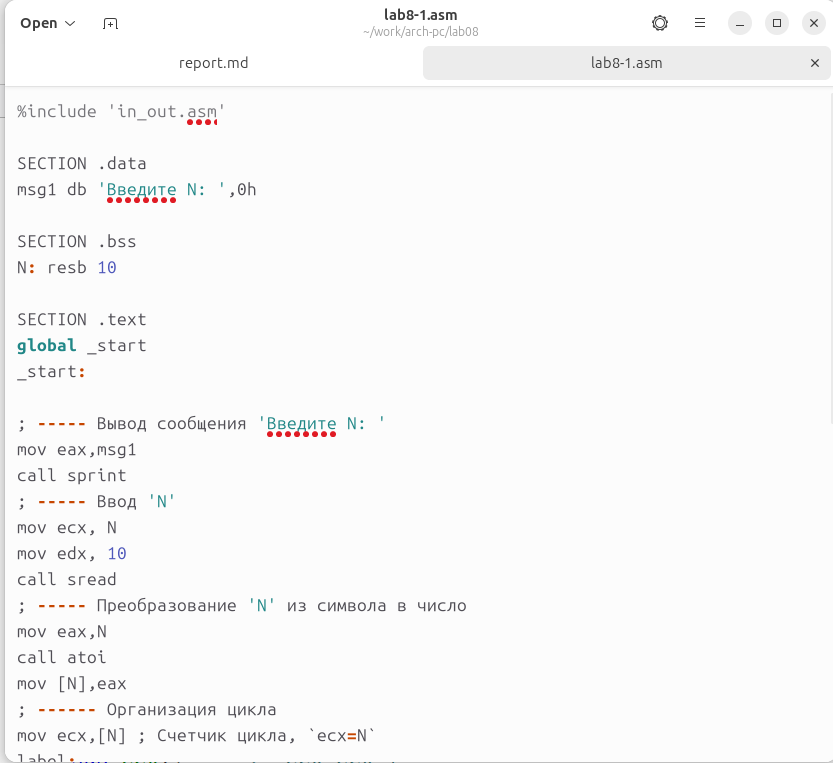


Рис. 1: Код программы в файле lab8-1.asm

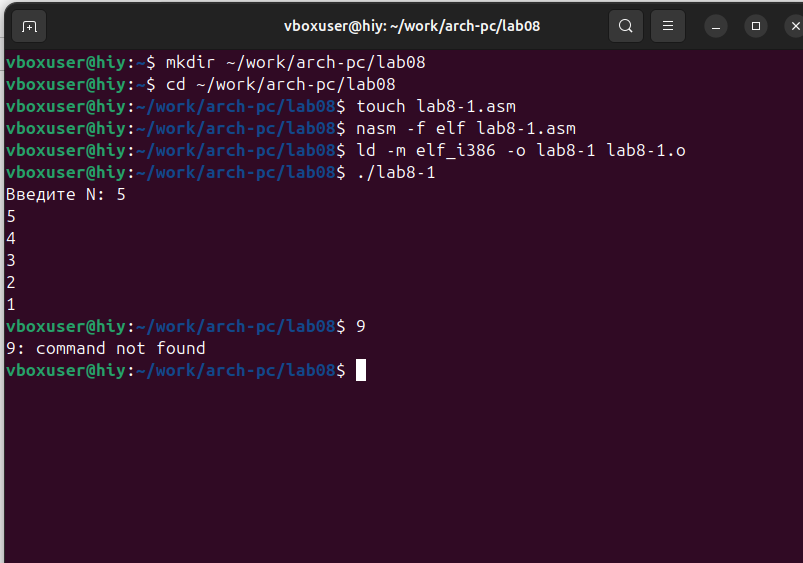


Рис. 2: Работа программы lab8-1.asm

Данный пример показывает, что использование регистра ecx в теле цилка loop может привести к некорректной работе программы. Я изменила текст программ, добавив изменение значение регистра ecx в цикле (рис. 3). Создала исполняемый файл и проверила его работу.

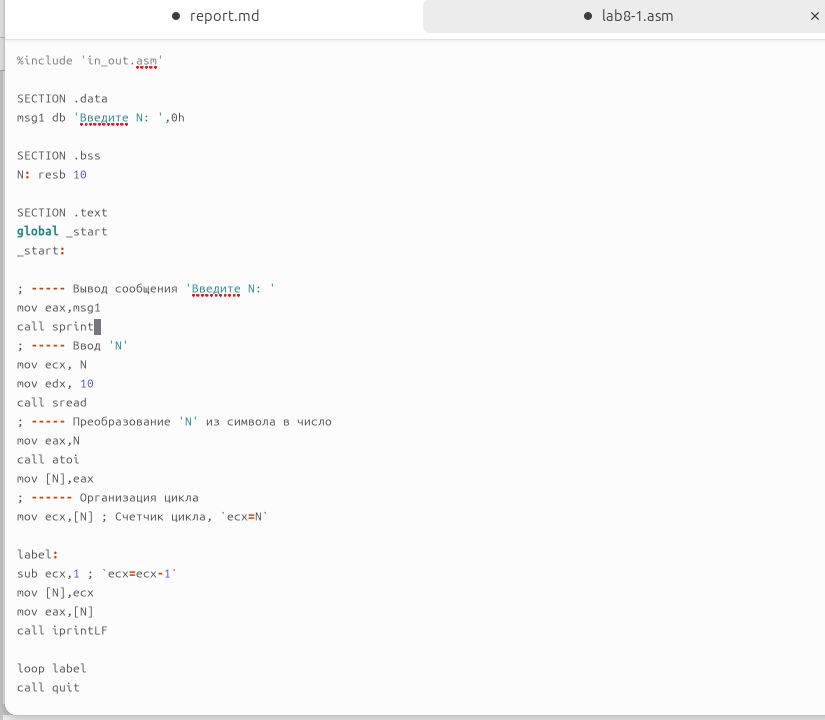


Рис. 3: Код программы в файле lab8-1.asm

При нечетном N программа запускает бесконечный цикл (рис. 4), а при четном N выводит только нечетные числа (рис. 5)

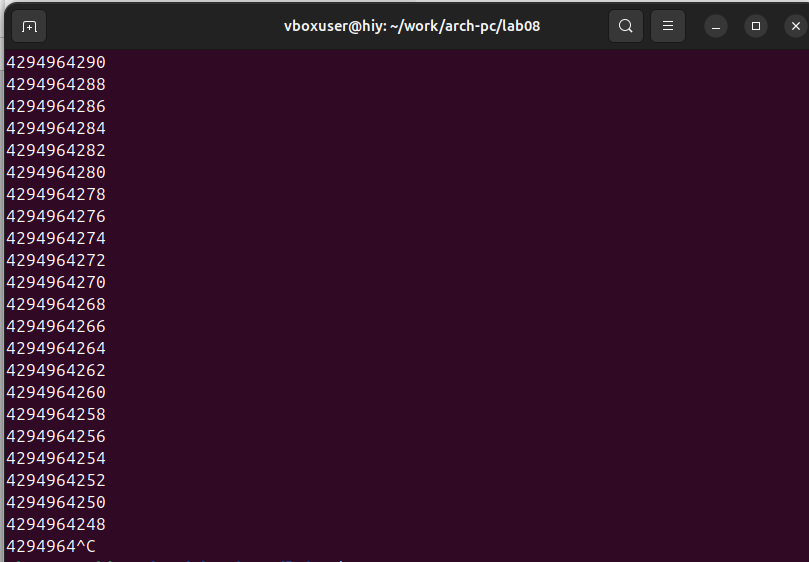


Рис. 4: Запуск программы lab8-1.asm

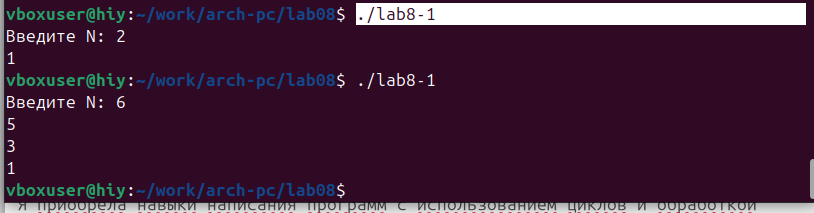


Рис. 5: Запуск программы lab8-1.asm

Для использования регистра ecx в цикле и сохранения корректности работы программы можно использовать стек. Я добавила в программу команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop (рис. 6) Создала исполняемый файл и проверила его работу (рис. 7) Программа выводит числа от N-1 до 0, число проходов цикла соответствует N.

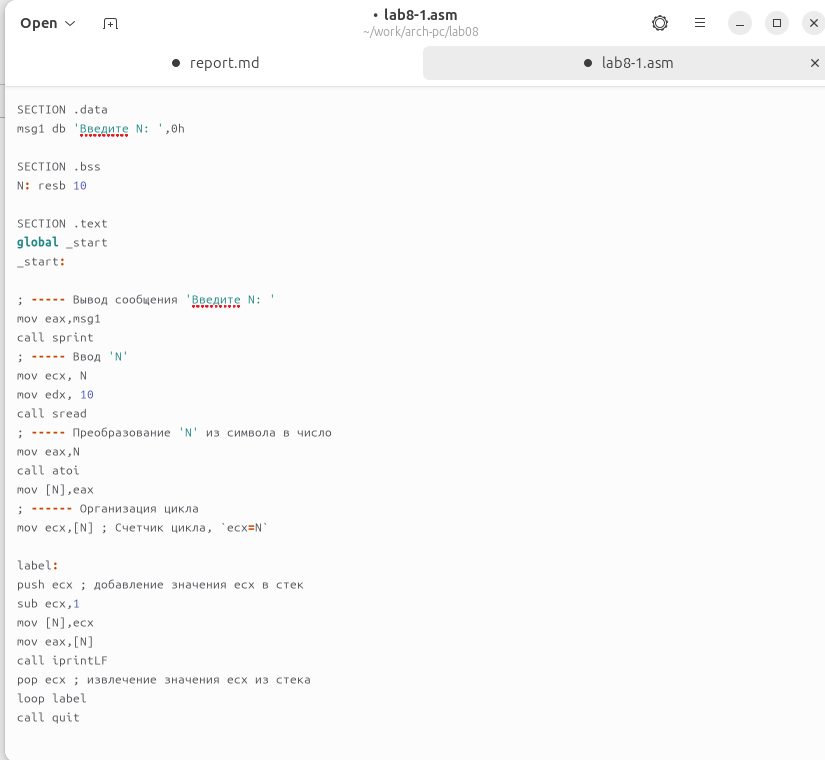


Рис. 6: Код программы lab8-1.asm

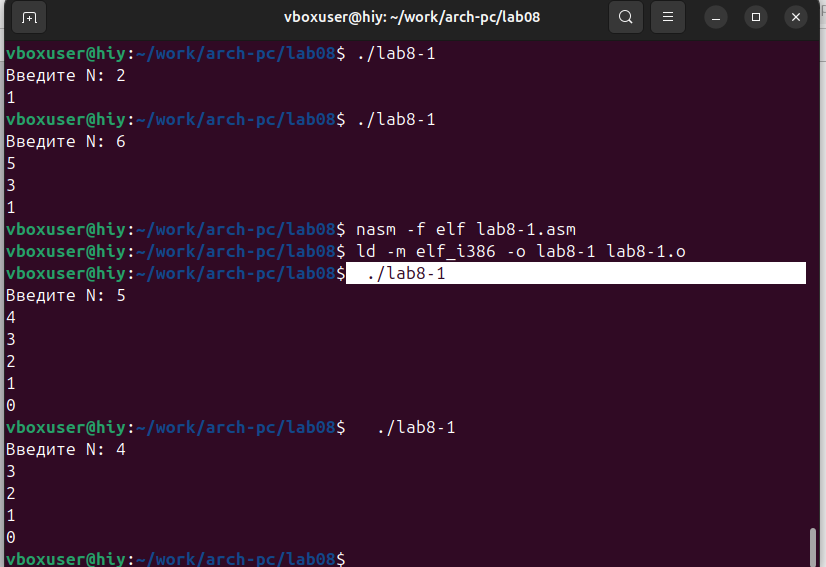


Рис. 7: Работа программы lab8-1.asm

При разработке программ иногда встает необходимость указывать аргументы, которые будут использоваться в программе, непосредственно из командной строки при запуске программы. Для того чтобы использовать аргументы в программе, их просто нужно извлечь из стека. Обработку аргументов нужно проводить в цикле.В качестве примера рассмотрим программу, которая выводит на экран аргументы командной строки. Я внимательно изучила текст программы из Листинга 8.2.

Создала файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 и ввела в него текст программы из листинга 8.2.(рис. 8) Далее создала исполняемый файл и запустила его, указав аргументы (рис. 9). Было обработано 4 аргумента

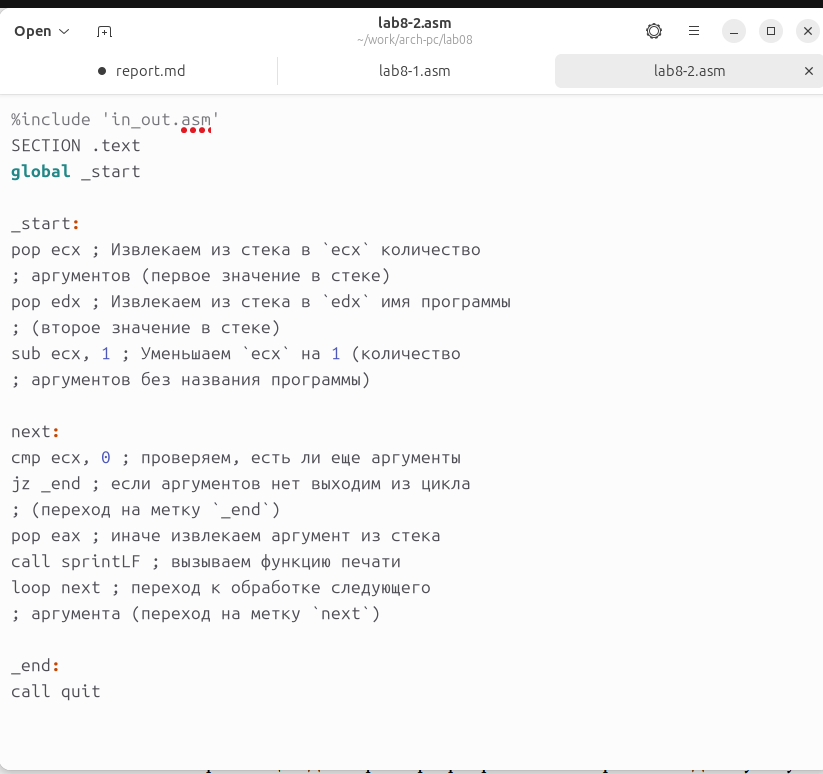


Рис. 8: Код программы lab8-2.asm

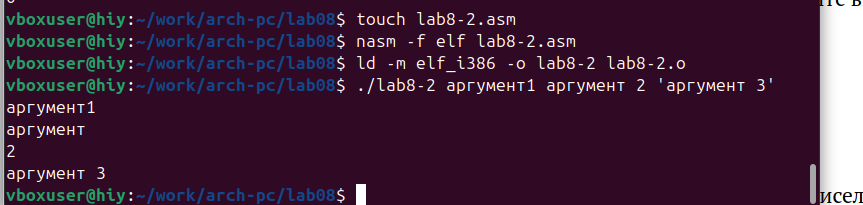


Рис. 9: Работа программы lab8-2.asm

Рассмотрим еще один пример программы которая выводит сумму чисел, которые передаются в программу как аргументы. Я создала файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 и ввела в него текст программы из листинга 8.3. (рис. 10) Создала исполняемый файл и запустите его, указав аргументы. (рис. 11)

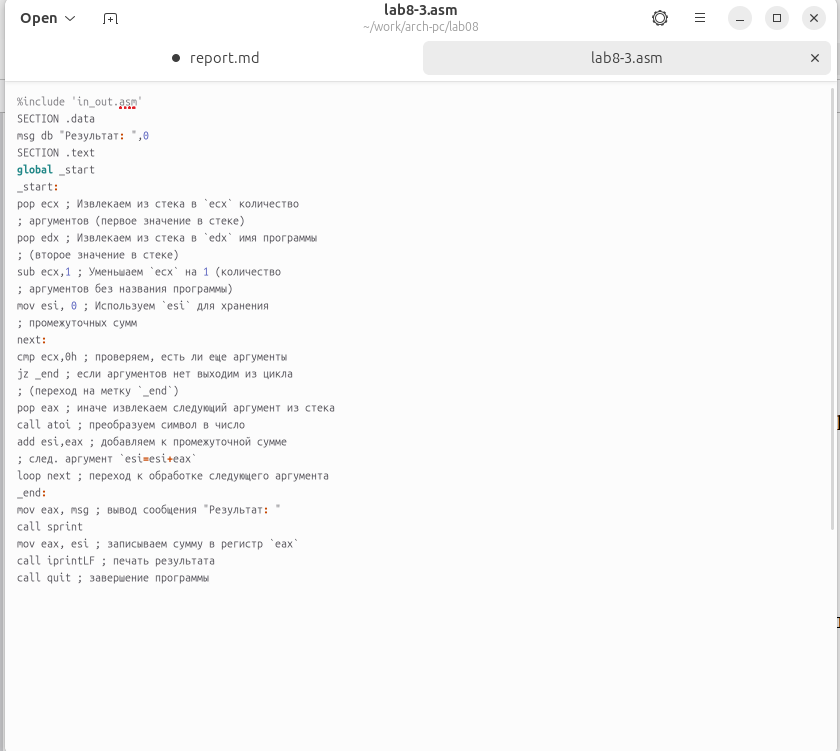


Рис. 10: Код программы lab8-3.asm

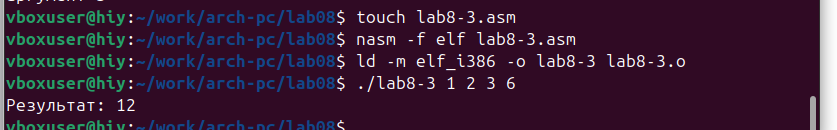


Рис. 11: Работа программы lab8-3.asm

Изменила текст программы из листинга 8.3 для вычисления произведения аргументов командной строки.(рис. 12) (рис. 13)

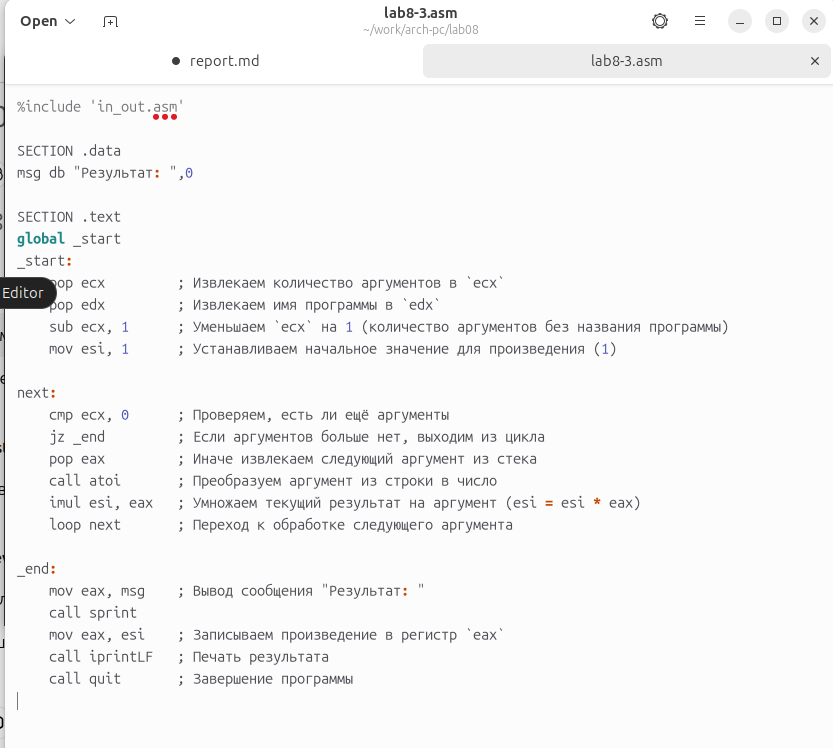


Рис. 12: Код программы lab8-3.asm

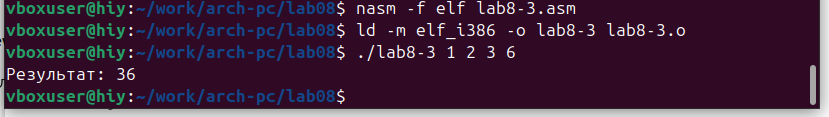


Рис. 13: Работа программы lab8-3.asm

# 3 Задание для самостоятельной работы

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.

Для варианта 5: 4𝑥 + 3

Сначала я создала файл task8-1.asm. Затем ввела соответствующий код. (рис. 14)Создала исполняемый файл и проверила работу файла. Я ввела несколько аргументов и получила сумму значений (рис. 15)

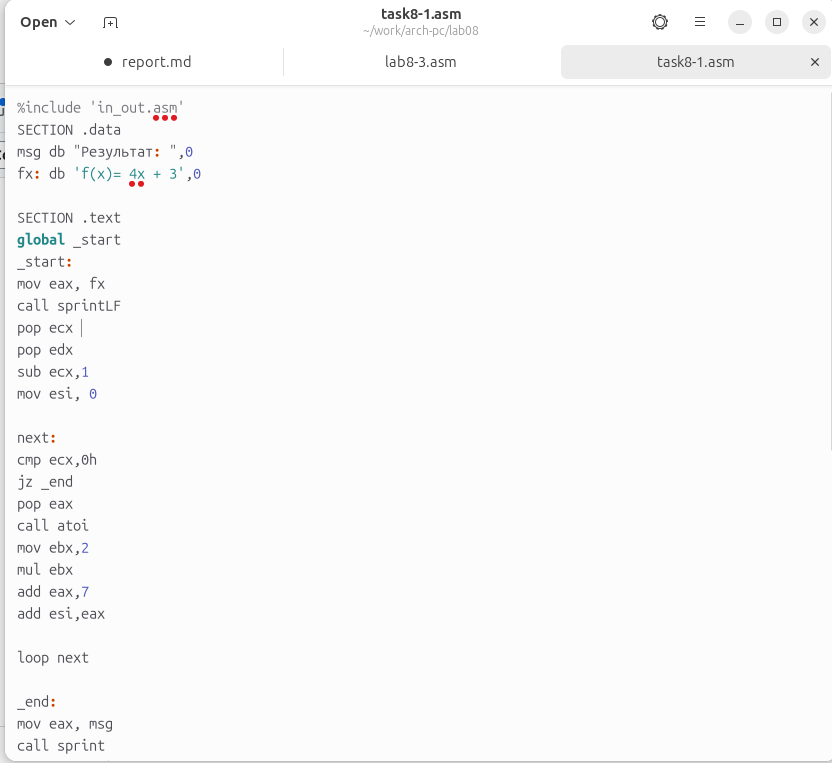


Рис. 14: Код программы task8-1.asm

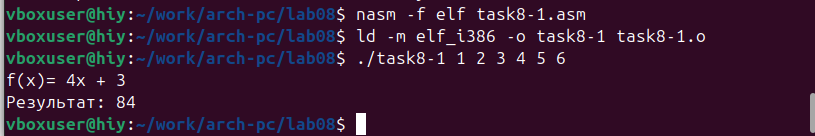


Рис. 15: Работа программы task8-1.asm

# 4 Выводы

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