Лабораторная работа №8

Программирование цикла. Обработка аргументов командной строки

Коровкин Никита Михайлович

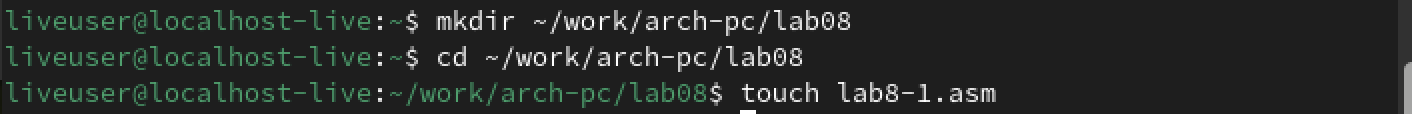
Содержание

# 1 Цель работы

Научиться работать с циклами на языке Ассемблера, а также научиться обрабатывать аргументы командной строки

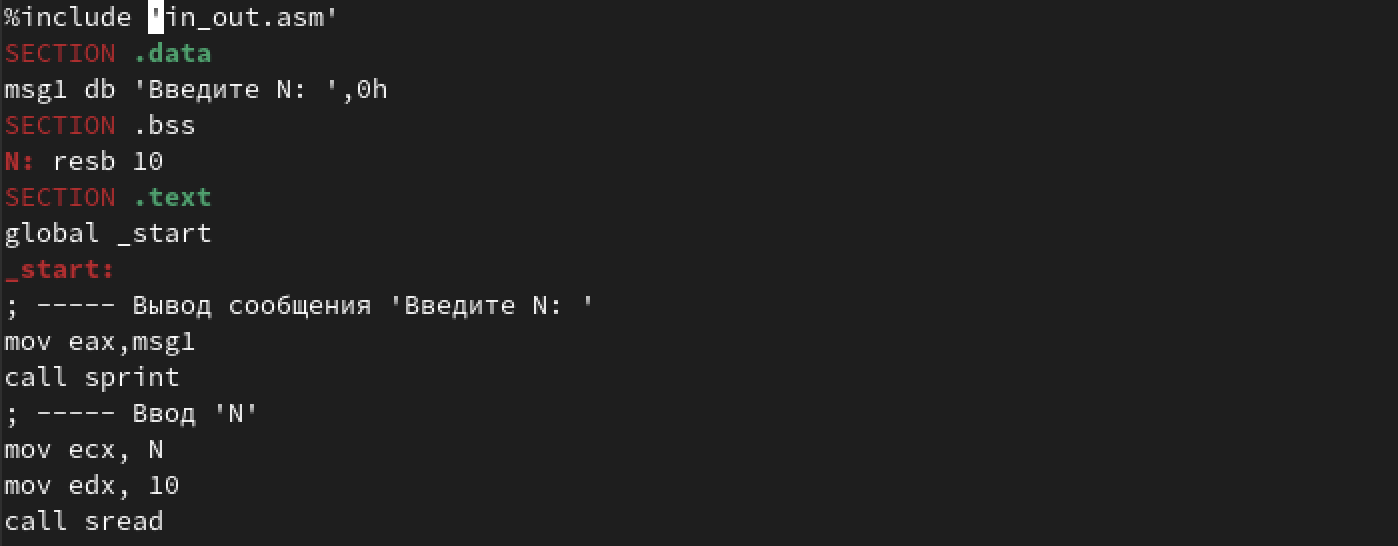
# 2 Выполнение лабораторной работы

Для начала создадим новую папку для 8-й лабораторной работы и первый файл.(рис.1)



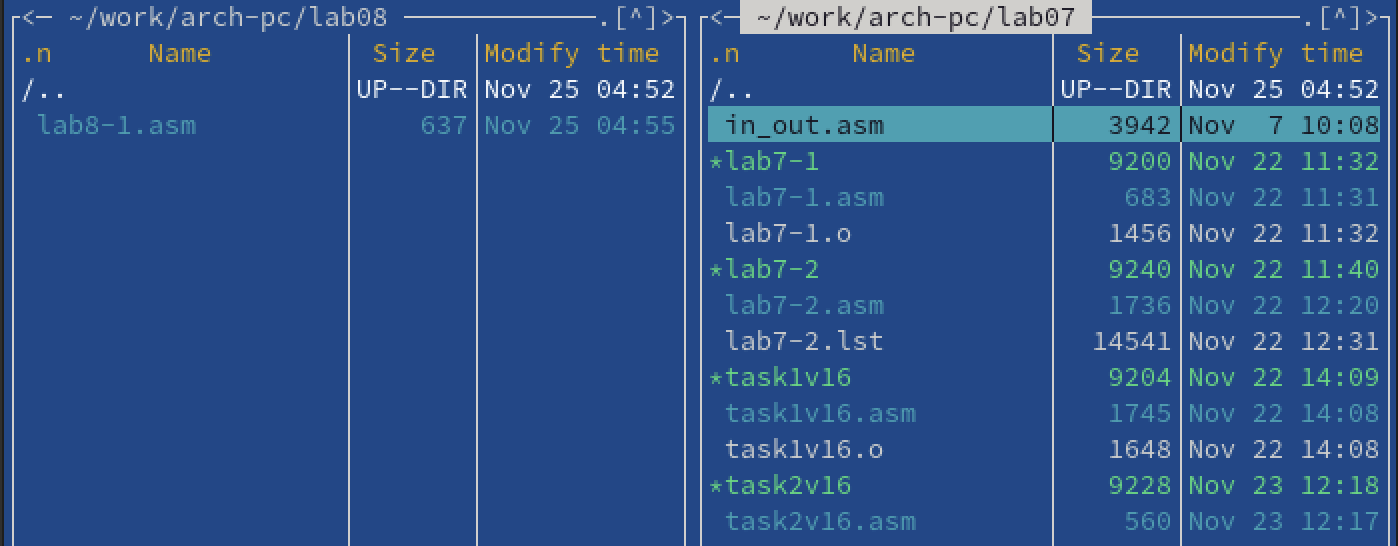
Создание папки и файла lab8-1.asm

Запустив Midnight commander, вставляем код из первого листинга.(рис.2)



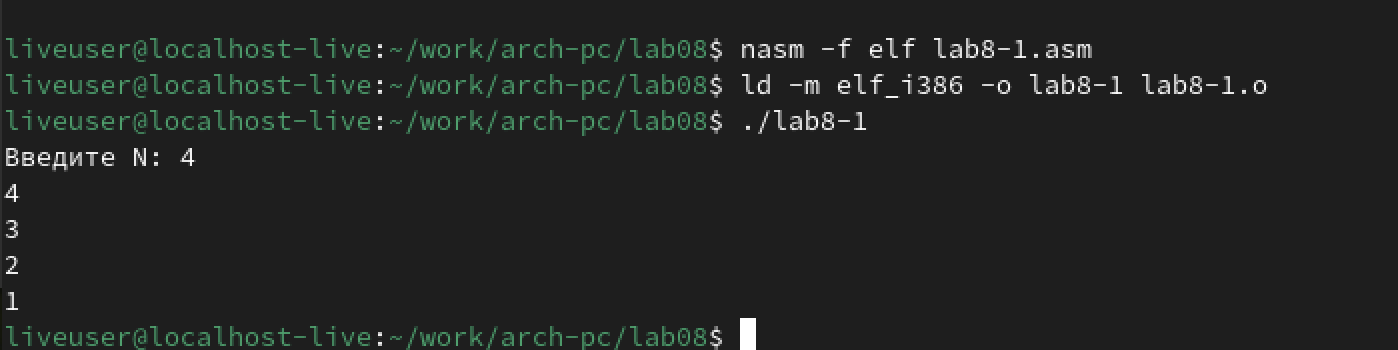
Вставляем код из листинга

Код должен запускать цикл и выводить каждую итерацию число, которое будет на единицу меньше предыдущего. Чтобы он работал перенесем файл in\_out.asm из папки предыдущей лабораторной работы(рис.3)



Переносим in\_out.asm

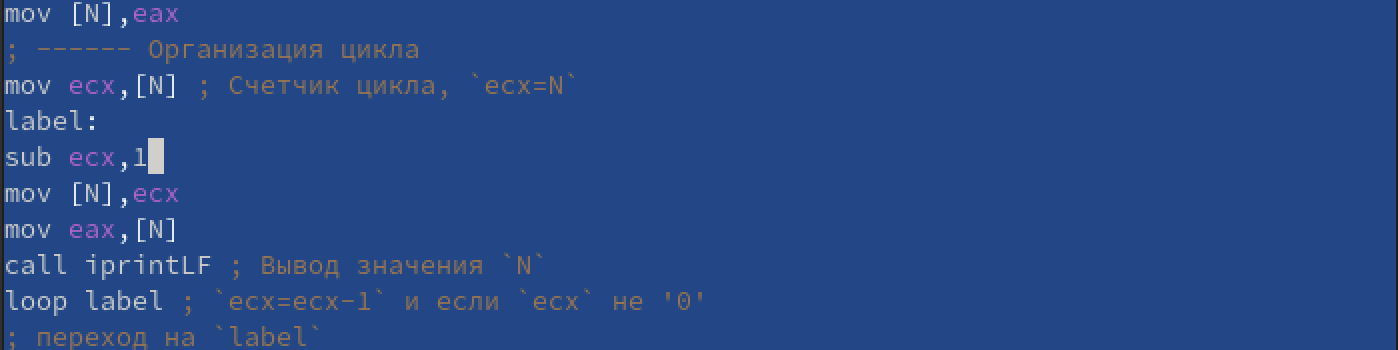
Запустим код и посмотрим результат(рис.4)



Запуск кода

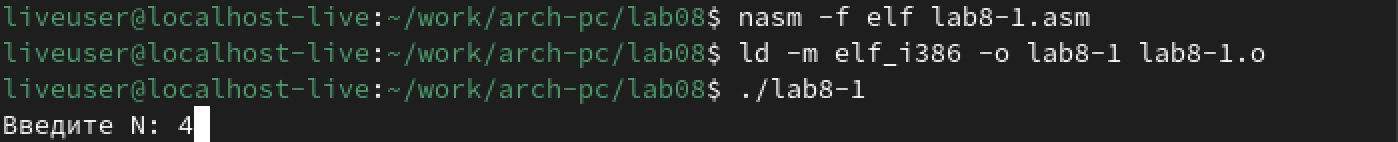
Видно, что программа выводит числа от N до единицы.

Теперь изменим код, чтобы в цикле отнималась единица у регистра ecx.(рис.5)



Изменение кода программы

Соберем файл еще раз и при запуске тоже введем 4.(рис.6)



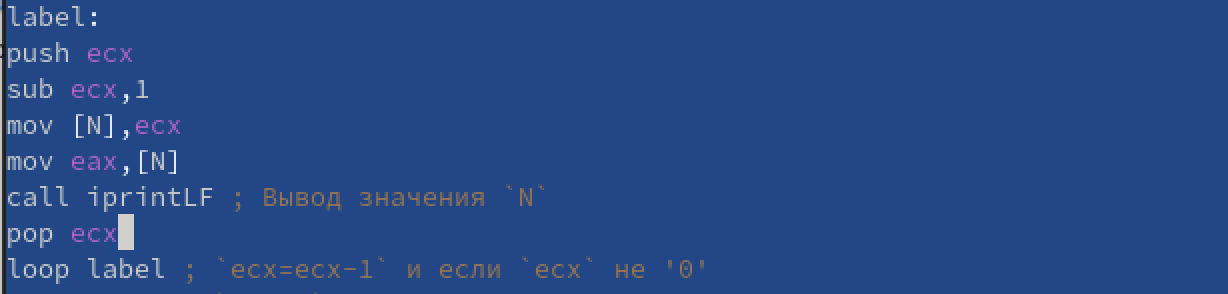
Повторный запуск программы

На выходе мы получаем бесконечный вывод, а значит количество проходов не совпадает с введенным значением.(рис.7)



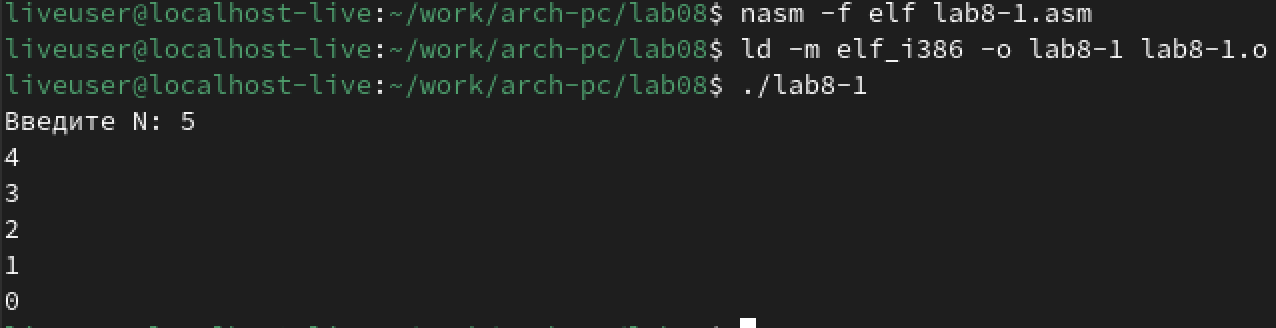
Бесконечный цикл на выходе

Тогда изменим программу так, чтобы она сохраняла значение регистра ecx в стек.(рис.8)



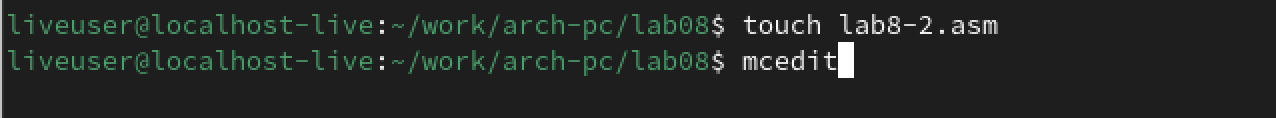
Редактируем код

Запустим код еще раз. На этот раз вывод правильный, значит код написан верно.(рис.9)



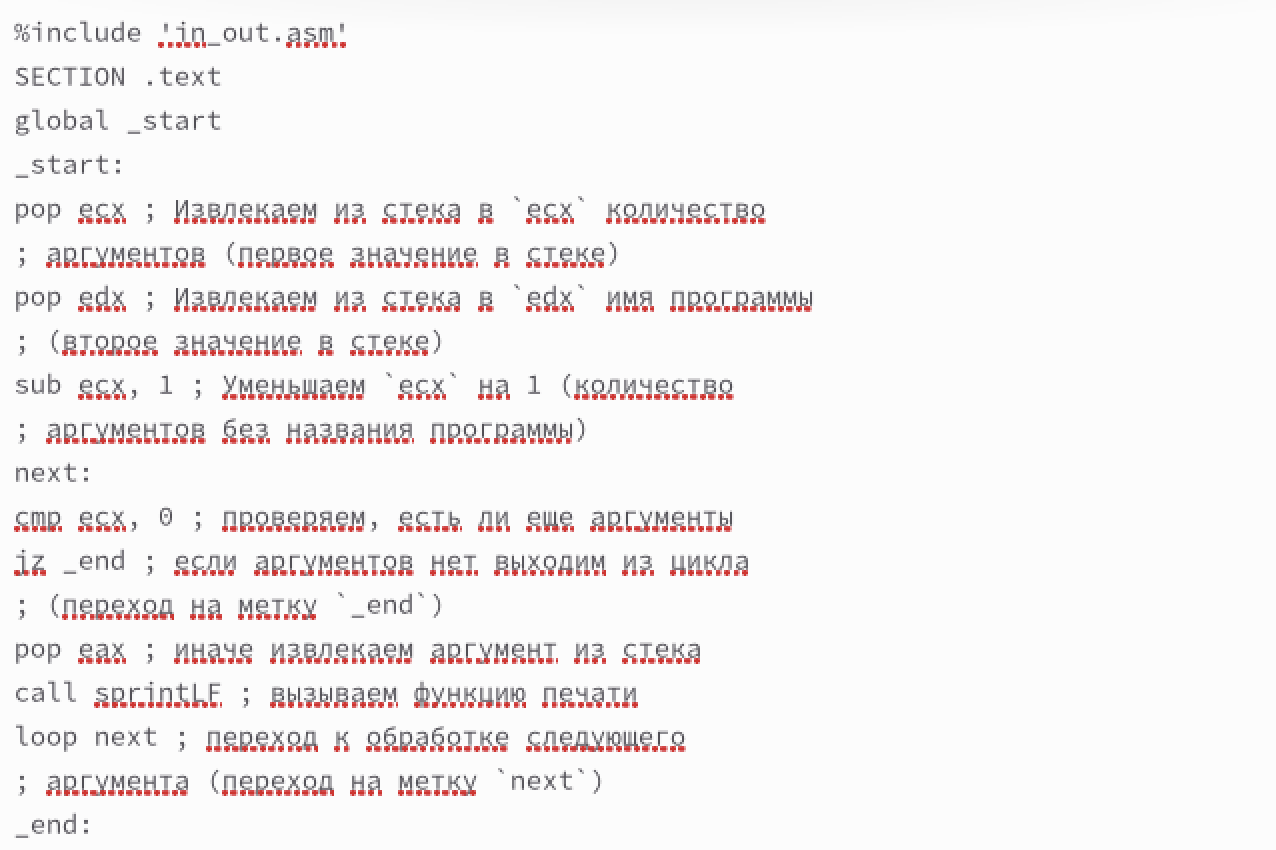
Запуск и вывод программы

Теперь создадим второй файл.(рис.10)



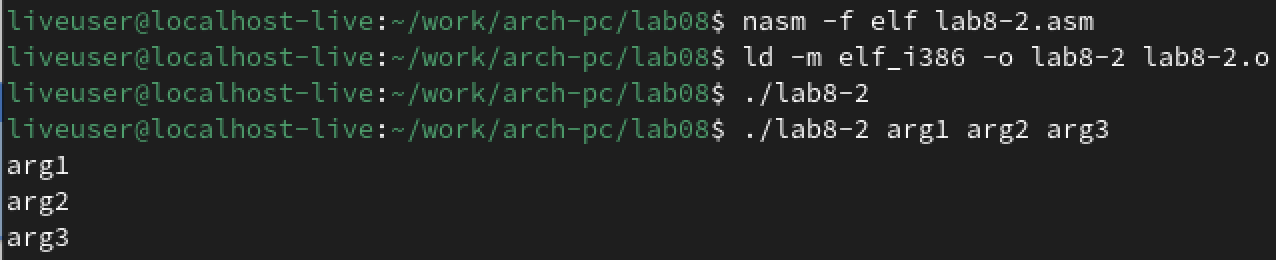
Создание второго файла

Вставим туда код из второго листинга.(рис.11)



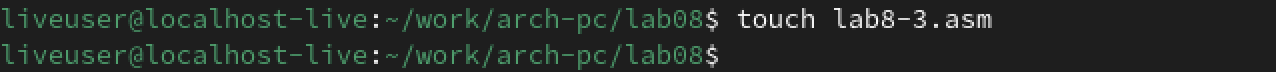
Вставляем код в файл

Теперь запустим программу и дадим при запуске 3 аргумента. На выходе мы получим их же. Программа выводит их по порядку.(рис.12)



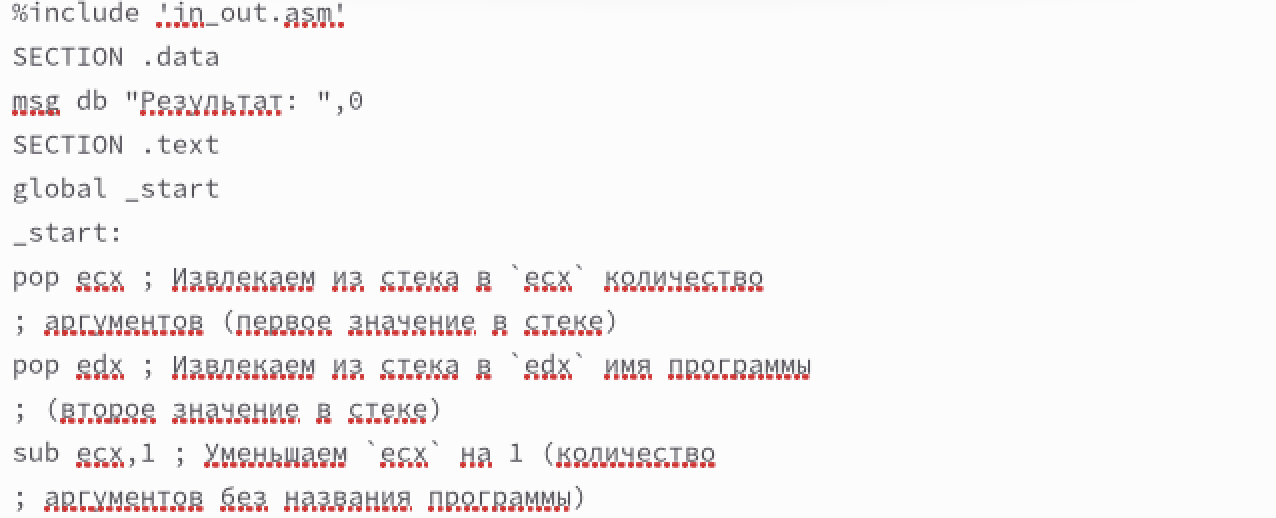
Проверка работы кода

Теперь создадим 3-й файл.(рис.13)



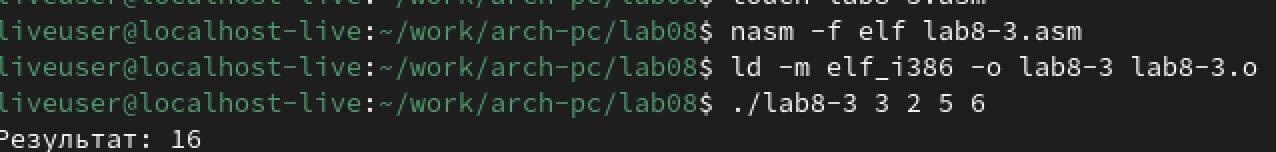
Создание третьего файла

Вставляем туда код из третьего листинга.(рис.14)



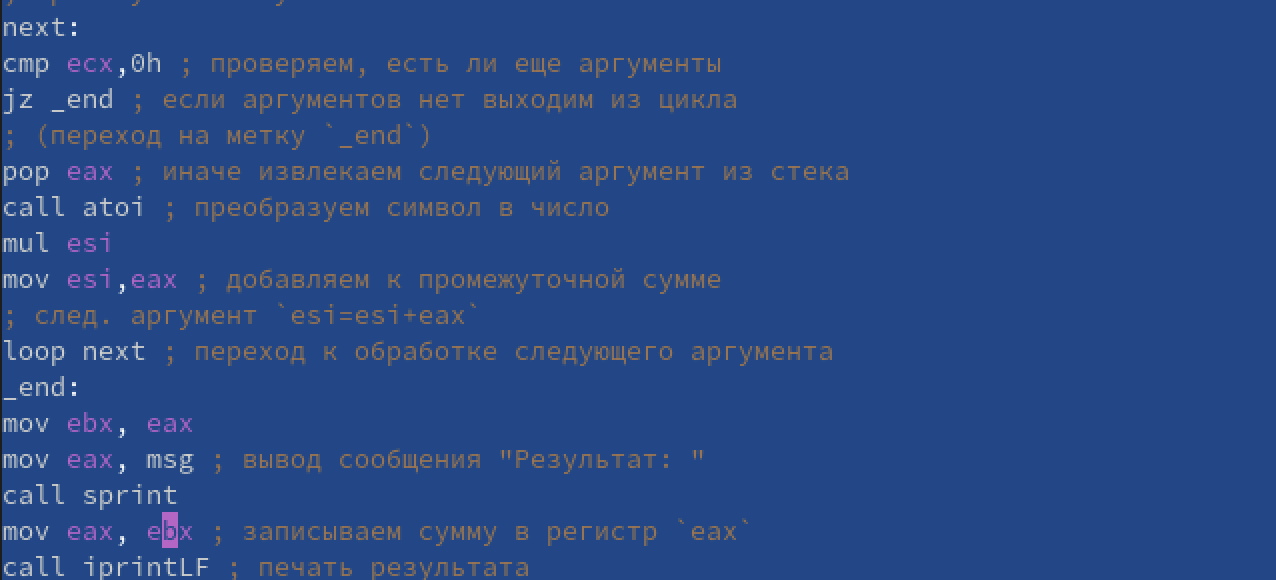
Вставляем код в файл

Теперь снова соберем файл. При запуске введем несколько чисел, которые программа должна сложить.(рис.15)



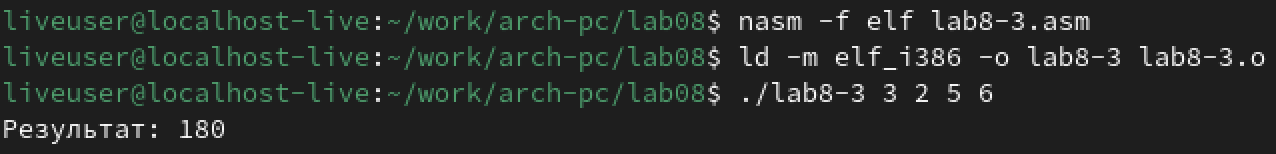
Запуск третьего файла

Все вывелось верно. Теперь попробуем немного отредактировать код, чтобы числа перемножались.(рис.16)



Редактируем файл, чтобы числа перемножались

Теперь запустим файл еще раз. как мы видим, все числа перемножились и программа вывела верный ответ(рис.17)



Запуск отредактированного кода

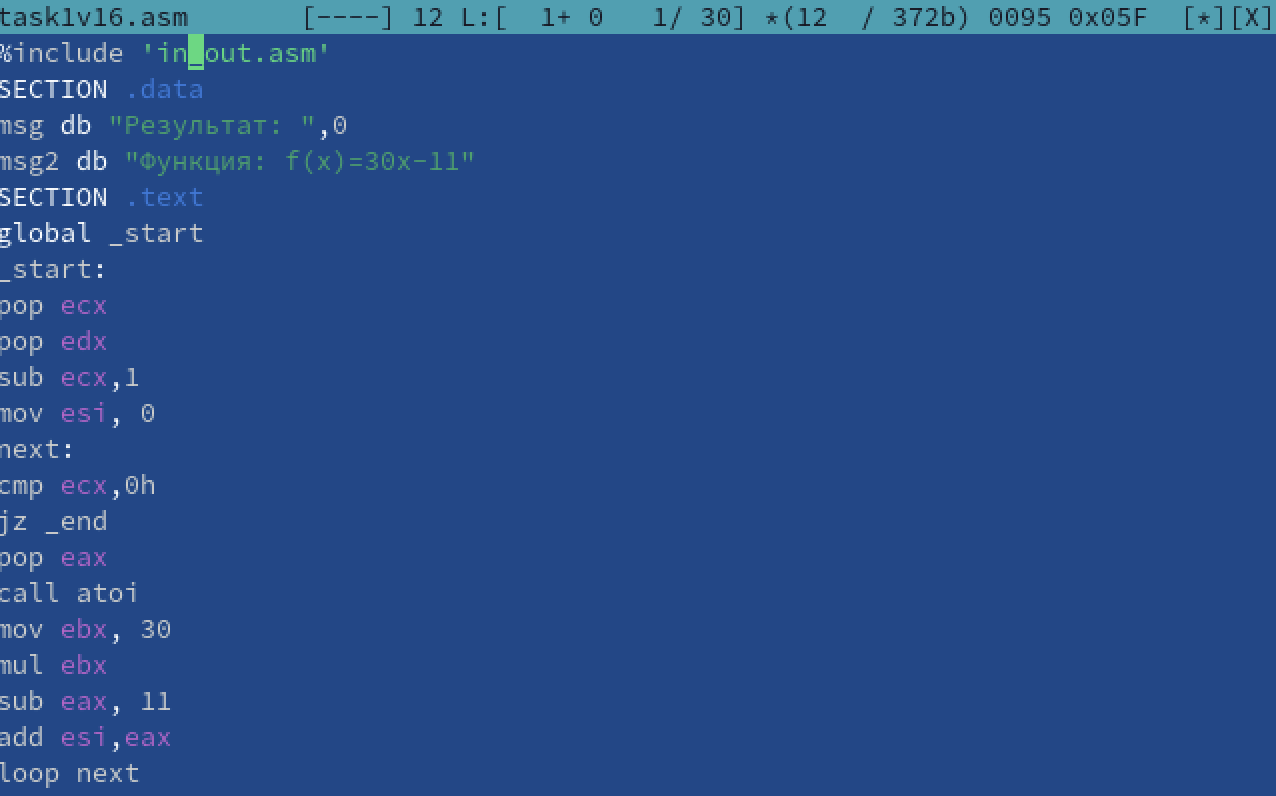
#Выполнение самостоятельной работы

Для начала создадим файл, где буем писать код.(рис.18)

Создание файла самостоятельной работы

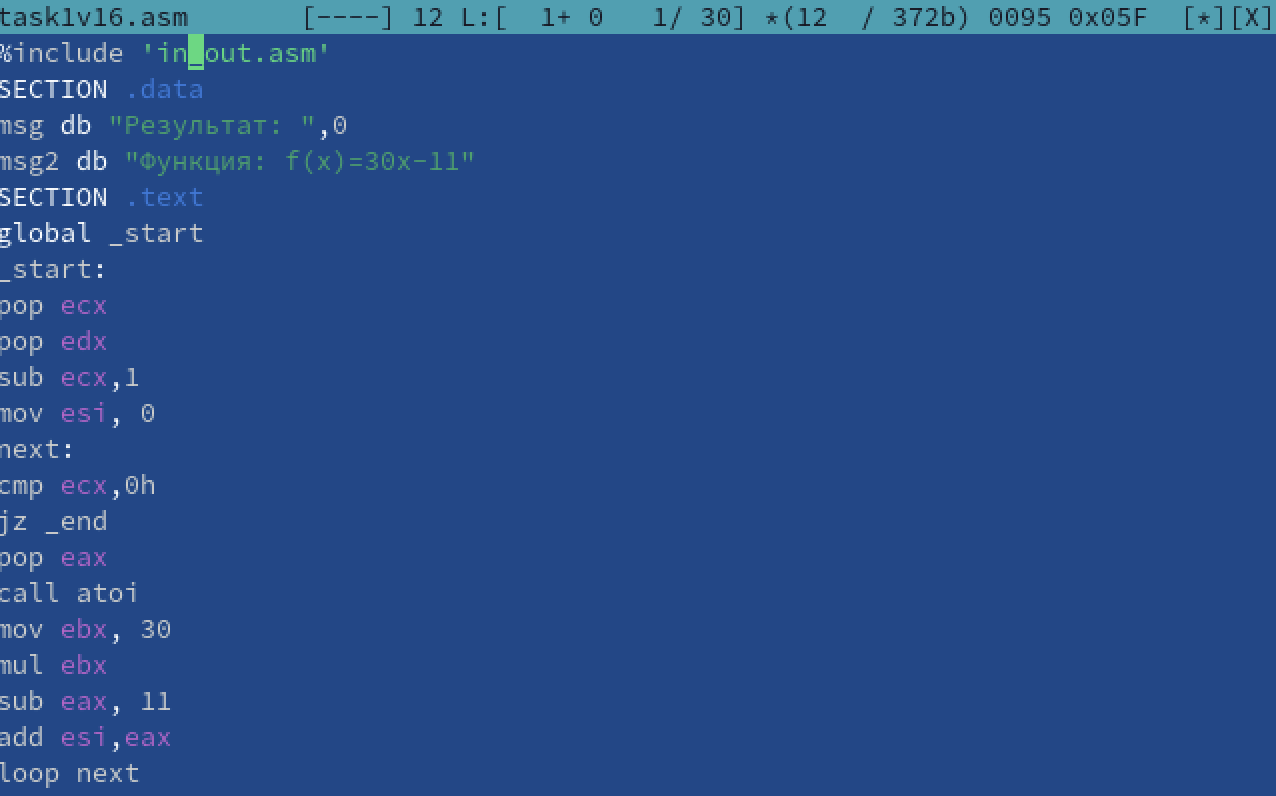
Создание файла самостоятельной работы

Далее откроем файл.(рис.19)



Открываем файл

Запишем код, который будет вычислять функцию из варианта с заданными аргументами 16 и затем сложит ответы при всех аргументах.(рис.20-21)



Код для задания 16 ч.1



Код для задания 16 ч.2

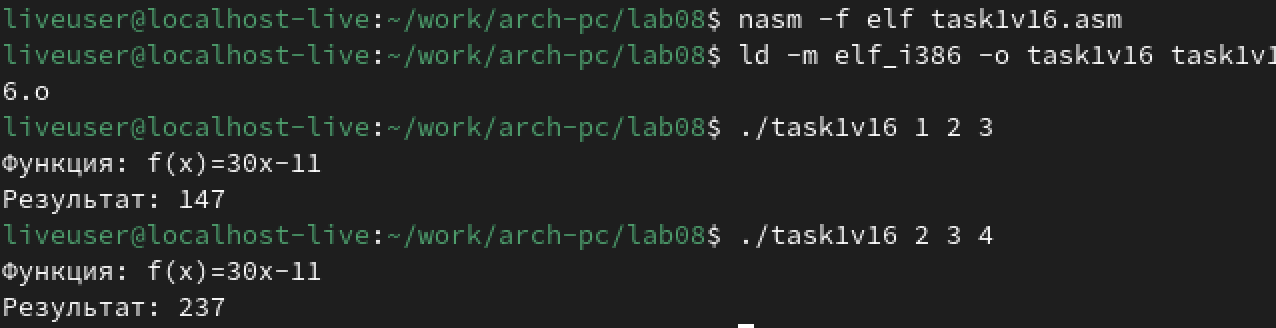
Опишем те строки кода, которые могут вызывать вопросы:

pop ecx - здесь из регистра достаем количество аргументов в стеке

esi отвечает за хранение промежуточных сумм.

cmp ecx необходимо для проверки, есть ли еще аргументы на входе.

Теперь запустим код и введем несколько разных аргументов для проверки.(рис.22)



Запуск кода для сложения значений функции

Код работает правильно, значит самостоятельная работа выполнена верно.

# 3 Выводы

В ходе выполнения данной лабораторной работы были получены знания о циклах на языке ассемблера и получены навыки о работе с ними.