Лабораторная работа №8

НКАбд-02-23

Выборнов Дмитрий Валерьевич

# 1 Цель работы

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

# 2 Задание

1. Реализация циклов в NASM.
2. Обработка аргументов командной строки.
3. Задание для самостоятельной работы.

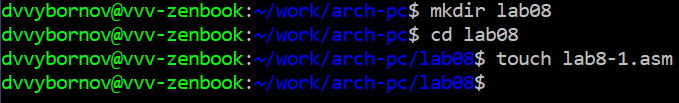
# 3 Теоретическое введение

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. Стек имеет вершину, адрес последнего добавленного элемента, который хранится в регистре esp (указатель стека). Противоположный конец стека называется дном. Значение, помещённое в стек последним, извлекается первым. Команда push размещает значение в стеке, т.е. помещает значение в ячейку памяти, на которую указывает регистр esp, после этого значение регистра esp увеличивается на 4. Данная команда имеет один операнд — значение, которое необходимо поместить в стек. Команда pop извлекает значение из стека, т.е. извлекает значение из ячейки памяти, на которую указывает регистр esp, после этого уменьшает значение регистра esp на 4. У этой команды также один операнд, который может быть регистром или переменной в памяти. Для организации циклов существуют специальные инструкции. Для всех инструкций максимальное количество проходов задаётся в регистре ecx. Наиболее простой является инструкция loop. Она 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

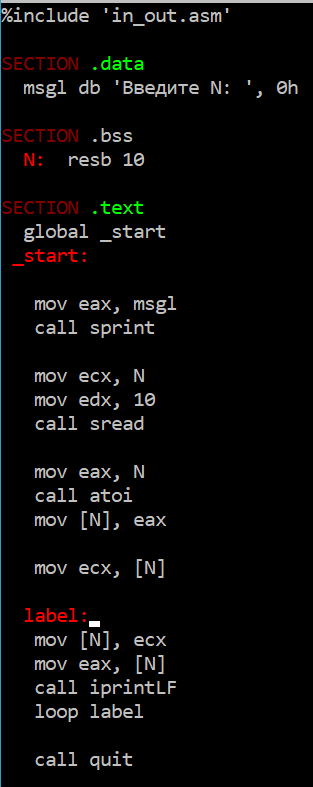
## 4.1 Реализация циклов в NASM.

Создаю каталоги и файлы для новой лабораторной работы.



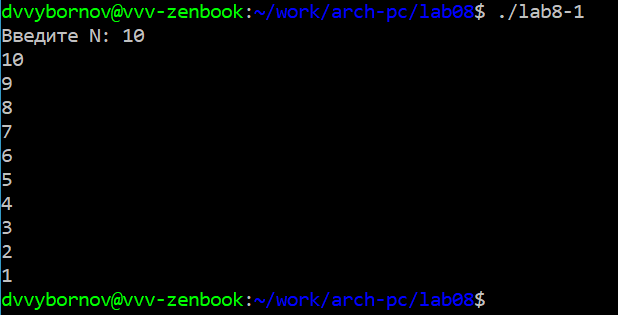
Первый шаг.

Ввожу в файл текст первой программы.



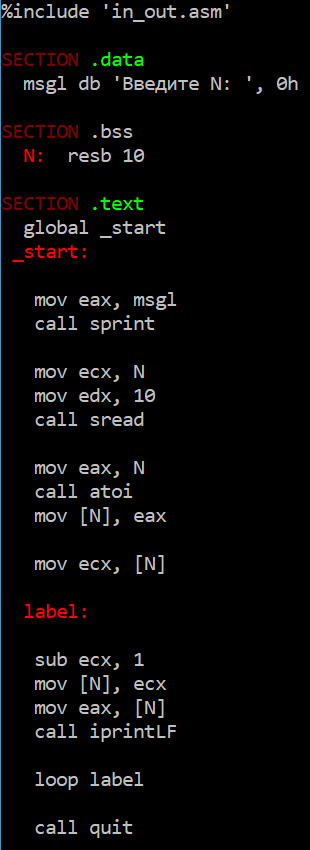
Второй шаг.

Создаю исполняемый файл и запускаю его.



Третий шаг.

Добавляю в программу изменение значения регистра ecx в цикле.



Четвёртый шаг.

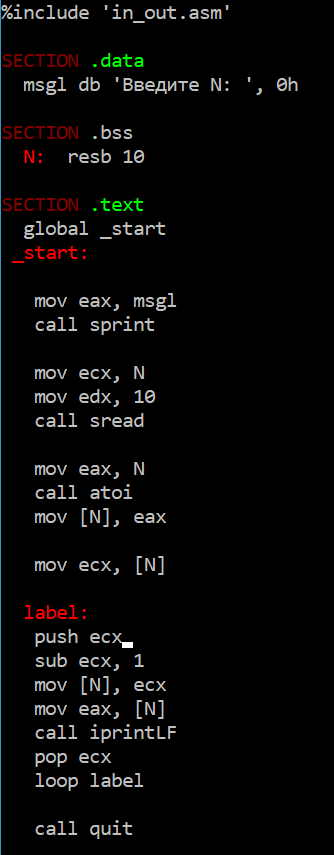
Запускаю изменённую программу.



Пятый шаг.

В данном случае число проходов цикла не соответствует значению N, так как за один проход значение регистра ecx уменьшается на 2.

Добавляю в текст программы комманды push и pop.



Шестой шаг.

Зарускаю изменённую программу.

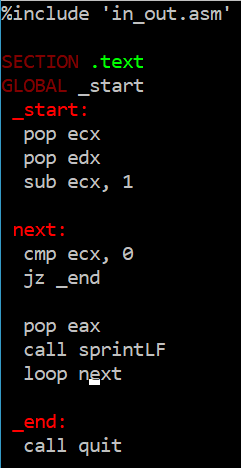


Седьмой шаг.

Число проходов соответствует значению N, но теперь вместо “10” выводится “0”.

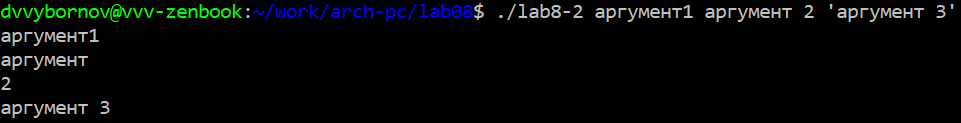
## 4.2 Реализация циклов в NASM.

Создаю новый файл и ввожу в него текст второй программы.



Восьмой шаг.

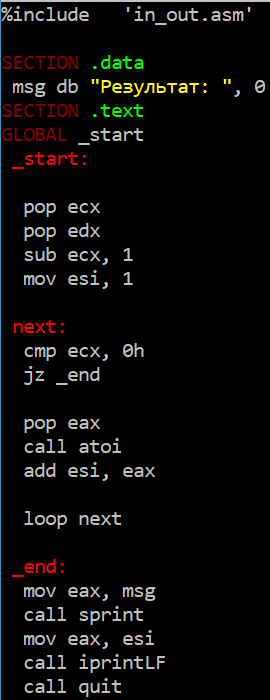
Запускаю программу с указанными аргументами.



Девятыйй шаг.

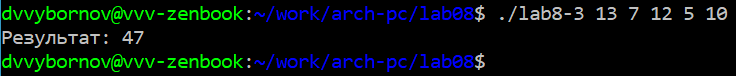
Программа обработала 4 аргумента, так как ‘аргумент’ и ‘2’ разделились.

Создаю новый файл и ввожу в него текст третьей программы.



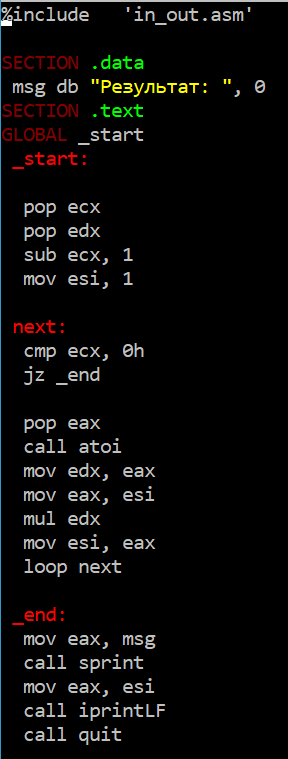
Десятый шаг.

Создаю машинный файл и запускаю его.



Одиннадцатый шаг.

Изменяю программу так, чтобы она выводила произведение аргументов.



Двенадцатый шаг.

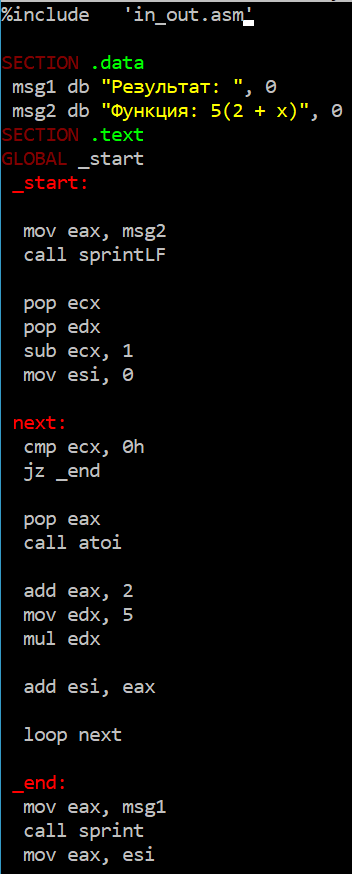
Создаю исполняемый файл и запускаю его.

Тринадцатый шаг.

Тринадцатый шаг.

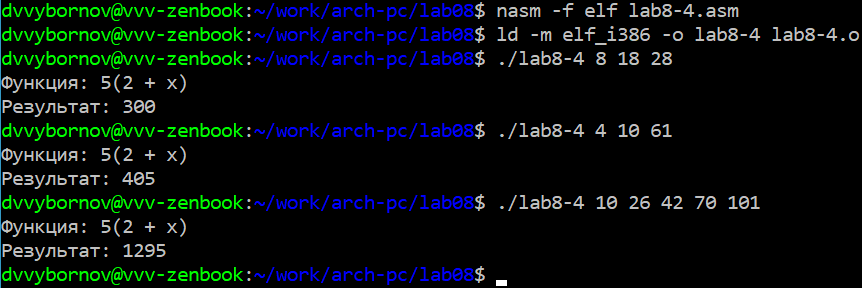
## 4.3 Задание для самостоятельной работы.

Создаю программу, которая находит сумму значений функции 5(2 + x).



Первый шаг самостоятельного загания.

Проверяю работу программы с несколькими наборами аргументов.



Второй шаг самостоятельного задания.

# 5 Выводы

Завершив эту лабораторную работу, я приобрёл навыки написания программ с использованием циклов и обработкой аргументов командной строки NASM.