Отчет по лабораторной работе №8

Дисциплина: Архитектура компьютера

Зайцева Ульяна Владимировна

Содержание

# 1 Цель работы

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

# 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. При помещении значения в стек указатель стека уменьшается, а при извлечении — увеличивается.

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

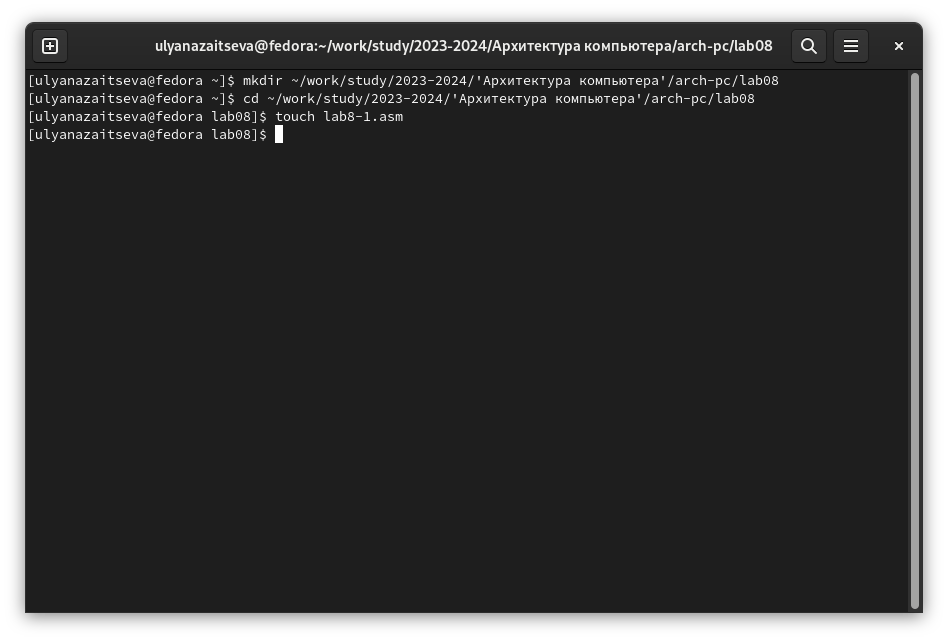
Команда pop извлекает значение из стека, т.е. извлекает значение из ячейки памяти, на которую указывает регистр esp, после этого уменьшает значение регистра esp на 4. У этой команды также один операнд, который может быть регистром или переменной в памяти. Нужно помнить, что извлечённый из стека элемент не стирается из памяти и остаётся как “мусор”, который будет перезаписан при записи нового значения в стек.

Для организации циклов существуют специальные инструкции. Для всех инструкций максимальное количество проходов задаётся в регистре ecx. Наиболее простой является инструкция loop. Она позволяет организовать безусловный цикл

# 4 Выполнение лабораторной работы

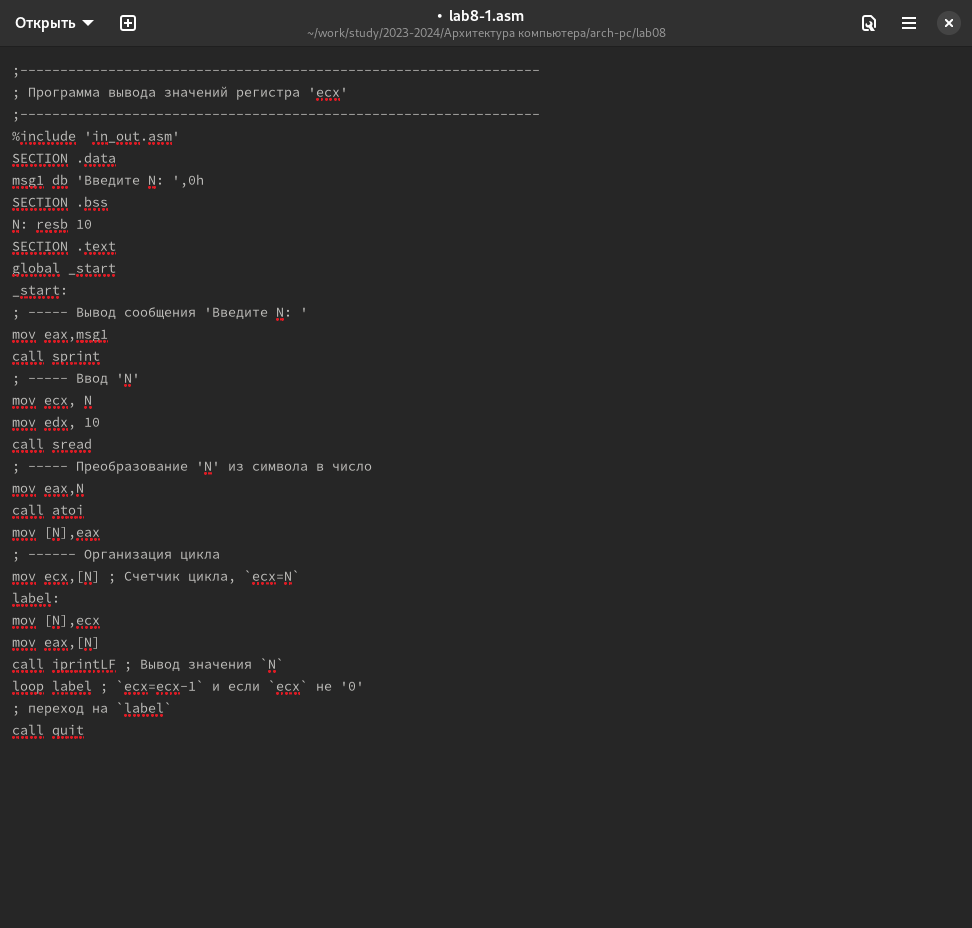
1. Реализация циклов в NASM

Перехожу в нужный мне каталог, создайте каталог для программ лабораторной работы № 8, перехожу в него и создаю файл lab8-1.asm(рис. ??).

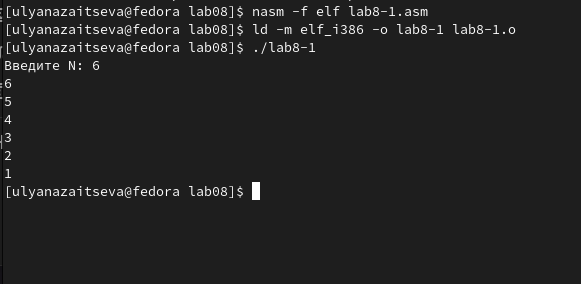


Создание рабочей папки и файла

Ввожу в файл lab8-1.asm текст программы из листинга 8.1.(рис. ??) Создаю исполняемый файл и проверяю его работу.(рис. ??)



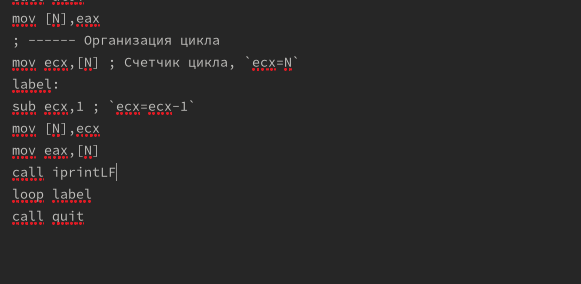
Ввод текста программы листинга 8.1



Проверка работы

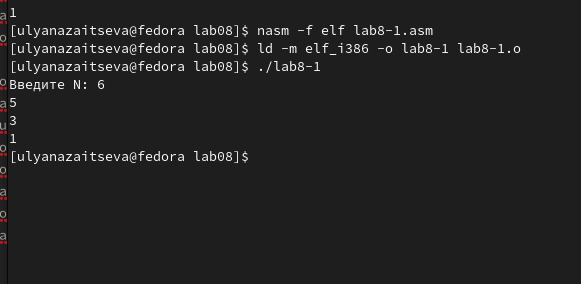
Данная программа выводит числа от N до 1 включительно.

Изменяю текст программы, добавив изменение значения регистра ecx в цикле. (рис. ??).



Изменение текста программы

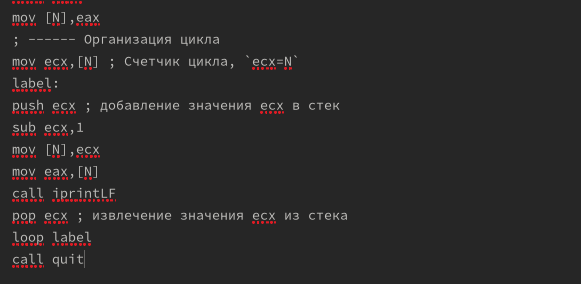
Создаю файл и проверяю работу(рис. ??)



Проверка работы программы

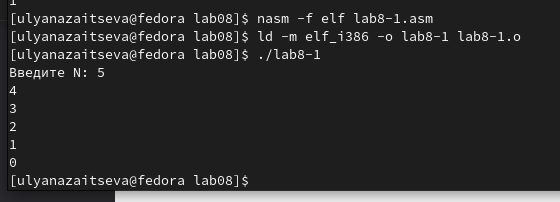
В данном случае число проходов цикла не соответствует введенному значению.

Вношу изменения в текст программы, добавив команды push и pop для сохранения значения счетчика цикла loop. (рис. ??).



Изменения

Создаю файл и проверяю его работу(рис. ??)

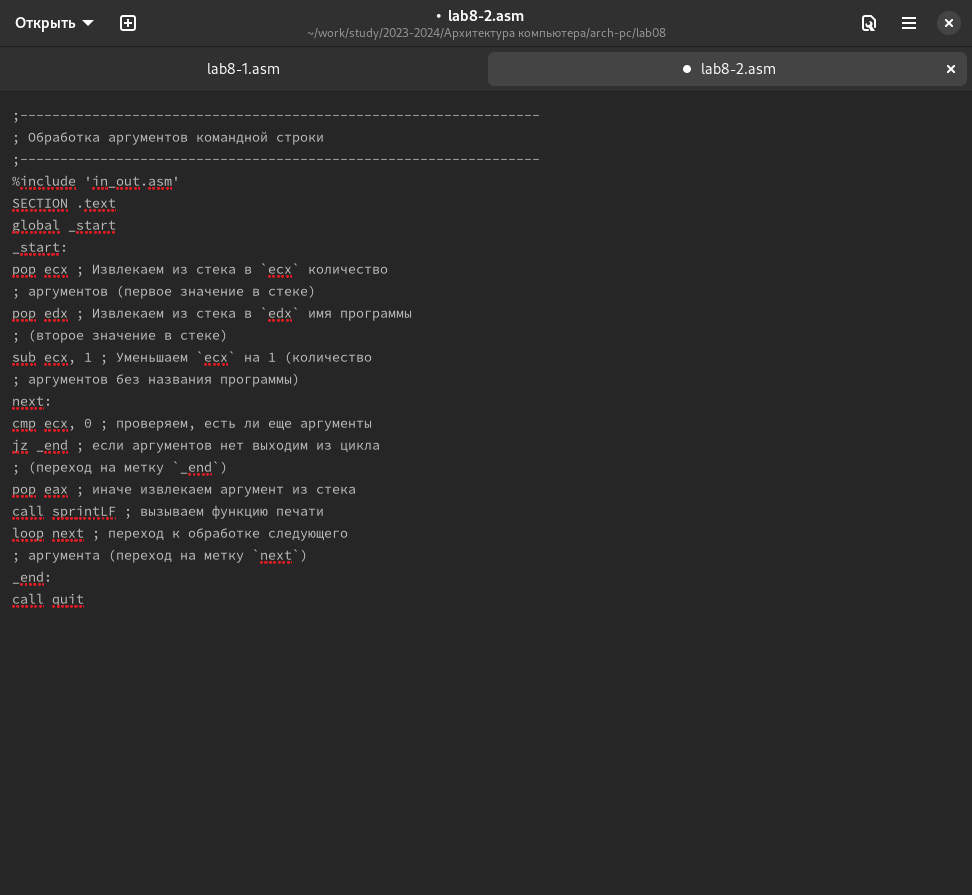


Проверка работы программы

В данном случае число проходов цикла соответствует введенному с клавиатуры значению и выводит числа от N-1 до 0 включительно.

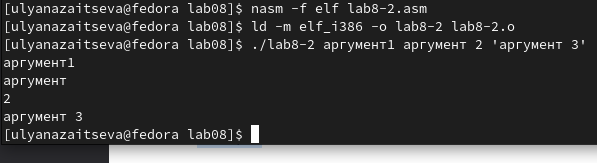
1. Обработка аргументов командной строки

Создаю файл lab8-2.asm и ввожу в него программу из листинга 2(рис. ??)



Листинг 2

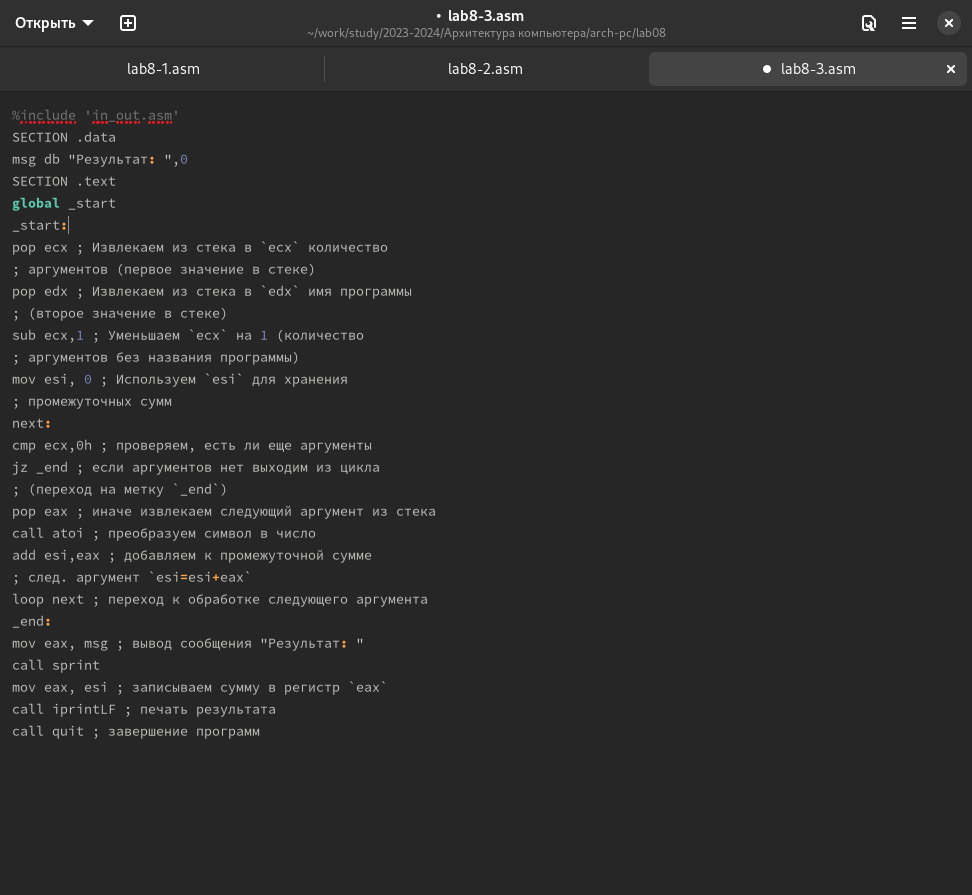
Создаю файл и проверяю работу(рис. ??)



Проверка

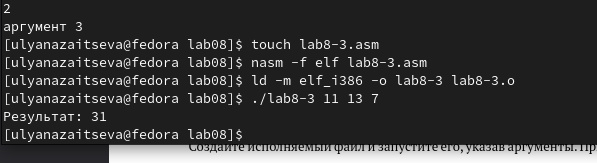
Программа вывела 4 аргумента, т.к. аргумент 2 не взят в кавычки программа считывает “2” как отдельный аргумент из-за пробела.

Создаю файл lab8-3.asm и ввожу в него текст программы из листинга 8.3. (рис. ??).



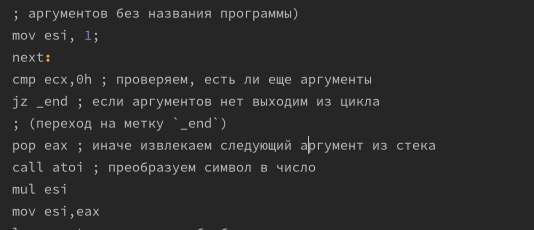
Листинг 3

Создаю файл и проверяю работу(рис. ??).



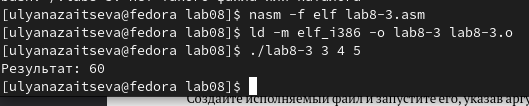
Проверка работы

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



Изменяю текст программы

Проверяю работу(рис. ??).

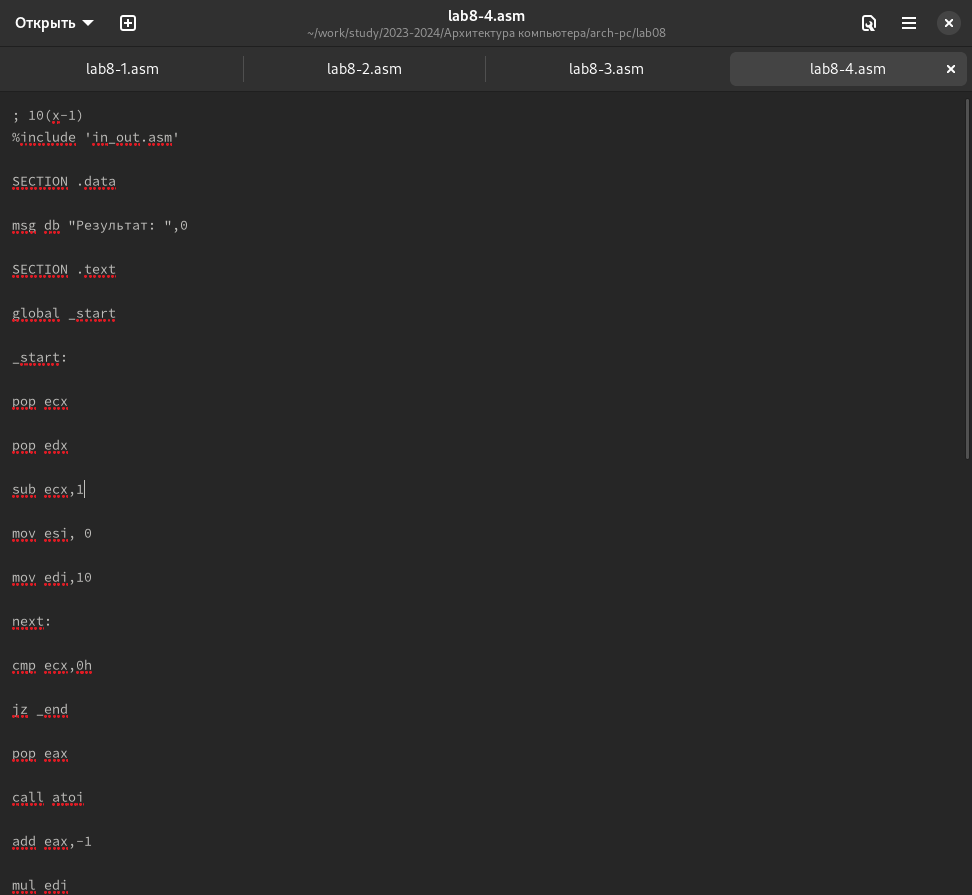


Измененный листинг 8.3

Программа работает исправно

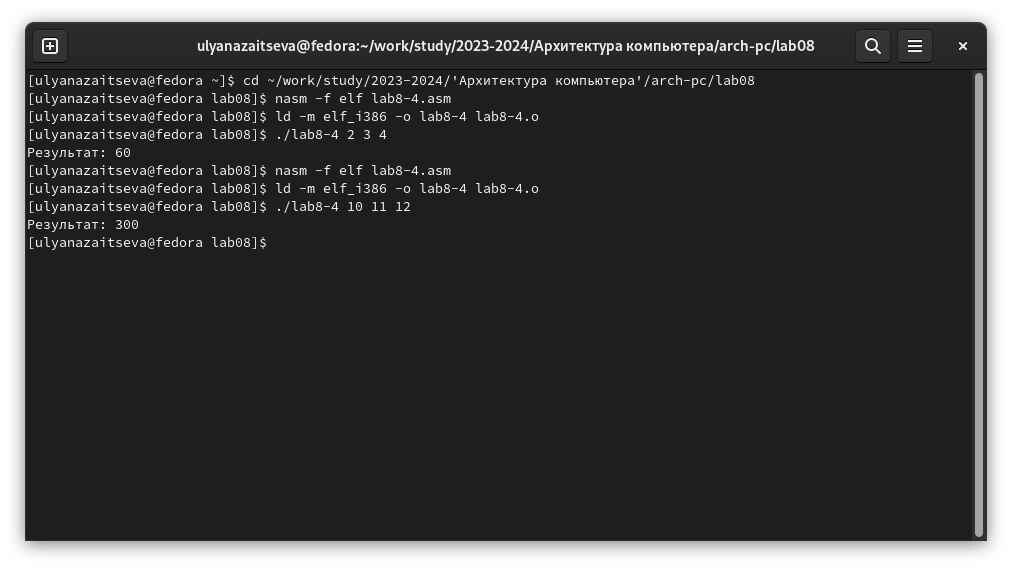
1. Задание для самостоятельной работы

Пишу программу, которая находит сумму значений функции 𝑓(𝑥) для 𝑥 = 𝑥1, 𝑥2, …, 𝑥 . Мой вариант - 17. Выражение 10(x-1)(рис. ??).



Программа 8-4

Компилирую и проверяю работу на нескольких х(рис. ??).



Проверка

Программа работает исправно

# 5 Выводы

При выполнение работы я приобрела навыки написания программ с использованием циклов и обработкой аргументов командной строки