Отчёт по лабораторной работе №8

Дисциплина: Архитектура компьютеров

Лазарев Даниил Михайлович

Содержание

# 1 Цель работы

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

# 2 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.  
Для стека существует две основные операции:

• добавление элемента в вершину стека (push); • извлечение элемента из вершины стека (pop).

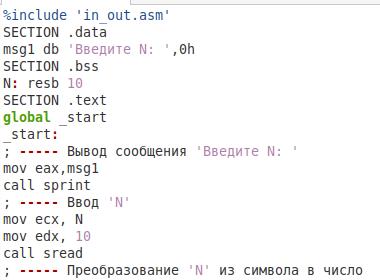
# 3 Выполнение лабораторной работы

Создадим каталог для программ лаб. работы н.8, перейдем в него и создадим файл “lab8-1.asm” (рис. ??)



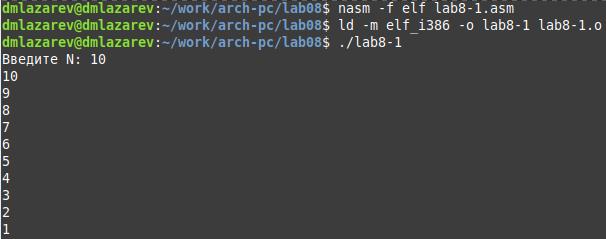
Создание файла в каталоге

Введем в созданный файл текст программы из предложенного нам листинга 8.1 (рис. ??)



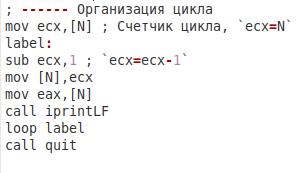
Код программы в файле

Создадим исполняемый файл и запустим его, предварительно скопировав из предыдущей лаб. работы файл “in\_out.asm” для корректной работы (рис. ??)



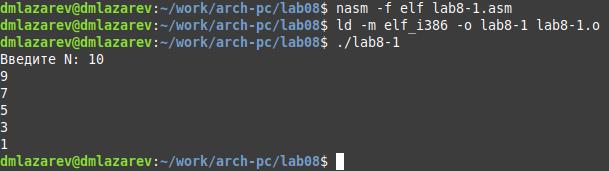
Преобразование в исполняемый файл

Далее изменим часть текста. (рис. ??)



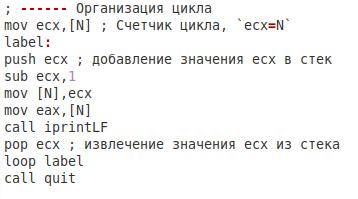
Измененный код программы

Преобразуем в исполняемый файл и проверим правильность выполнения. (рис. ??)



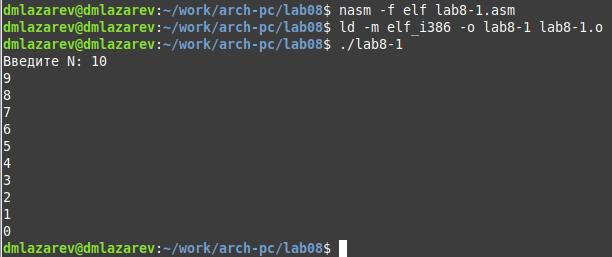
Преобразование измененного файла

Изменим текст листинга повторно. (рис. ??)



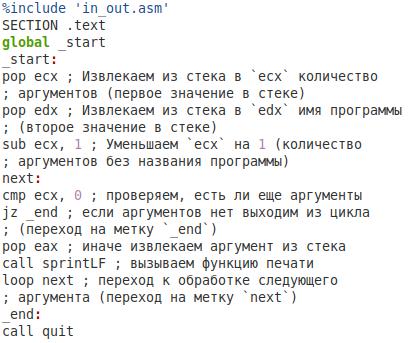
Изменения исходного текста

Преобразуем файл в исполняемый и проверим правильность выполнения. (рис. ??)



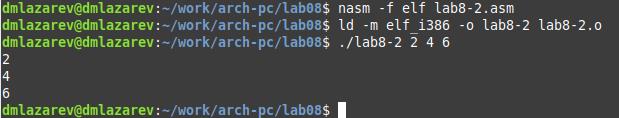
Преобразование файла

Создадим файл “lab8-2.asm” и вставим в него предложенный нам листинг 8.2 (рис. ??)



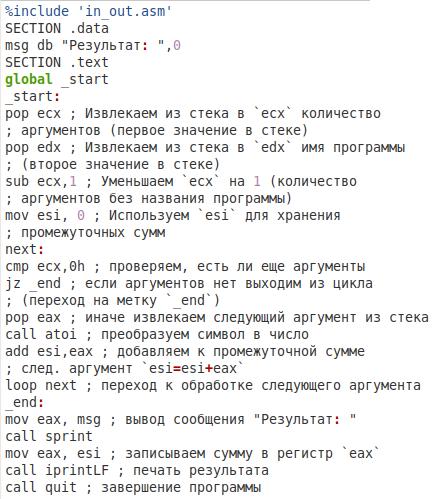
Текст программы в файле

Преобразуем файл “lab8-2.asm” в исполняемый и проверим правильность выполнения. (рис. ??)



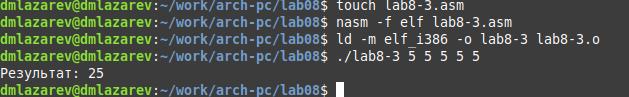
Преобразование файла в исполняемый

Создадим файл “lab8-3.asm” и вставим в него предложенный нам листинг 8.3. (рис. ??)



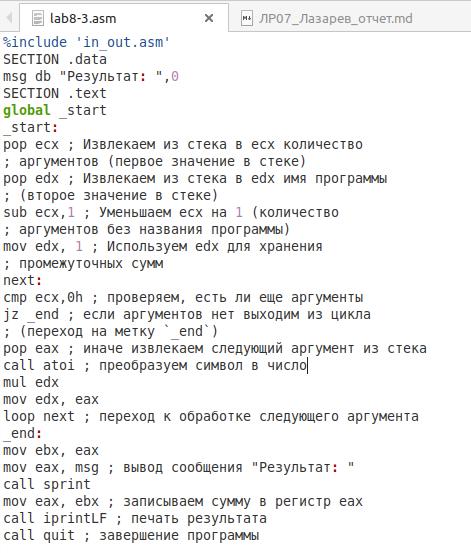
Текст программы в файле

Преобразуем файл “lab8-3.asm” в исполняемый и проверим правильность выполнения. (рис. ??)



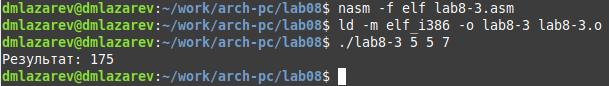
Преобразование файла в исполняемый

Изменим текст листинга 8.3 так, чтобы аргументы после подставления в функцию не складывались, а умножались. (рис. ??)



Исправленный текст программы

Преобразуем файл в исполняемый и проверим работоспособность (рис. ??)



Преобразование файла

# 4 Выполнение самостоятельной работы

Основываясь на результате файла “variant.asm” из лаб. работы н.6 выберем из таблицы 8.1 9 номер варианта. (рис. ??)

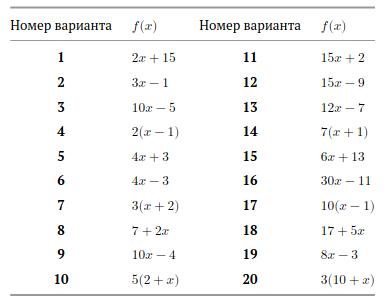
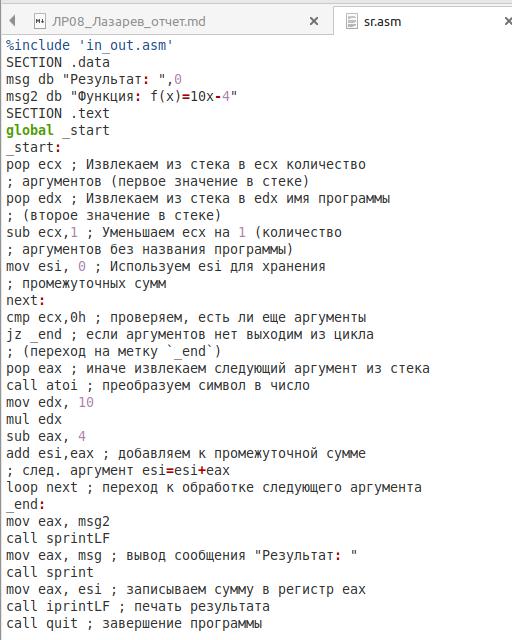


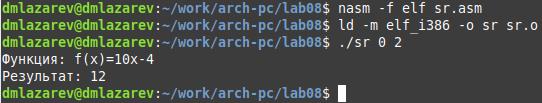
Таблица вариантов

Самостоятельно напишем код так, чтобы после подставления в функцию “10х-4” все переменные суммировались. (рис. ??)



Код программы

Преобразуем написанный нами файл в исполняемый и проверим правильность выполнения. (рис. ??)



Преобразование файла

# 5 Выводы

В ходе лабораторной работы мы освоили навыки написания программ с использованием циклов и обработкой аргументов командной строки.