Отчёт по лабораторной работе №8

Отчёт по лабораторной работе №8

Буриева Шахзода Акмаловна

Содержание

# 1 Цель работы

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

# 2 Теоретическое введение

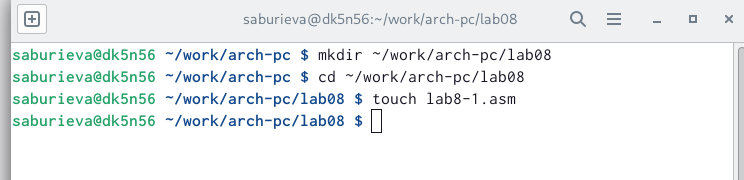
Стек — это структура данных, организованная по принципу LIFO («Last In — First Out» или «последним пришёл — первым ушёл»). Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основной функцией 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. оманда 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.Существует ещё две команды для добавления значений в стек. Это команда pusha, которая помещает в стек содержимое всех регистров общего назначения в следующем порядке: ах, сх, dx, bх, sp, bp, si, di. А также команда pushf, которая служит для перемещения в стек содержимого регистра флагов. Обе эти команды не имеют операндов.

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.

Иструкция loop 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

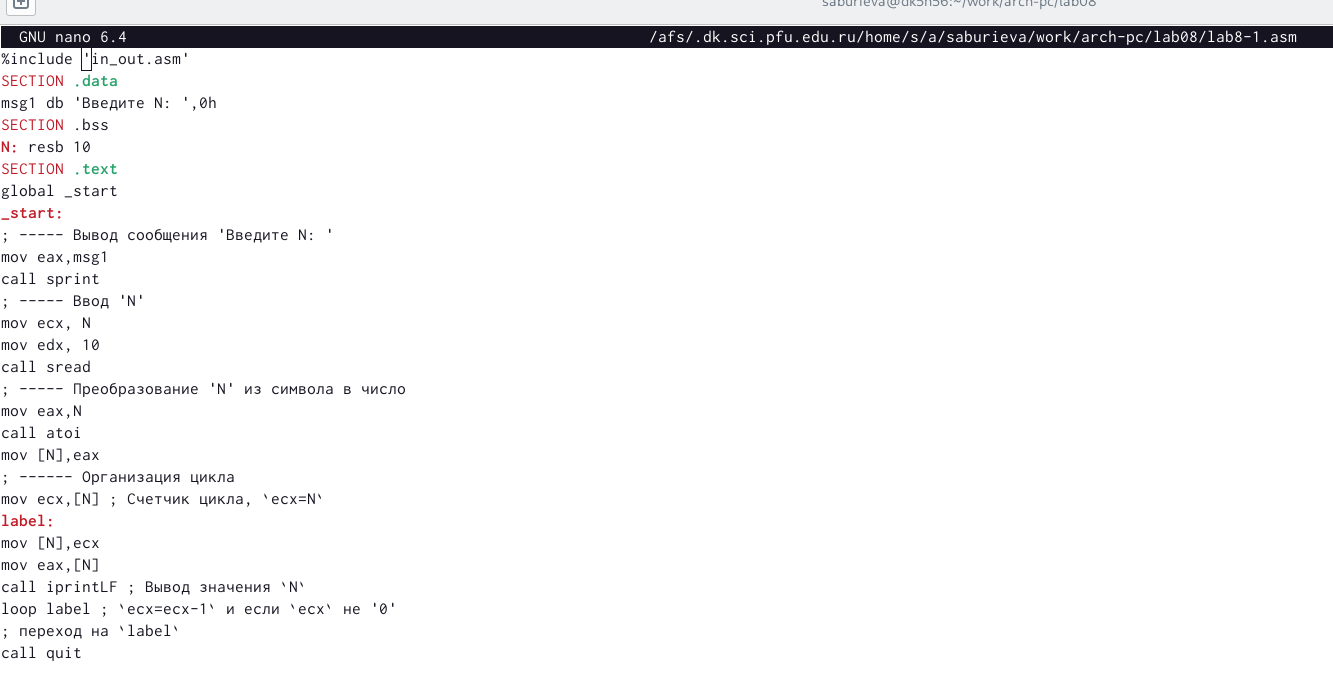
# 3 Выполнение лабораторной работы

Создала каталог для программам лабораторной работы №8, перешла в него и создала файл lab8-1.asm.



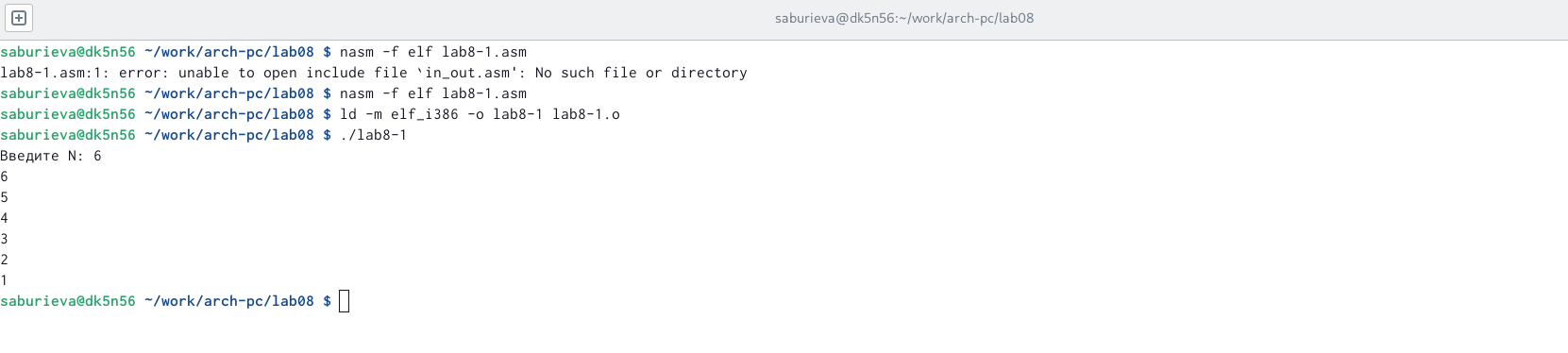
Создание каталога lab8,переход в него и создание файла lab8-1.asm

Рассмотрела пример программы вывода значений регистра ecx. Ввела в файл lab8-1.asm текст программы из листинга.



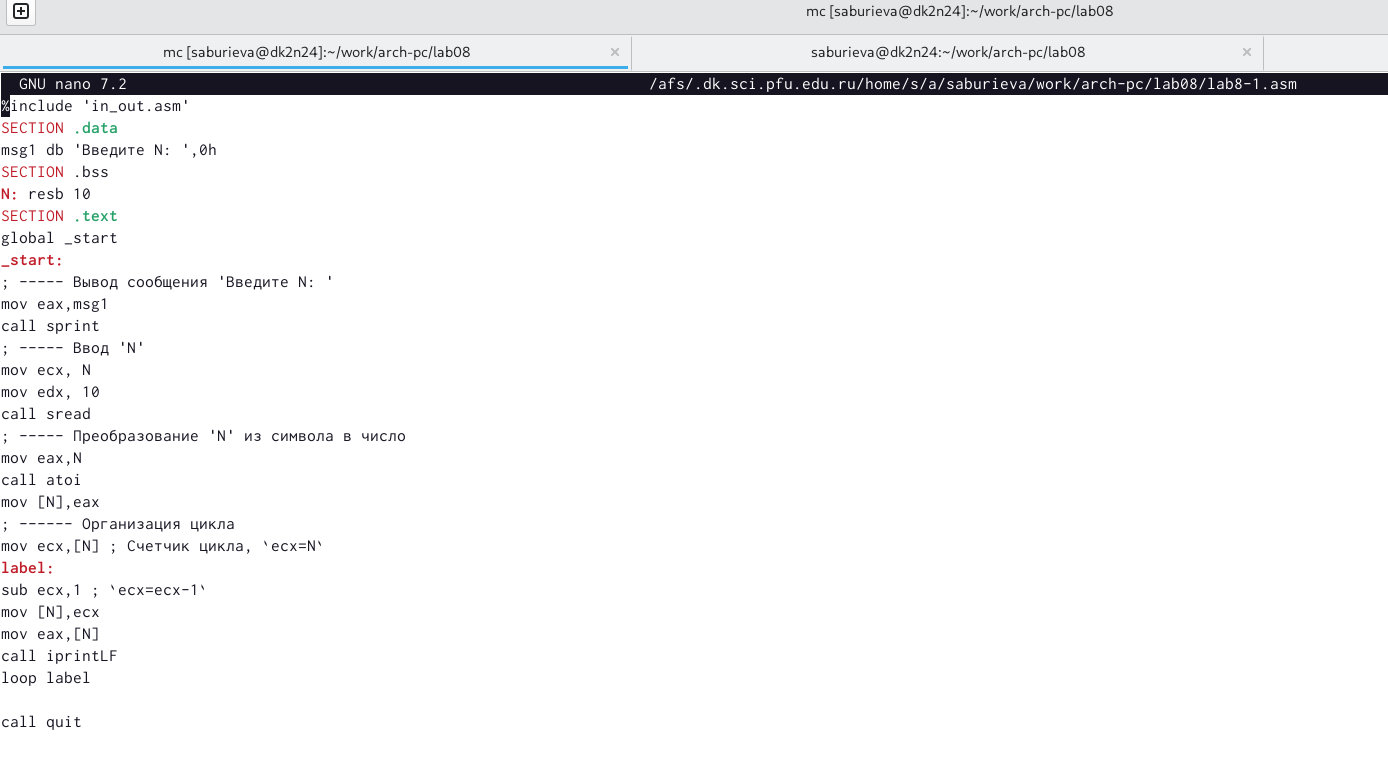
Ввод текста программы в файл lab8-1.asm

Создала исполняемый файл и запустила его.Данный пример показывает, что использование регистра ecx в теле цилка loop может привести к некорректной работе программы.На запрос “Введите N” ввела число 6.Программа вывела числа от 1 до 6 в порядке убывания.В данном случае число проходов цикла не соответсвует значению N введенному с клавиатуры.



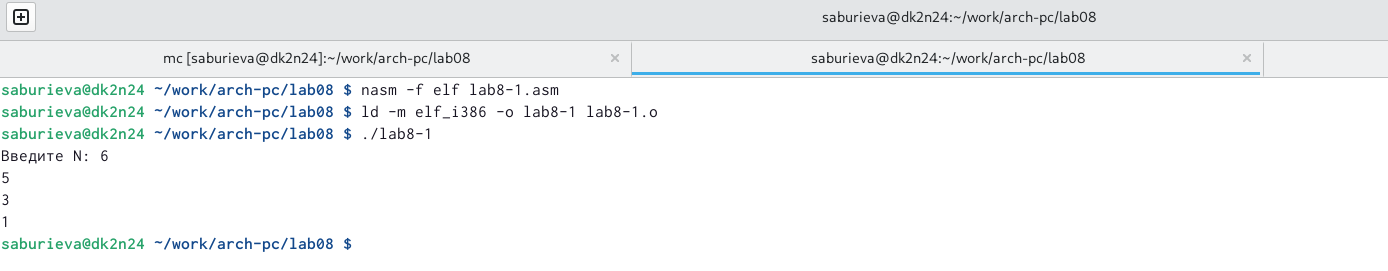
Создание исполняемого файла и его запуск

Изменила текст программы добавив изменение значение регистра ecx в цикле.



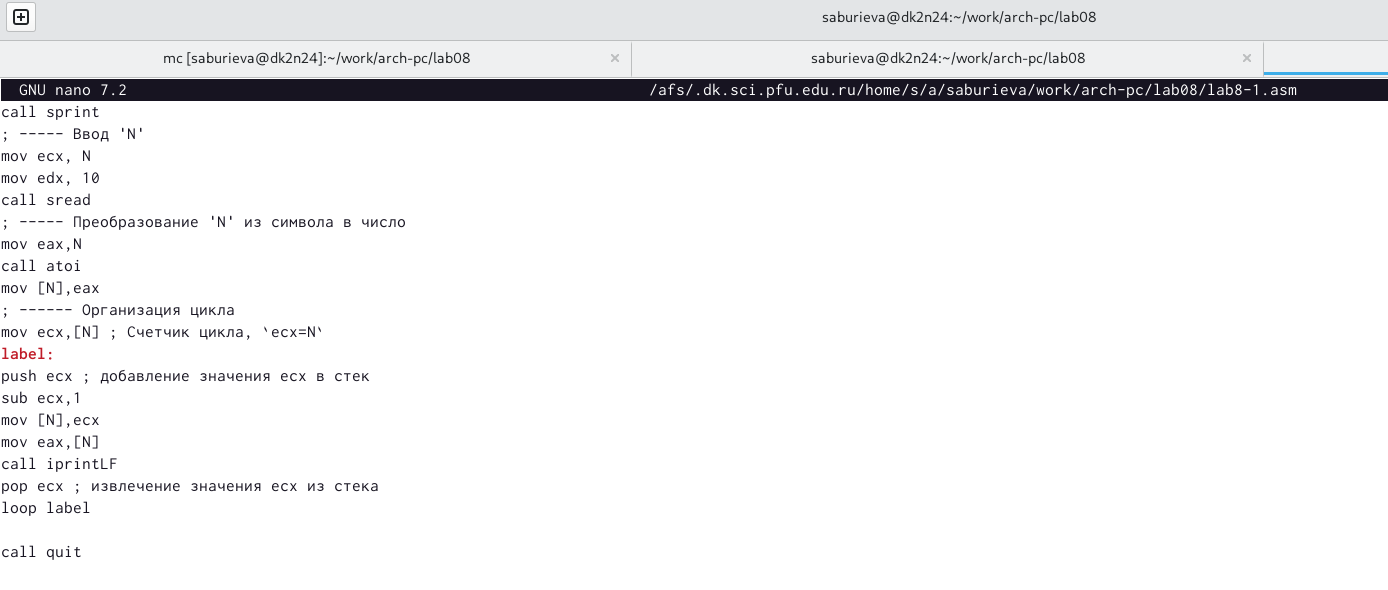
Изменение текста программы lab8-1.asm

Создала исполняемый файл и запустила его.На запрос “Введите N” ввела число 6.Программа вывела нечетные числа в порядке убывания.



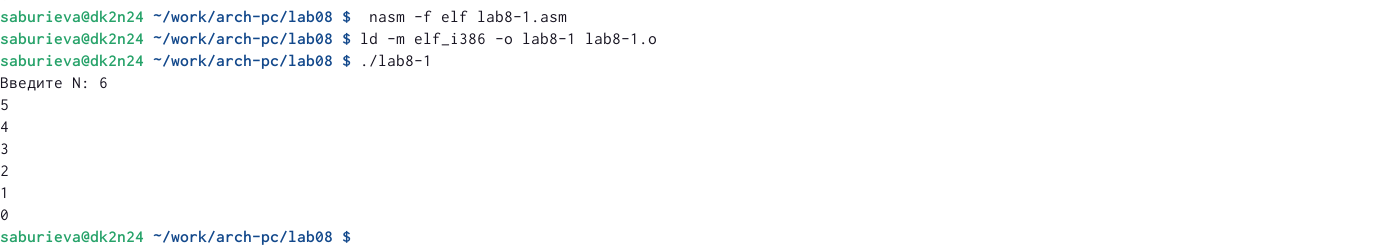
Создание исполняемого файла и его запуск

Внесла изменения в текст программы добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop.



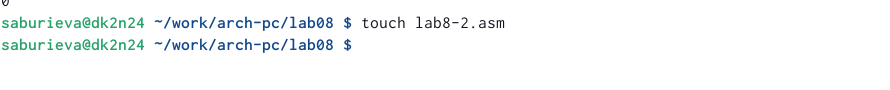
Изменение текста программы lab8-1.asm

Создала исполняемый файл и запустила его.На запрос “Введите N” ввела число 6.Программа вывела числа от 0 до 6 в порядке убывания.Да,в данном случае число проходов цикла соответсвует значению N введенному с клавиатуры.



Создание исполняемого файла и его запуск

Создала файл lab8-2.asm в каталоге ~/work/arch-pc/lab08.



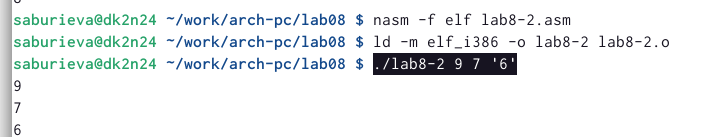
Создание файла lab8-2

Ввела в него текст программы из листинга.



Ввод текста программы в файл lab8-2.asm

Создала исполняемый файл и запустила его, указав аргументы.



Создание исполняемого файла и его запуск

Создала файл lab8-3.asm в каталоге ~/work/arch-pc/lab08.



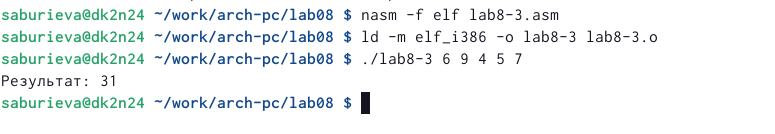
Создание файла lab8-3

Ввела в него текст программы из листинга для вычисления суммы аргументов командной строки.



Ввод текста программы в файл lab8-3.asm

Создала исполняемый файл и запустила его, указав аргументы.



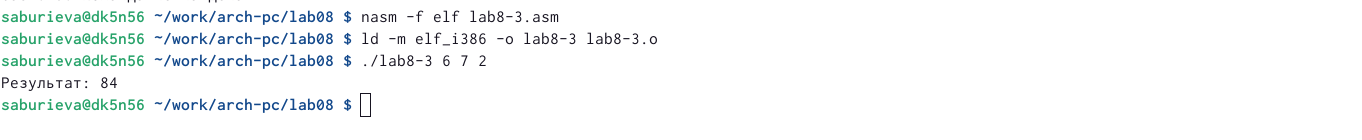
Создание исполняемого файла и его запуск

Изменила текст программы для ля вычисления произведения аргументов командной строки.



Изменение текста программы lab8-3.asm

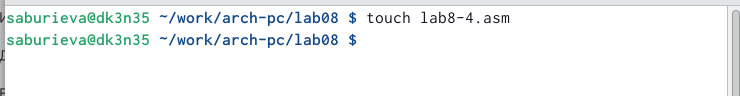
Создала исполняемый файл и запустила его, указав аргументы.



Создание исполняемого файла и его запуск

# 4 Задание для самостоятельной работы

Спрева создала файл lab8-4.asm для написания программы задания самостоятельного выполения.



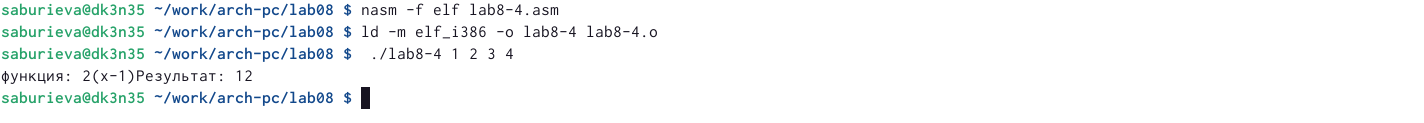
Создание файла lab8-4

Написала программу, которая находит сумму значений функции f(x). Вид функции f(x) выбрала из таблицы вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7(вариант 4).



Написание программы для нахождения суммы значений функции

Создала исполняемый файл и запустила его, указав аргументы.



Создание исполняемого файла и его запуск

# 5 Вывод

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

# Список литературы