Отчет по лабораторной работе №8

Архитектура компьютера

Николенко Анна Николаевна

Содержание

# 1 Цель работы

Цель работы заключается в приобретении навыков написания программ с использованием циклов и обработкой аргументов командной строки.

# 2 Задание

1. Проверка работы программы вывода значений регистра ecx с добавлением различных изменений
2. Проверка работы программы, выводящая на экран аргументы командной строки
3. Проверка работы программы вычисления суммы и произведения аргументов командной строки
4. Задания по самостоятельной работе

# 3 Теоретическое введение

##Организация стека Стек — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO. Стек является частью архитектуры процессора и реализован на аппаратном уровне. Для работы со стеком в процессоре есть специальные регистры (ss, bp, sp) и команды. Основая функция стека - сохранение 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 ре- гистре esp (указатель стека)) и дно (противоположный конец стека). 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).

Добавление элемента в стек: 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.

Извлечение элемента из стека: 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.

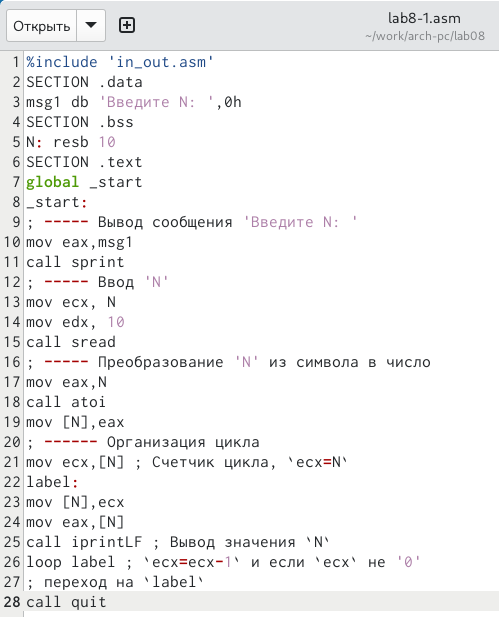
# 4 Выполнение лабораторной работы

Создаю каталог для программам лабораторной работы № 8, перехожу в него и создаю файл lab8-1.asm (рис. [??]).

Создание каталога, переход в этот каталог и создание файла

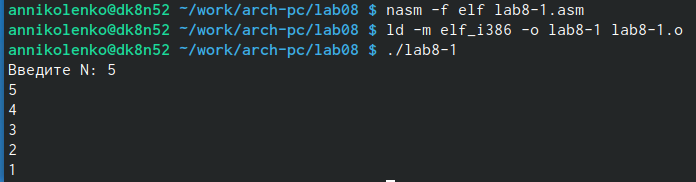
Создание каталога, переход в этот каталог и создание файла

Ввожу программу вывода значений регистра ecx в файл lab8-1.asm (рис. [??]).



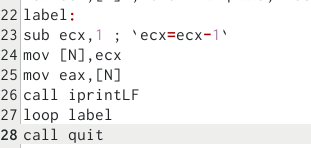
Редактирование файла

Создаю исполняемый файл и запускаю его (рис. [??]).



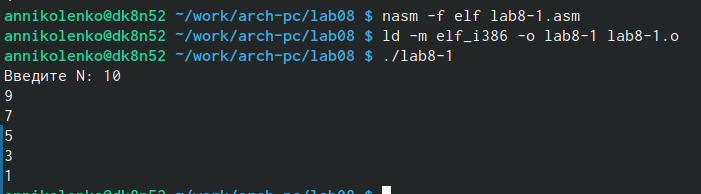
Создание исполняемого файла и его запуск

Изменяю текст программы в файле lab8-1.asm, добавив изменение значение регистра ecx в цикле (рис. [??]).



Редактирование файла

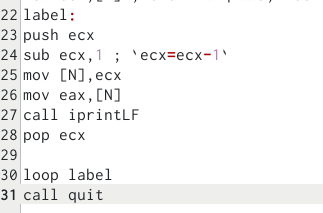
Создаю исполняемый файл и запускаю его (рис. [??]).



Создание исполняемого файла и его запуск

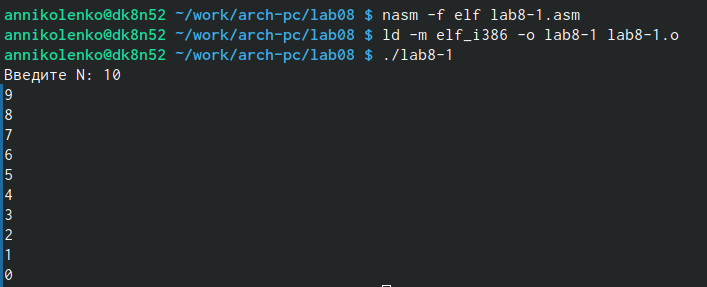
Регистр ecx в цикле принимает нечетные значения. меньшие 10. Число проходов цикла не соответствует значению 𝑁 введенному с клавиатуры. Получаю результат отличный от ожидаемого: получаю N/2 значений

Вношу изменения в файл lab8-1.asm, добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop (рис. [??]).



Редактирование файла

Создаю исполняемый файл и запускаю его (рис. [??]).



Создание исполняемого файла и его запуск

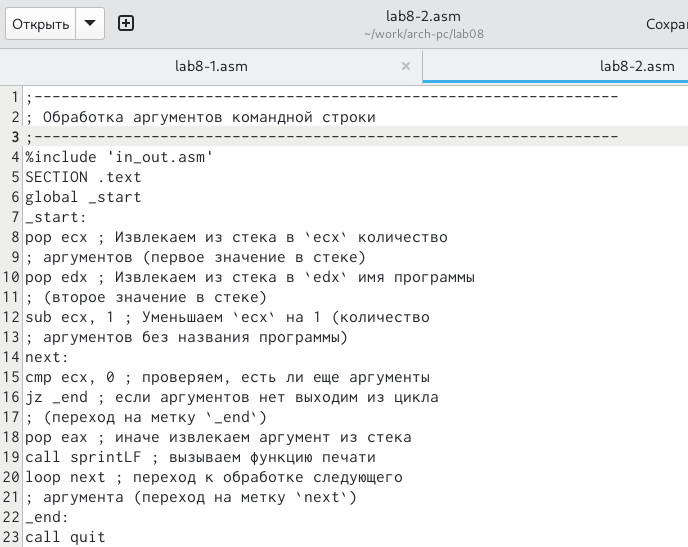
Теперь ввелось N значений.

Создаю файл lab8-2.asm (рис. [??]).

Создание файла

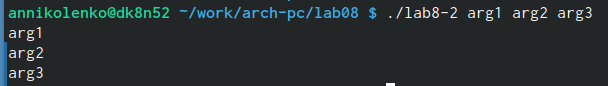
Создание файла

Ввожу программу, выводящая на экран аргументы командной строки в файл lab8-2.asm (рис. [??]).



Редактирование файла

Создаю исполняемый файл и запускаю его (рис. [??]).



Создание исполняемого файла и его запуск

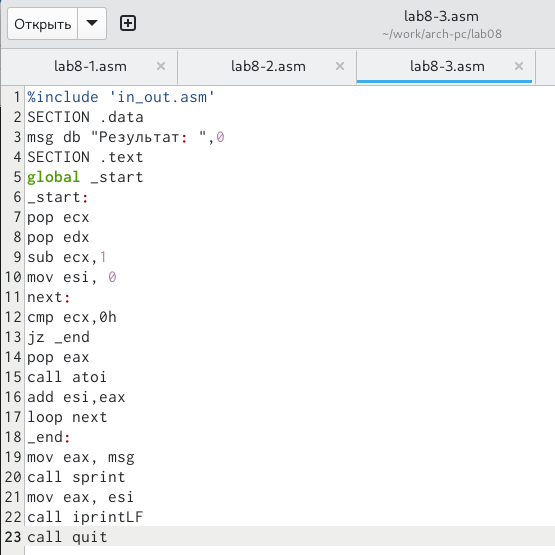
Все аргументы были обработаны программой.

Создаю файл lab8-3.asm (рис. [??]).

Создание файла

Создание файла

Ввожу программу вычисления суммы аргументов командной строки в файл lab8-3.asm (рис. [??]).



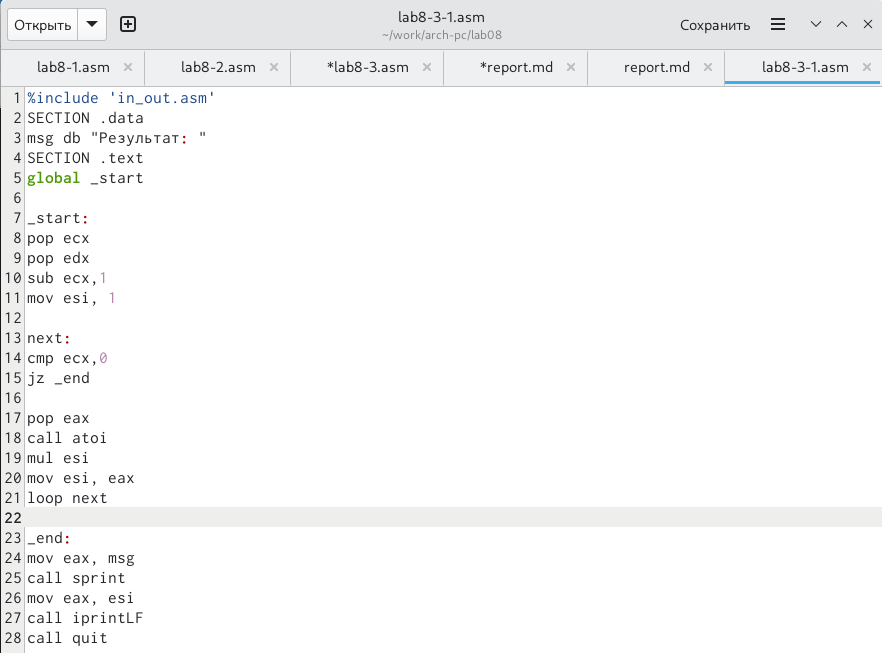
Редактирование файла

Создаю исполняемый файл и запускаю его (рис. [??]).

Создание исполняемого файла и его запуск

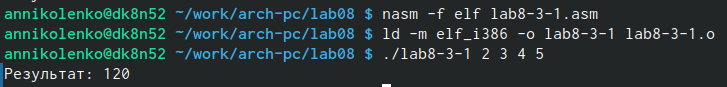
Создание исполняемого файла и его запуск

Вношу изменения в файл lab8-3.asm, чтобы она вычисляла произведение аргументов командной строки (рис. [??]).



Редактирование файла

Создаю исполняемый файл и запускаю его (рис. [??]).

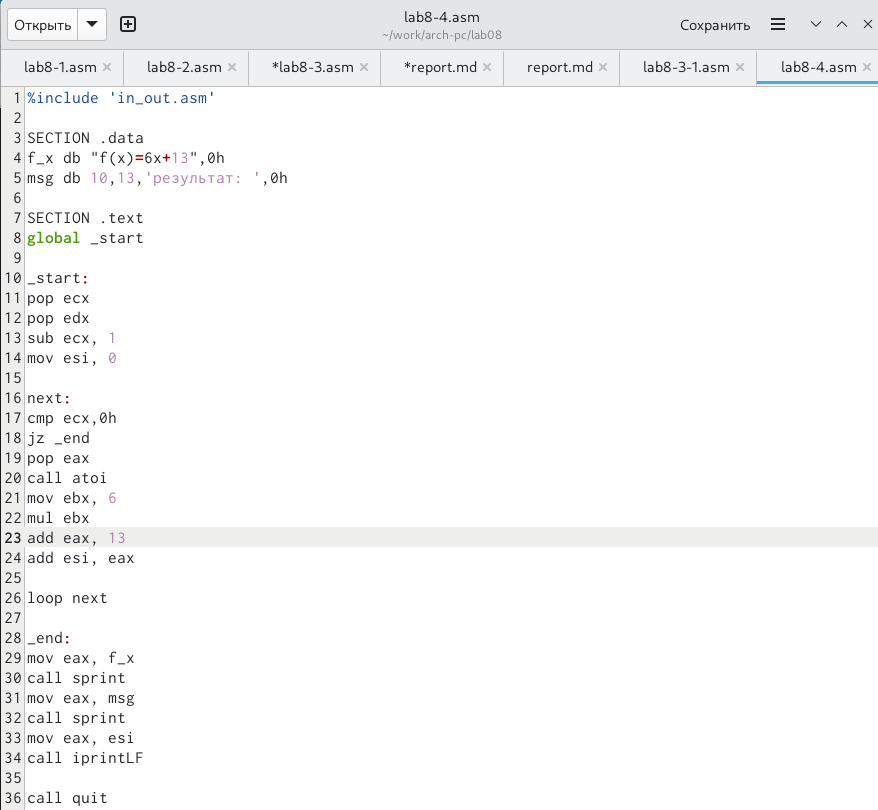


Создание исполняемого файла и его запуск

#Самостоятельная работа

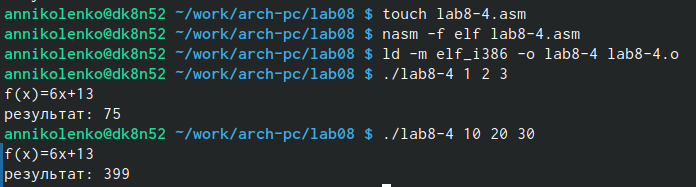
Мой вариант: 15

Пишу программу, которая находит сумму значений функции f(x) (рис. [??]).



Программа суммы значений функции

Создаю исполняемый файл и запускаю его (рис. [??]).



Создание исполняемого файла и его запуск

# 5 Выводы

Были полученынавыки по организации циклов и работе со стеком на языке NASM. <3

# 6 Листинги

%include 'in\_out.asm'  
  
SECTION .data  
f\_x db "f(x)=6x+13",0h  
msg db 10,13,'результат: ',0h  
  
SECTION .text  
global \_start  
  
\_start:  
pop ecx  
pop edx  
sub ecx, 1  
mov esi, 0  
  
next:  
cmp ecx,0h  
jz \_end  
pop eax  
call atoi  
mov ebx, 6   
mul ebx  
add eax, 13  
add esi, eax  
  
loop next  
  
\_end:  
mov eax, f\_x  
call sprint  
mov eax, msg  
call sprint  
mov eax, esi  
call iprintLF  
  
call quit