Лабораторная работа №8

Архитектура компьютера

Овчинников Антон Григорьевич

Содержание

# 1 Цель работы

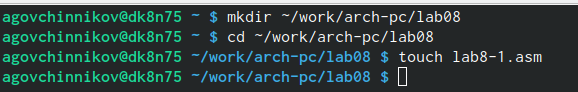
Приобрести навыки написания программ с использованием циклов и обработкой аргументов командной строки

# 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. На рис. 8.1 показана схема организации стека в процессоре. Стек имеет вершину, адрес последнего добавленного элемента, который хранится в ре- гистре esp (указатель стека). Противоположный конец стека называется дном. Значение, помещённое в стек последним, извлекается первым. При помещении значения в стек указа- тель стека уменьшается, а при извлечении — увеличивается. Для стека существует две основные операции: • добавление элемента в вершину стека (push); • извлечение элемента из вершины стека (pop).

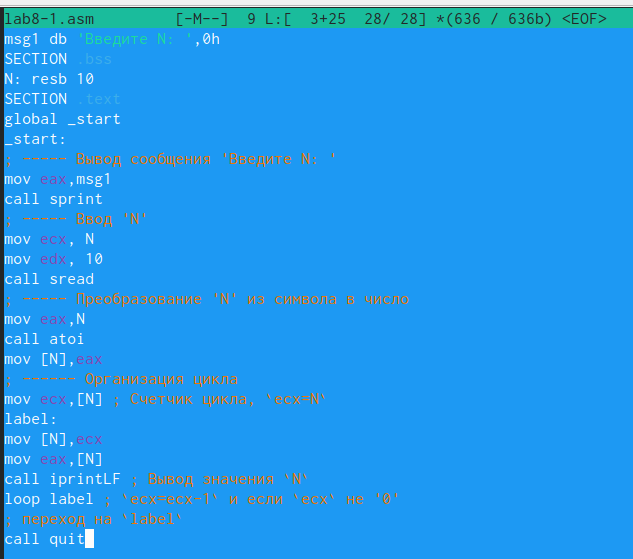
# 3 Выполнение лабораторной работы

Создаю каталог для программ лабораторной работы №8 и создаю файл lab8-1.asm (рис. ??).



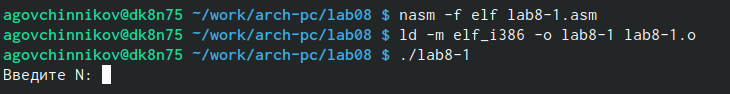
Создание каталога

Ввожу в файл текст из листинга 8.1 (рис. ??).



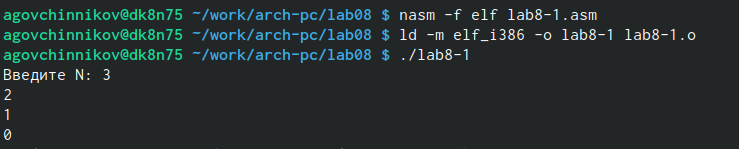
Текст файла

Создаю исполняемый файл и проверяю его работу (рис. ??). В данном случае число проходов цикла не соответствует значению N введенному с клавиатуры.



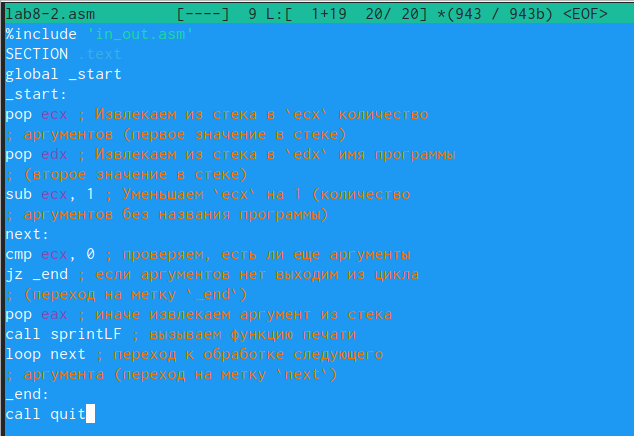
Вывод программы

Меняю текст листинга файла и проверяю рузльтат программы (рис. ??). В данном случае число проходов цикла соответствует значению N введенному с клавиатуры.



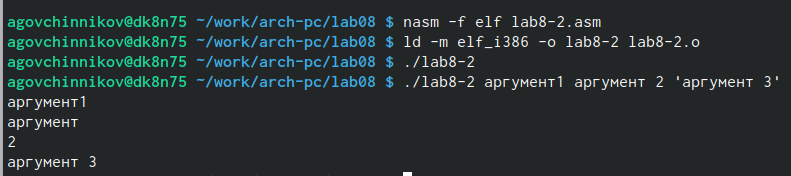
Результат изменненого файла

Создаю файл lab8-2.asm и ввожу туда текст из листинга 8.2 (рис. ??).



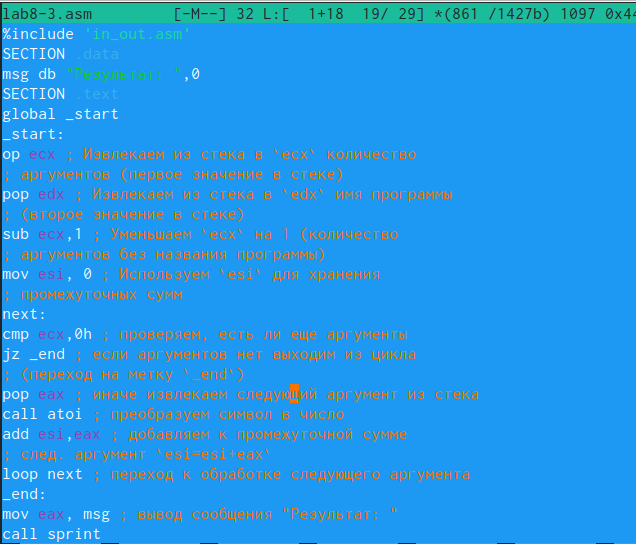
Текст файла

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



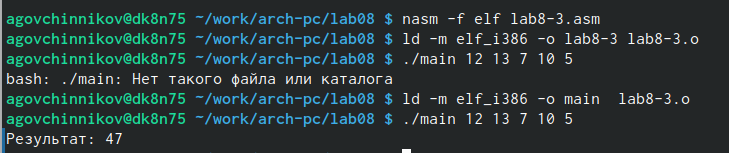
Вывод программы

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



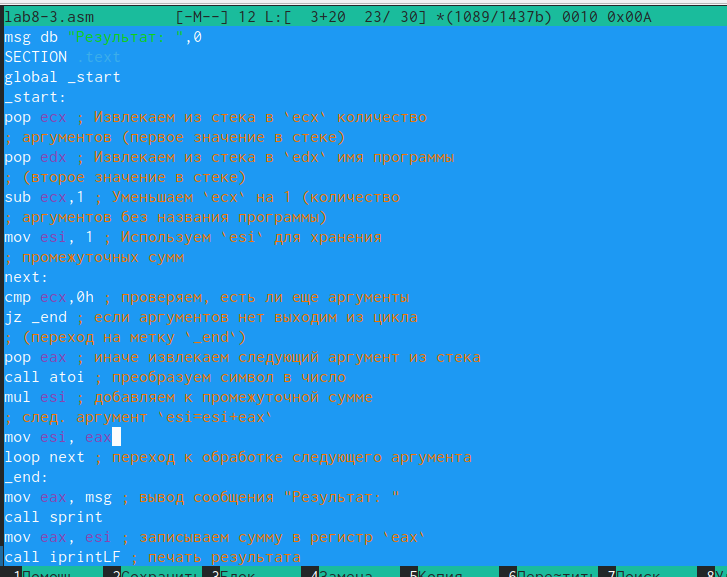
Файл lab8-3.asm

Создаю исполняемый файл и запускаю его, при этом указывая аргументы (рис. ??)



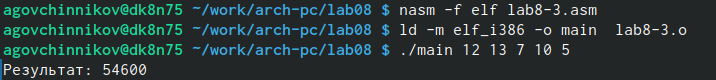
Вывод программы

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



Изменненый текст

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

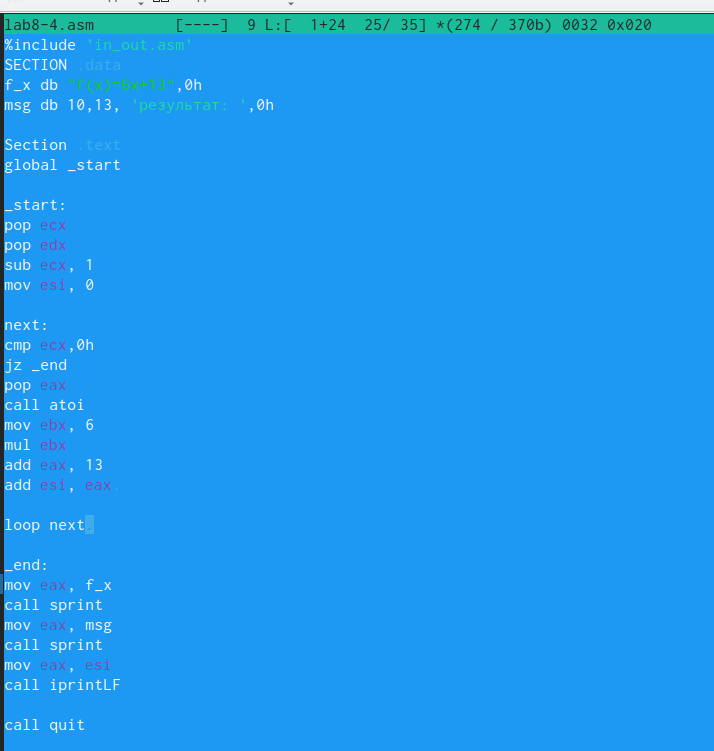


Вывод программы

# 4 Задание для самостоятельной работы

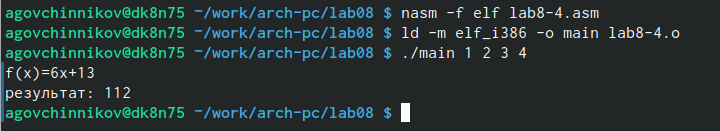
Вариант 15

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



Текст программы

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



Вывод программы

# 5 Листинг

%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

# 6 Выводы

Я приобрел навыки написания программ с использованием циклов и обраткой аргументов командной строки