Отчёт по лабораторной работе №8

Дисциплина: Архитектура компьютера

Мутале Чали

Содержание

[1 Цель работы 1](#_Toc183816385)

[2 Задание 1](#_Toc183816386)

[3 Выполнение лабораторной работы 1](#_Toc183816387)

[4 Выполнение самостоятельной работы 9](#_Toc183816388)

[5 Выводы 11](#_Toc183816389)

[6 Список литературы 11](#_Toc183816390)

# 1 Цель работы

Цель лабораторной 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 Выполнение лабораторной работы

**1. Реализация циклов в NASM**

Создаю каталог для программам лабораторной работы № 8, перехожу в него и создаю файл lab8-1.asm:

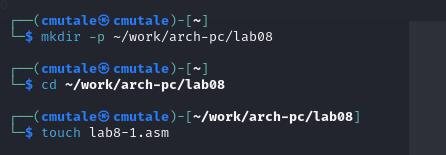


Рис 1

В файле lab8-1.asm вставляю код программы, которая показывает, как инструкция loop использует регистр ecx в качестве счетчика и на каждом шаге уменьшает его значение на единицу:

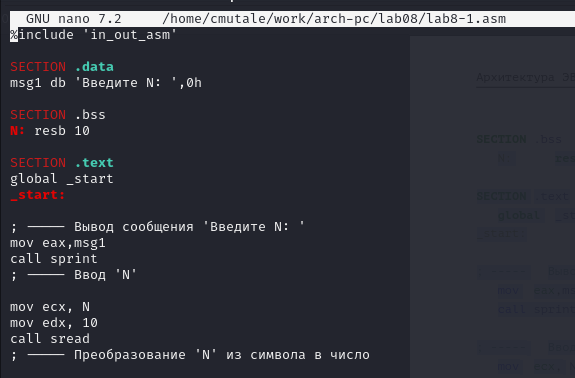


Рис 2

Создаю исполняемый файл:



Рис 3

При запуске, программа выводит значение регистра ecx:

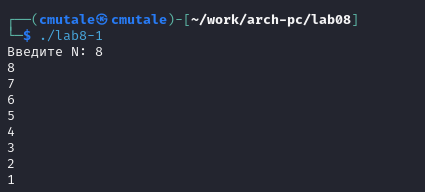


Рис 4

Изменяю текст программы, чтобы она показала, что использование регистра ecx в теле цилка loop может привести к некорректной работе программы:

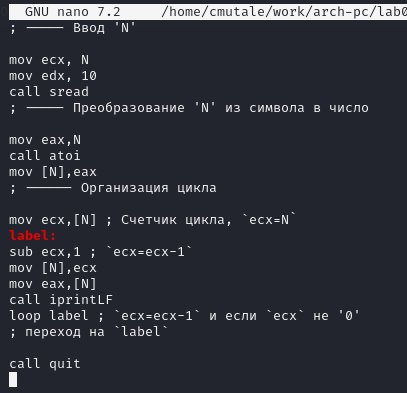


Рис 5

Создаю исполняемый файл:



Рис 6

При запуске, программа выводит бесконечное значение, которое не соответсвует значению N введенному с клавиатуры:

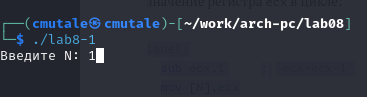


Рис 7

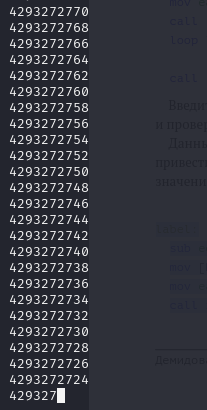


Рис 8

Для использования регистра ecx в цикле и сохранения корректности работы программы можно использовать стек. Изменяю текст программы добавив команды push и pop:

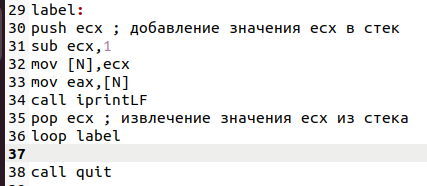


Рис 9

Создаю исполняемый файл:

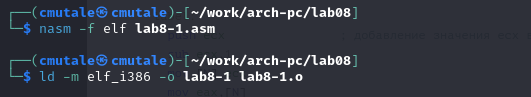


Рис 10

При запуске, программа выводит значение, которое соответсвует значению N введенному с клавиатуры:

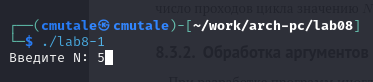


Рис 11

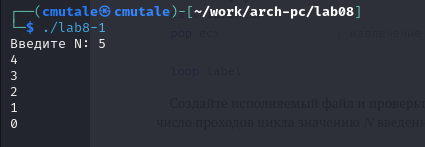


Рис 12

**2. Обработка аргументов командной строки**

Создаю файл lab8-2.asm:

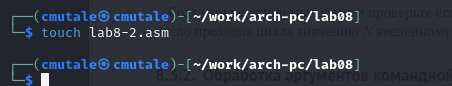


Рис 13

Ввожу в него текст программы, которая выводит на экран аргументы командной строки:

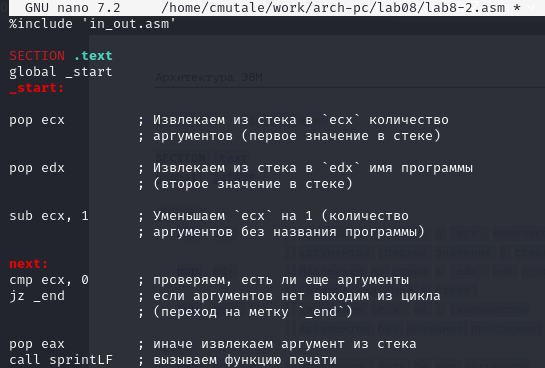


Рис 14

Создаю исполняемый файл и запускаю его:

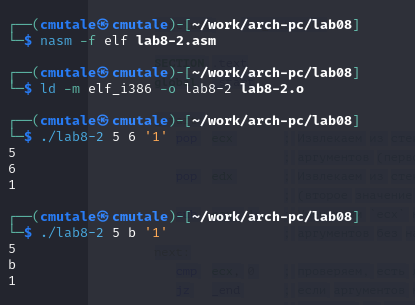


Рис 15

Создаю файл lab8-3.asm:



Рис 16

Ввожу в него текст программы, которая выводит на экран сумму аргументов:

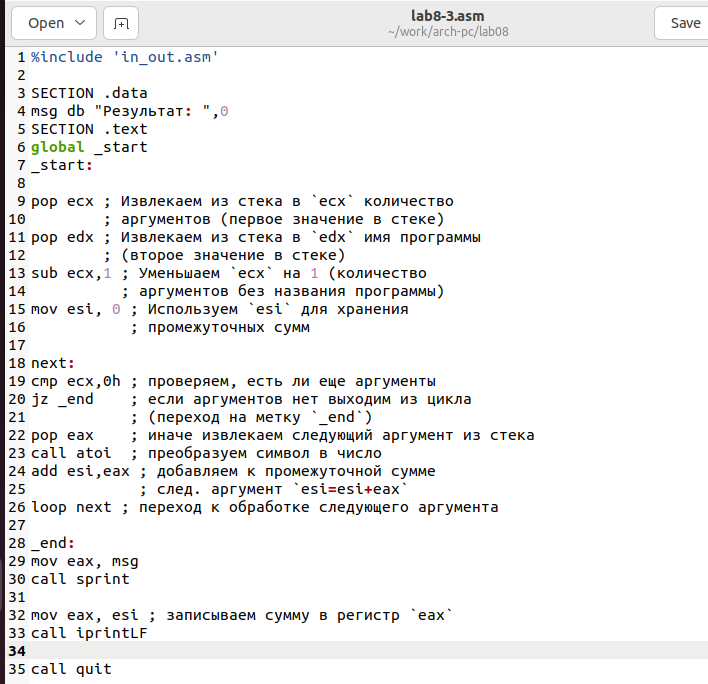


Рис 17

Создаю исполняемый файл и запускаю его:

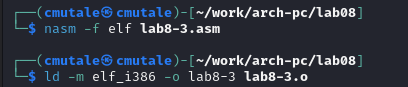


Рис 18



Рис 19

Изменяю текст программы, чтобы она выводила произведение аргументов:



Рис 20

Создаю исполняемый файл и запускаю его:

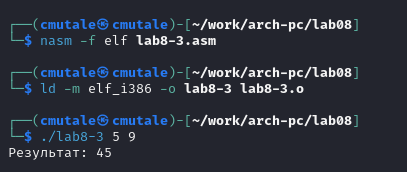


Рис 21

# 4 Выполнение самостоятельной работы

Создаю файл task8.asm:



Рис 22

В него пишу программу, которая находит сумму значений функции f(x) = 5(2 + x)$ для некоторых значении x (вариант 10):

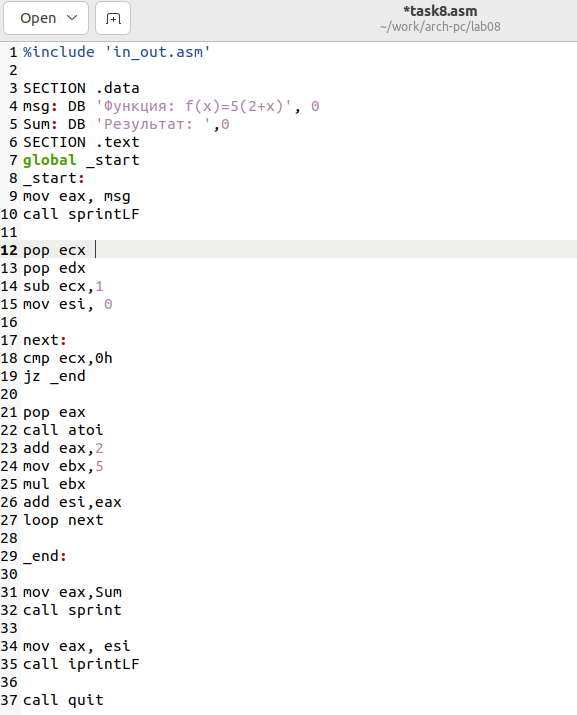


Рис 23

Код программы:

%include 'in\_out.asm'  
  
SECTION .data  
msg: DB 'Функция: f(x)=5(2+x)', 0  
Sum: DB 'Результат: ',0  
SECTION .text  
global \_start  
\_start:  
mov eax, msg  
call sprintLF  
  
pop ecx   
pop edx   
sub ecx,1   
mov esi, 0  
   
next:  
cmp ecx,0h   
jz \_end   
   
pop eax   
call atoi   
add eax,2  
mov ebx,5  
mul ebx  
add esi,eax  
loop next   
  
\_end:  
  
mov eax,Sum  
call sprint  
  
mov eax, esi   
call iprintLF  
   
call quit

Создаю исполняемый файл и запускаю его. Программа выводит сумму f(1)+f(2)+f(1):

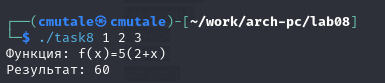
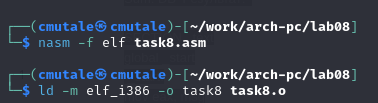


Рис 24

# 5 Выводы

При выполнение данной работы я освоила использование циклов и обработку аргументов командной строки.

# 6 Список литературы

[Архитектура ЭВМ](https://esystem.rudn.ru/pluginfile.php/2089095/mod_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%968.%20%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%20%D1%86%D0%B8%D0%BA%D0%BB%D0%B0.%20%D0%9E%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0%20%D0%B0%D1%80%D0%B3%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2%20%D0%BA%D0%BE%D0%BC%D0%B0%D0%BD%D0%B4%D0%BD%D0%BE%D0%B9%20%D1%81%D1%82%D1%80%D0%BE%D0%BA%D0%B8..pdf)