**Отчет по лабораторной работе №8**

*дисциплина: Архитектура компьютера*

Сергеев Даниил Олегович

Содержание

# 1 Цель лабораторной работы

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

# 2 Ход выполнения лабораторной работы

1. Создаем каталог для программ лабораторной работы №8. Переходим в него и создаем файл lab8-1.asm.

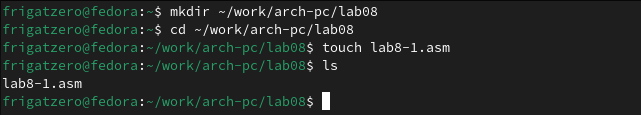


Рис. 1: Создание lab8-1.asm

1. Вводим в файл lab8-1.asm код из листинга 8.1, создаем исполняемый файл и проверяем его работу.

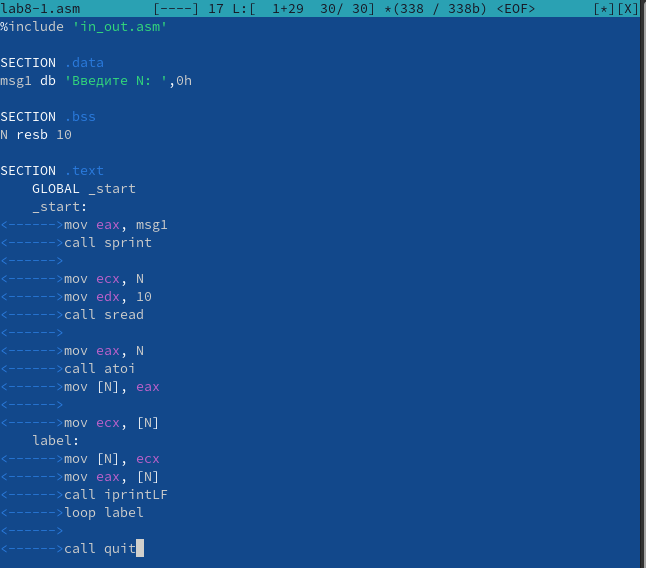


Рис. 2: Код файла lab8-1.asm

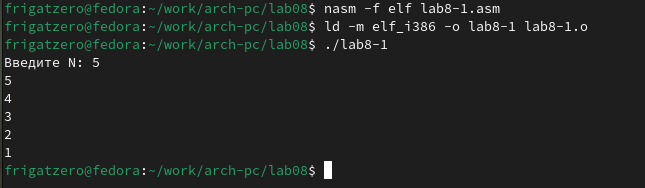


Рис. 3: Проверка работы lab8-1

1. Изменяем текст программы так, чтобы значение регистра ecx изменялось в цикле.



Рис. 4: Изменение lab8-1.asm

1. Создадим исполняемый файл измененной программы и проверим его работу. При нечетных N, значение ecx доходит до 0 и из него вычитается 1, из-за чего все биты ecx становятся равными 1 и счетчик начинает идти с максимального значения ecx. При четных N, значение ecx доходит до 1 и, после вычитания из него 1, цикл заканчивается, также число проходов цикла становится меньше значения N введенного с клавиатуры.

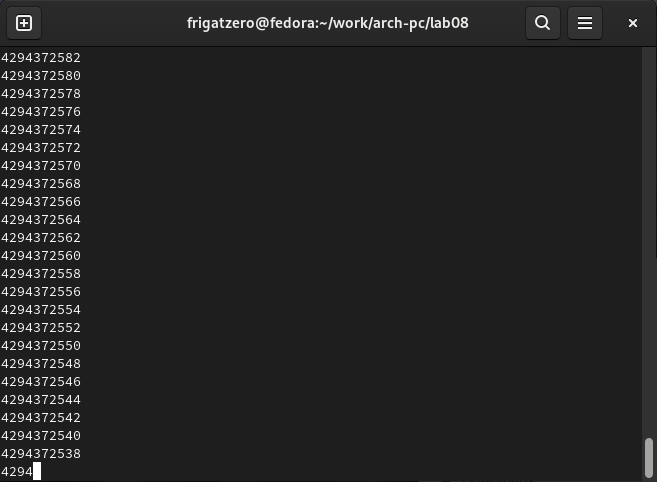


Рис. 5: Вывод программы с нечетными N

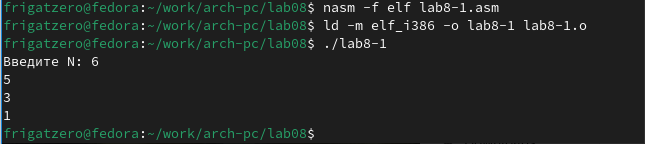


Рис. 6: Вывод программы с четными N

1. Вписываем в программу команды для добавления в стек и извлечения из стека.



Рис. 7: Изменение lab8-1.asm со стеком

1. Создадим исполняемый файл и проверим его работу. В данном случае число проходов цикла равно введенному при любых значениях N, однако каждый элемент меньше на единицу.

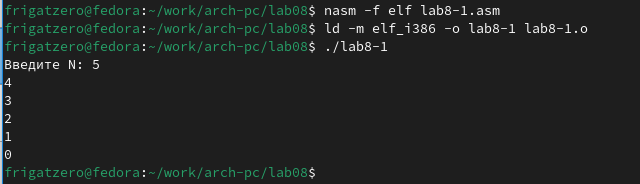


Рис. 8: Вывод программы со стеком

1. Создадим файл lab8-2.asm и введем в него текст программы из листинга 8.2. Создадим исполняемый файл и запустим его с заданными аргументами. В итоге были обработаны аргументы: аргумент1, аргумент, 2, ‘аргумент 3’.

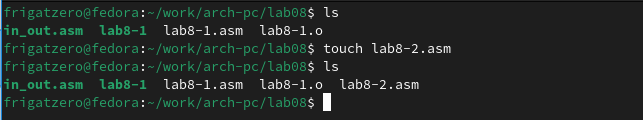


Рис. 9: Создание lab8-2.asm

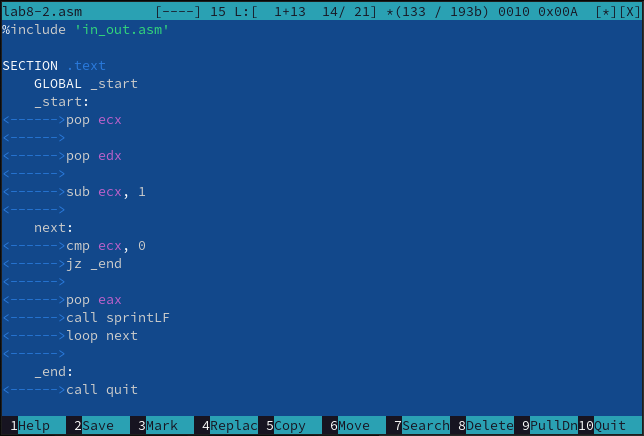


Рис. 10: Код программы lab8-2 из листинга

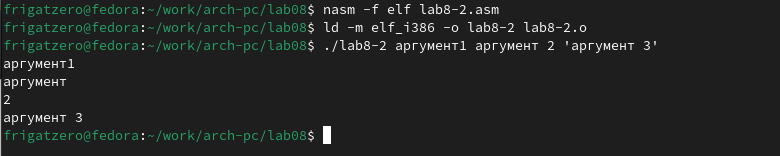


Рис. 11: Проверка файла lab8-2

1. Создадим файл lab8-3.asm и введем в него текст листинга 8.3. Создадим исполняемый файл и проверим его работу, указав аргументы.

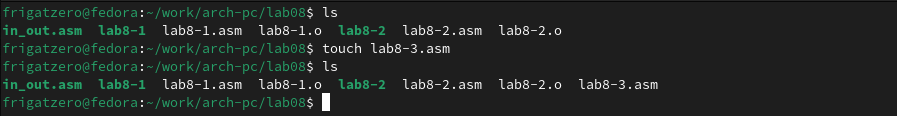


Рис. 12: Создание lab8-3.asm

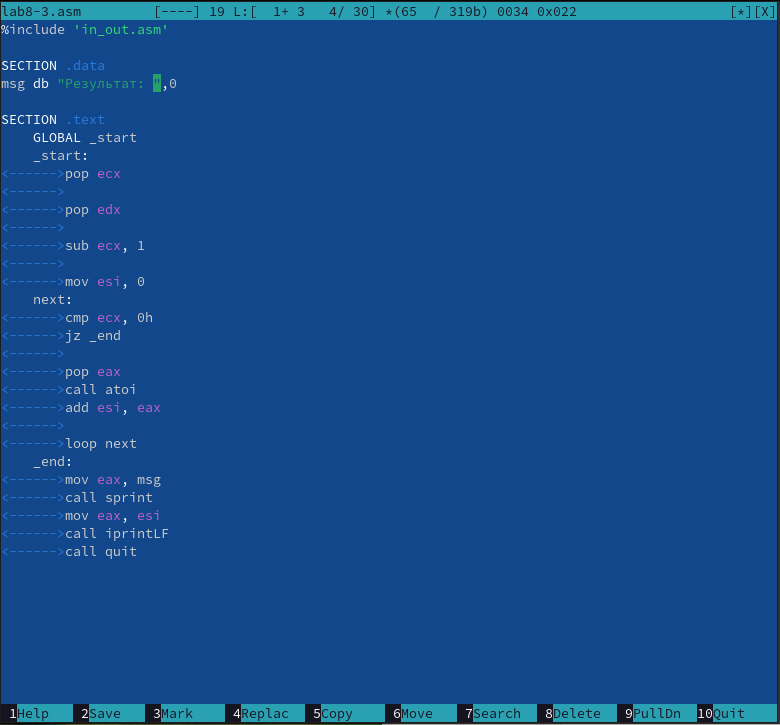


Рис. 13: Код программы lab8-3 из листинга

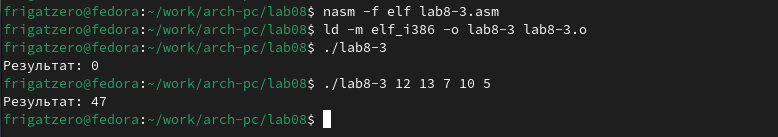


Рис. 14: Проверка файла lab8-3

1. Изменим текст программы для вычисления произведения аргументов командной строки.

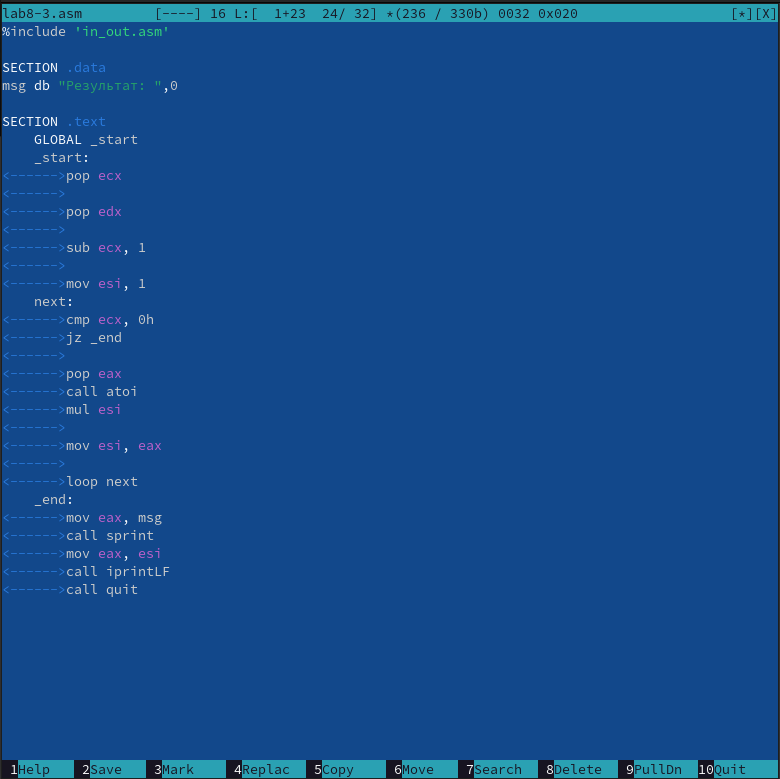


Рис. 15: Изменение кода программы lab8-3.asm

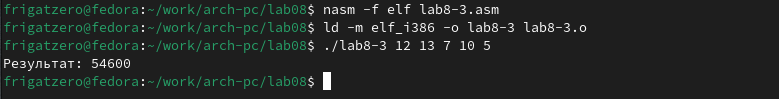


Рис. 16: Проверка работы измененного файла lab8-3

# 3 Ход выполнения заданий для самостоятельной работы

1. Напишем программу для нахождения суммы значений функции для введенных в качестве аргумента программы значений x. Выберем значение функции из №18, табл. 7.5. Создадим исполняемый файл и проверим его работу

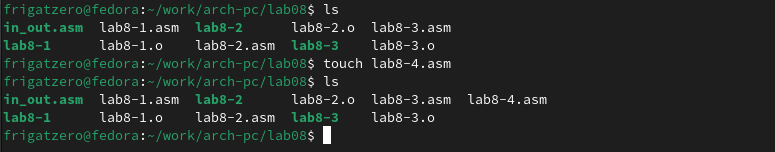


Рис. 17: Создаем файл lab8-4.asm

**Листинг 3.1** Программа для нахождения суммы значений функции для введенных x.

%include 'in\_out.asm'  
  
SECTION .data  
fun db 'Функция: f(x) = 17 + 5x',0  
res db 'Результат: ',0  
  
SECTION .bss  
x resb 10  
  
SECTION .text  
 GLOBAL \_start  
 \_start:  
 ;=[Выводим сообщение с функцией]  
 mov eax, fun  
 call sprintLF  
 ;  
   
 ;=[Извлекаем количество аргументов и имя программы с стека]  
 pop ecx ; Количество аргументов в <ecx>  
 pop edx ; Имя программы в <edx>  
 sub ecx, 1 ; <ecx - 1> => кол-во аргументов без названия программы  
 ;  
 main:  
 ;=[Сравниваем ecx с нулем]  
 cmp ecx, 0 ; Если <ecx = 0>  
 je \_exit ; Завершаем программу  
 ;  
   
 ;=[Забираем n-ый аргумент с стека]  
 pop eax  
 call atoi ; конвертируем из кода символа в число  
 ;  
   
 ;=[Находим значение функции]  
 mov edi, 5 ; <edi = 5>  
 mul edi ; <eax = 5x>  
 add eax, 17 ; <eax = 17+5x>  
 mov edi, eax ; <edi = eax>  
 add [x], edi ; <x = x + edi>  
 ;  
   
 loop main ; <ecx - 1>  
 \_exit:  
 ;=[Выводим сообщение 'Результат: <x>']  
 mov eax, res  
 call sprint  
 mov eax, [x]  
 call iprintLF  
 ;  
 call quit

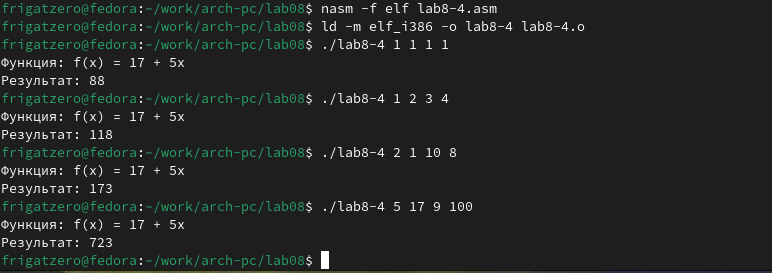


Рис. 18: Результат работы программы lab8-4

# 4 Вывод

После выполнения заданий лабораторной работы и заданий для самостоятельной работы я приобрел навыки написания программ с использованием циклов и обработкой аргументов командой строки, научился работать с структурой данных стек и использовать регистр ecx для работы с циклами.