Лабораторная работа №9.

Программирование цикла. Обработка аргументов командной строки.

Боровиков Даниил Александрович

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Создадим каталог для программам лабораторной работы № 9, перейдем в него и создадим файл lab9-1.asm(рис. 1)

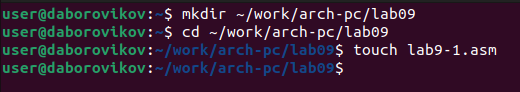


Рис. 1: Создание файла lab9-1.asm в соответствующем каталоге

Введем в файл lab9-1.asm текст программы из листинга 9.1.(рис. 2)

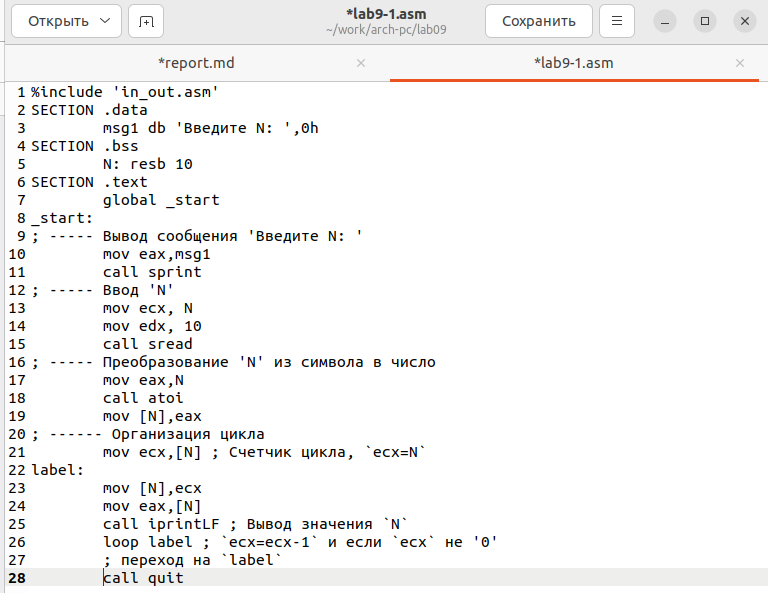


Рис. 2: Текст программы из листинга 9.1.

Создадим исполняемый файл и запустим его.(рис. 3)

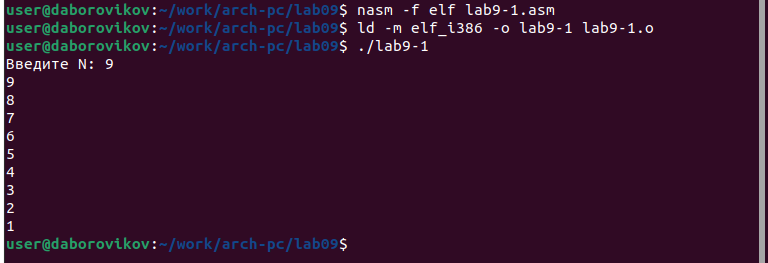


Рис. 3: Запуск исполняемого файла lab9-1.asm

Число проходов цикла соответствует введенному значению N.

Далее изменим текст программы добавив изменение значение регистра ecx в цикле(рис. 4)

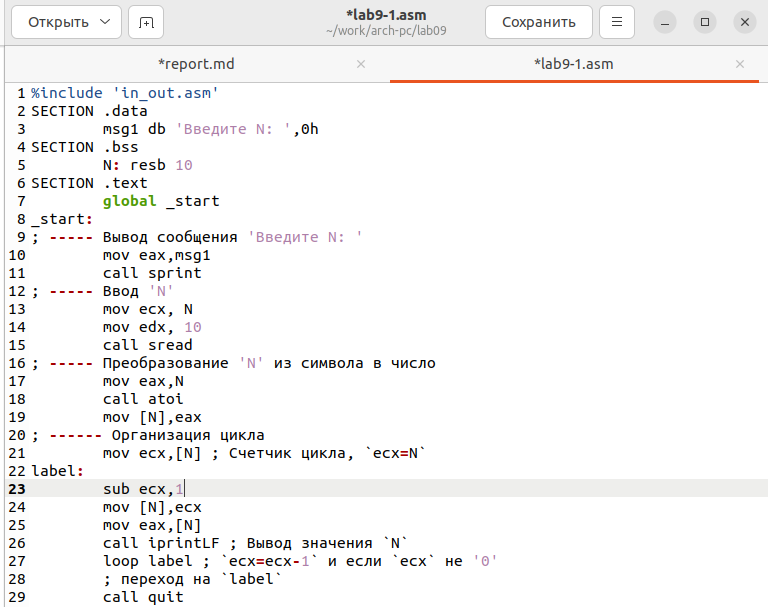


Рис. 4: Текст измененной программы

Создадим исполняемый файл исправленного текста программы lab9-1.asm и запустите его.(рис. 5)

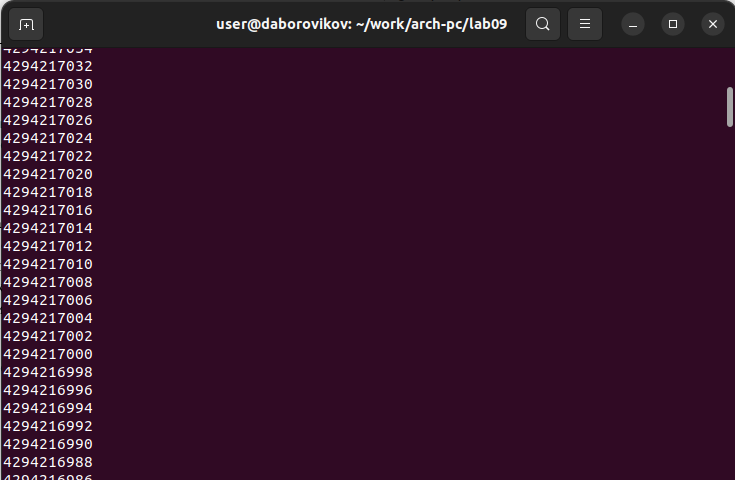


Рис. 5: Запуск исправленного исполняемого файла lab8-1.asm

Программа выводит некорректные значения из-за использования регистра ecx в теле цикла loop. Значения ecx перезаписываются в eax и выводятся на экран. Число проходов не соответствует значению N, как это было в первоначальной версии программы.

Внесем изменения в текст программы добавив команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop:(рис. 6)

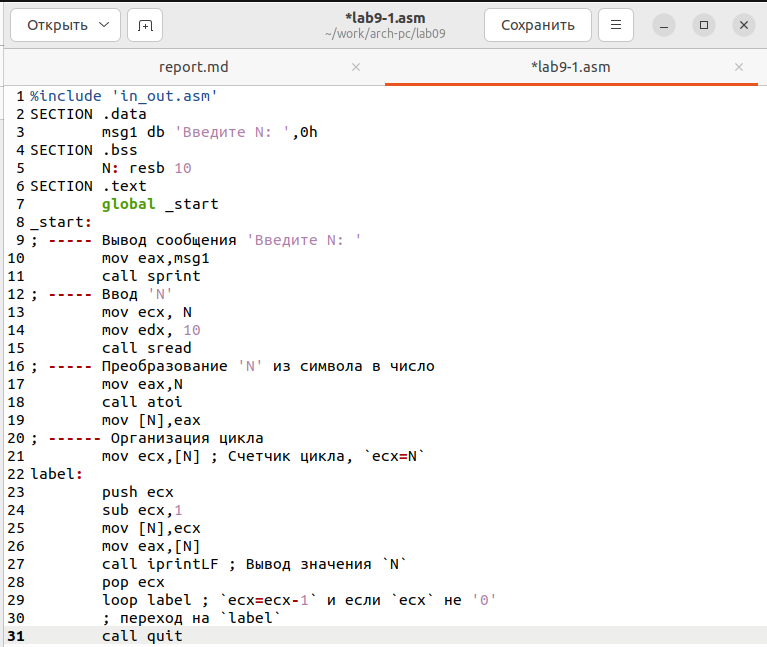


Рис. 6: Листинг программы с сохранением значения счетсика в цикле

Создадим исполняемый файл и запустим его(рис. 7)

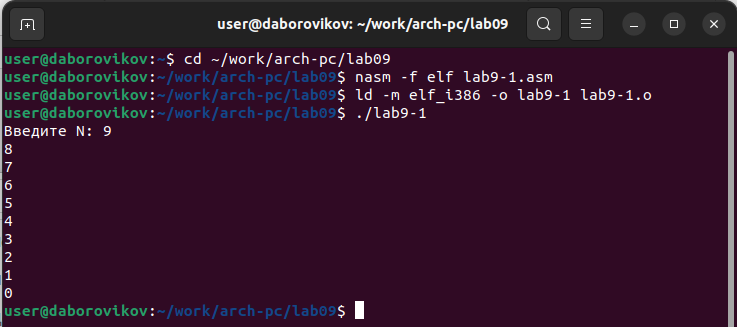


Рис. 7: Запуск измененного исполняемого файла lab9-1.asm

В данном случае число проходов цикла соответствует значению, введенному с клавиатуры

Создадим файл lab9-2.asm в каталоге ~/work/arch-pc/lab09. Внимательно изучим текст программы из листинга 9.2 и введем в lab9-2.asm.(рис. 8)

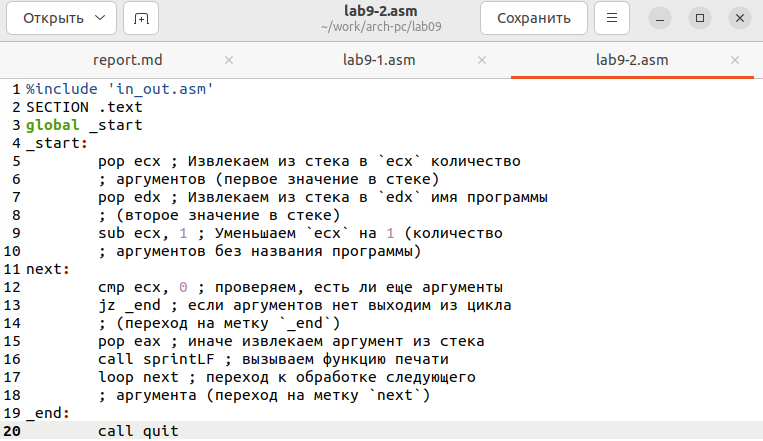


Рис. 8: Листинга 9.2

Создадим измененный исполняемый файл и запустим его(рис. 9)

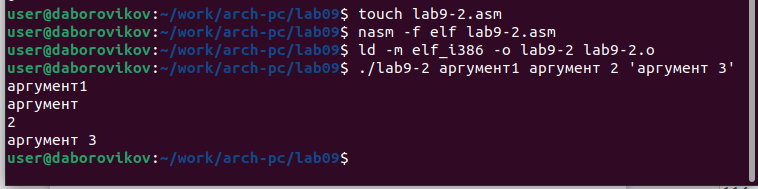


Рис. 9: Запуск исполняемого файла lab9-2.asm

Программой было обработано 4 аргумента.

Создадим файл lab9-3.asm в каталоге ~/work/arch-pc/lab09. Внимательно изучим текст программы из листинга 9.3 и введем в lab9-3.asm.(рис. 10)

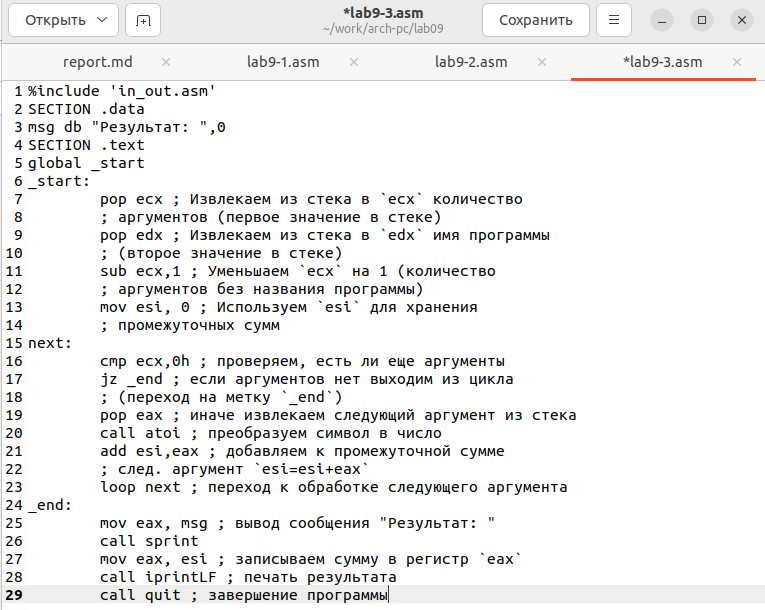


Рис. 10: Листинг 9.3

Создадим исполняем файл и запустим его, указав аргументы(рис. 11)

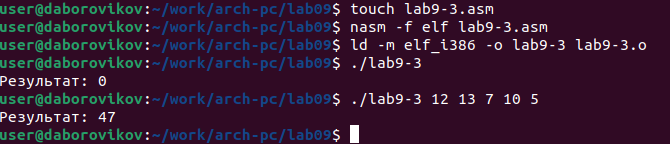


Рис. 11: Запуск исполняемого файла lab9-3.asm

Изменим файл lab9-3.asm для вычисления произведения аргументов командной строки (рис. 12)

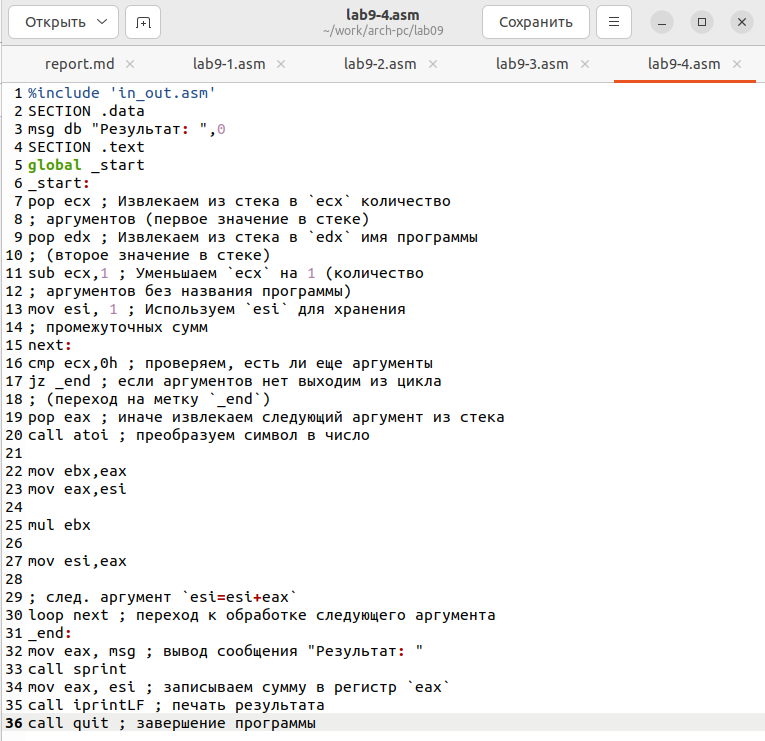


Рис. 12: Текст программы для вычисления произведения командной строки

Создадим исполняем файл и запустим его, указав аргументы(рис. 13)

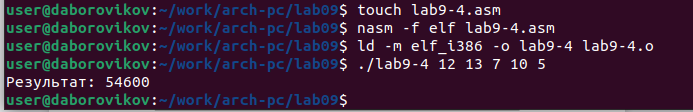


Рис. 13: Запуск исполняемого файла lab9-4.asm

Листинг программы:

%include ‘in\_out.asm’

SECTION .data

msg db “Результат:”,0

SECTION .text

global \_start

\_start:

pop ecx ; Извлекаем из стека в ecx количество

; аргументов (первое значение в стеке)

pop edx ; Извлекаем из стека в edx имя программы

; (второе значение в стеке)

sub ecx,1 ; Уменьшаем ecx на 1 (количество

; аргументов без названия программы)

mov esi, 1 ; Используем esi для хранения

; промежуточных сумм

next:

cmp ecx,0h ; проверяем, есть ли еще аргументы

jz \_end ; если аргументов нет выходим из цикла

; (переход на метку \_end)

pop eax ; иначе извлекаем следующий аргумент из стека

call atoi ; преобразуем символ в число

mov ebx,eax

mov eax,esi

mul ebx

mov esi,eax

; след. аргумент esi=esi+eax

loop next ; переход к обработке следующего аргумента

\_end:

mov eax, msg ; вывод сообщения “Результат:”

call sprint

mov eax, esi ; записываем сумму в регистр eax

call iprintLF ; печать результата

call quit ; завершение программы

# 3 Самостоятельная работа

Мой вариант номер 7

Напишем программу, которая находит сумму значений функции f(x) для x = x1 , x2 , …, xn , т.е. программа должна выводить значение f(x1) + f(x2)+…+f(xn). Значения xi передаются как аргументы. Вид функции f(x) выберем из таблицы 9.1 вариантов заданий в соответствии с вариантом номер 7, по- лученным при выполнении лабораторной работы № 7. Создадим исполня- емый файл и проверьте его работу на нескольких наборах x = x1 , x2 , …, xn .(рис. 14)

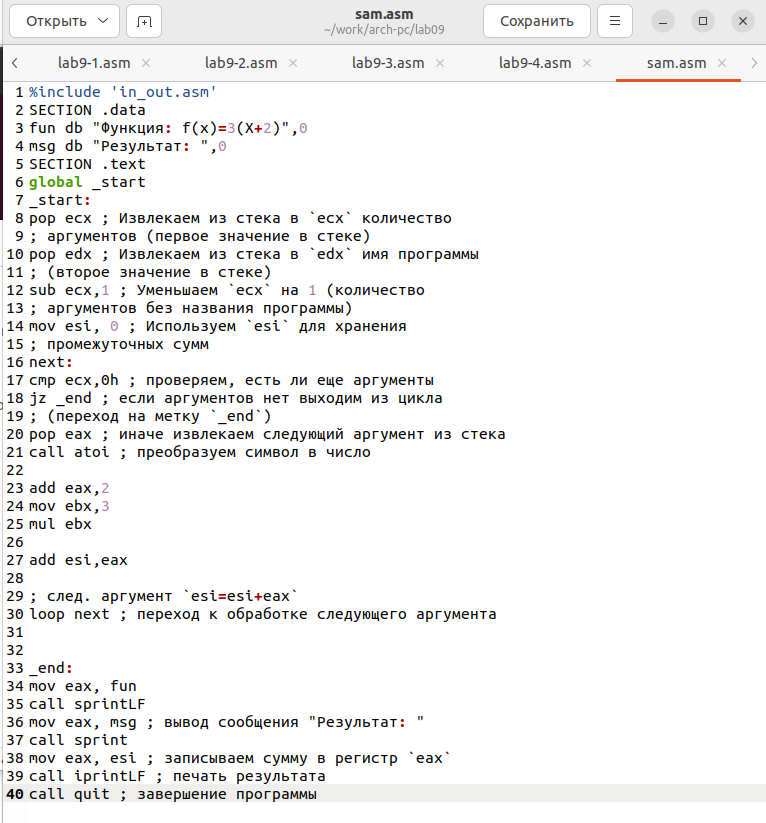


Рис. 14: Текст программы

Создадим исполняем файл и запустим его, указав аргументы(рис. 15)

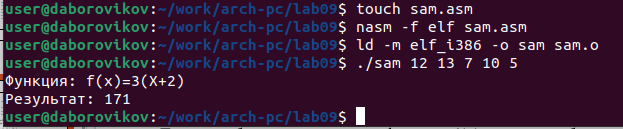


Рис. 15: Запуск исполняемого файла sam.asm

Листинг программы:

%include ‘in\_out.asm’

SECTION .data

fun db “Функция: f(x)=3(X+2)”,0

msg db “Результат:”,0

SECTION .text

global \_start

\_start:

pop ecx ; Извлекаем из стека в ecx количество

; аргументов (первое значение в стеке)

pop edx ; Извлекаем из стека в edx имя программы

; (второе значение в стеке)

sub ecx,1 ; Уменьшаем ecx на 1 (количество

; аргументов без названия программы)

mov esi, 0 ; Используем esi для хранения

; промежуточных сумм

next:

cmp ecx,0h ; проверяем, есть ли еще аргументы

jz \_end ; если аргументов нет выходим из цикла

; (переход на метку \_end)

pop eax ; иначе извлекаем следующий аргумент из стека

call atoi ; преобразуем символ в число

add eax,2

mov ebx,3

mul ebx

add esi,eax

; след. аргумент esi=esi+eax

loop next ; переход к обработке следующего аргумента

\_end:

mov eax, fun

call sprintLF

mov eax, msg ; вывод сообщения “Результат:”

call sprint

mov eax, esi ; записываем сумму в регистр eax

call iprintLF ; печать результата

call quit ; завершение программы

# 4 Выводы

В ходе лабораторной работы мы приобрели наыки написания программ с использованием циклов и обработкой аргументов командной строки.

https://github.com/daBorovikov/study\_2022-2023\_arh-pc-