Отчёт по лабораторной работе №8

Дисциплина: Архитектура компьютера

Калашникова Ольга Сергеевна

Содержание

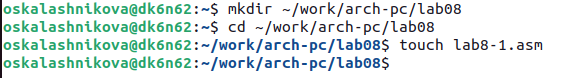
# 1 Цель работы

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

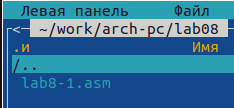
# 2 Выполнение лабораторной работы

## 2.1 Реализация циклов в NASM

Создаём каталог для программам лабораторной работы № 8 (при помощи команды mkdir ~/work/arch-pc/lab08), переходим в него (при помощи команды cd ~/work/arch-pc/lab08) и создаём файл lab8-1.asm (при помощи команды touch lab8-1.asm) (рис. ??),(рис. ??)

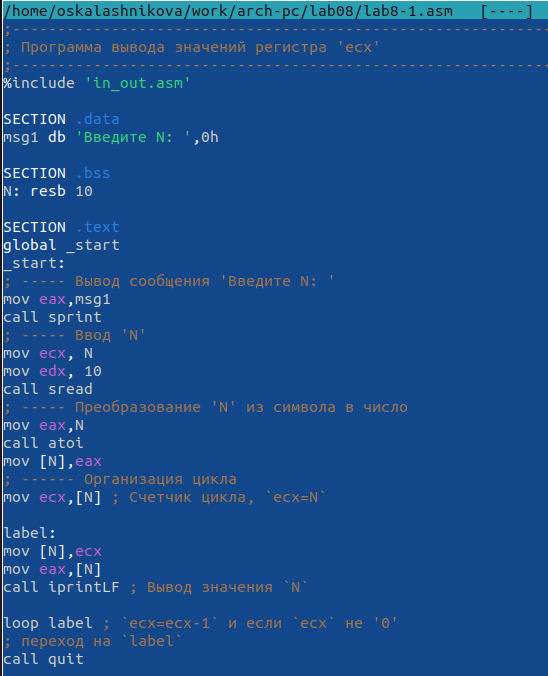


Создание папки, её открытие и создание файла

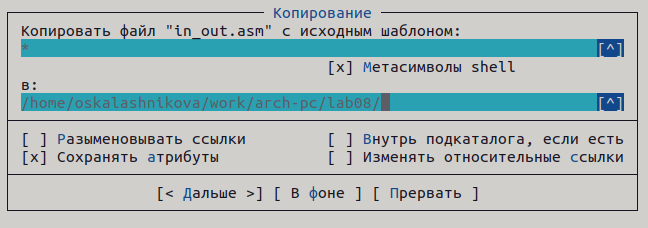


Проверка наличия папки

Введём в файл lab8-1.asm текст программы из листинга 8.1., для корректной работы нужно переместить файл “in\_out.asm” в тот же каталог, где лежит и файл с текстом программы.(рис. ??),(рис. ??)

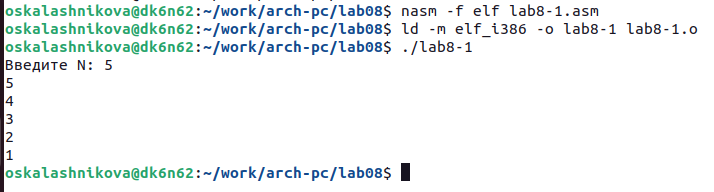


Текст программы



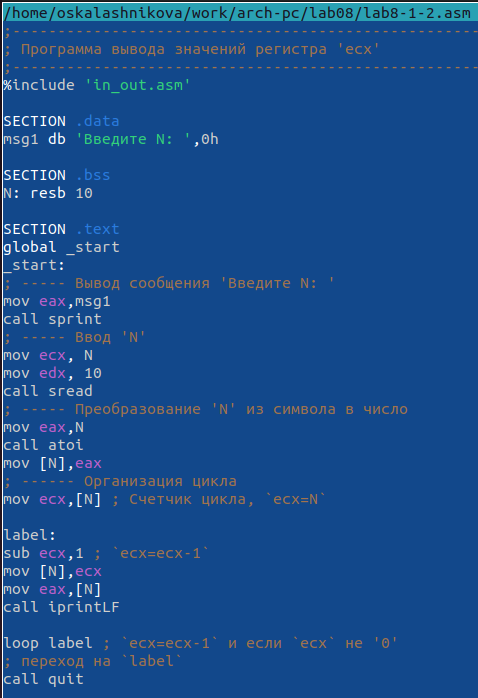
Перемещение файла “in\_out.asm”

Создаем исполняемый файл и запускаем его (компиляция: nasm -f elf lab8-1.asm , ld -m elf\_i386 -o lab8-1 lab8-1.o , запуск: ./lab8-1) (рис. ??)



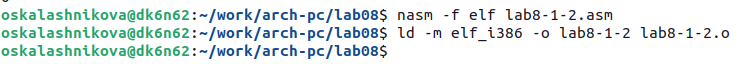
Создание и запуск исполняемого файла

Меняем текст программы (Используем регистр ecx в теле цилка loop) (рис. ??)



Изменённый текст программы

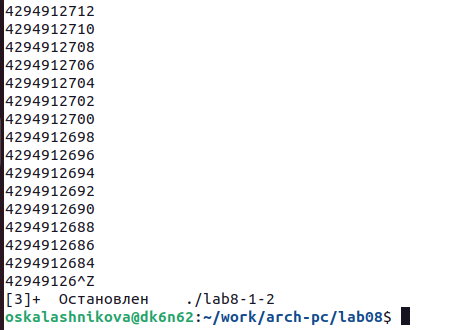
Создаём исполняемый файл и запускаем его (компиляция: nasm -f elf lab8-1-2.asm , ld -m elf\_i386 -o lab8-1-2 lab8-1-2.o , запуск: ./lab8-1-2) (рис. ??), (рис. ??), (рис. ??)



Создание исполняемого файла

Запуск исполняемого файла 1

Запуск исполняемого файла 1



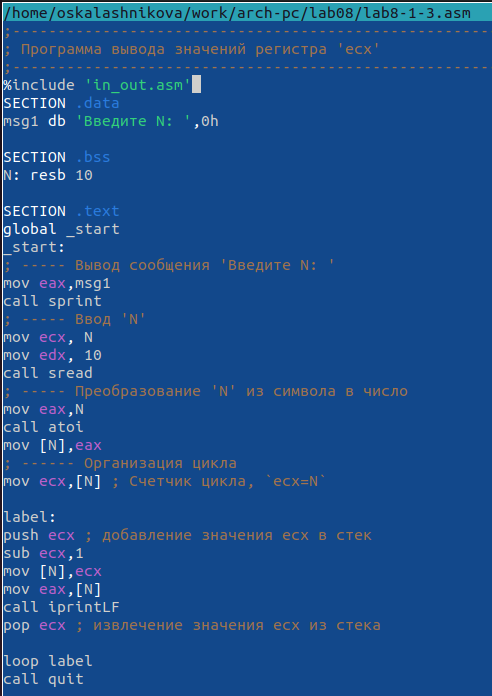
Запуск исполняемого файла 2

Использование регистра ecx в теле цилка loop может привести к некорректной работе программы, что мы и видим в данном примере.

Вопросы:

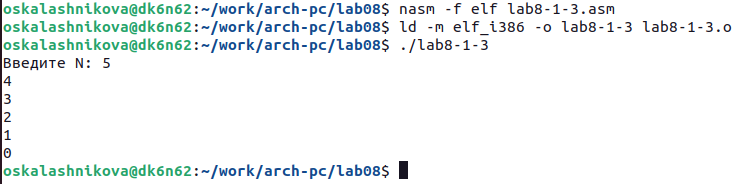
* Какие значения принимает регистр ecx в цикле? - Регистр ecx принимает некрректные значения.
* Соответствует ли число проходов цикла значению N введенному с клавиатуры? Число проходов цикла не соответствуют значению N введенному с клавиатуры.

Меняем текст программы (Используем стек. Добавим команды push и pop (добавления в стек и извлечения из стека) для сохранения значения счетчика цикла loop) (рис. ??)



Изменённый текст программы

Создаём исполняемый файл и запускаем его (компиляция: nasm -f elf lab8-1-3.asm , ld -m elf\_i386 -o lab8-1-3 lab8-1-3.o , запуск: ./lab8-1-3) (рис. ??)



Создание и запуск исполняемого файла

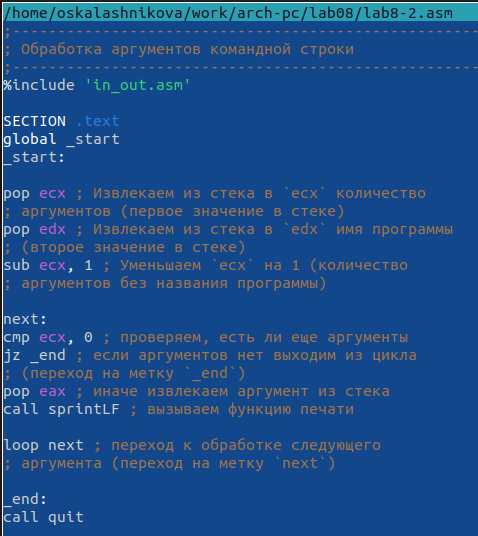
Соответствует ли в данном случае число проходов цикла значению N введенному с клавиатуры? - Число проходов цикла соответствуют значению N введенному с клавиатуры, но вывод идёт от 4 до 0 из-за того что ecx-1 теперь работает.

Создаём файл lab8-2.asm в каталоге ~/work/arch-pc/lab08 (при помощи команды touch lab8-2.asm) (рис. ??)

Создание файла

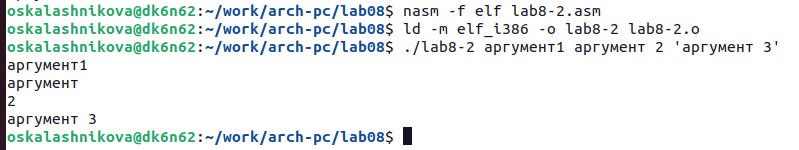
Создание файла

Введём в файл lab8-2.asm текст программы из листинга 8.2. (рис. ??)



Текст программы

Создаём исполняемый файл и запускаем его, указав аргументы: аргумент1 аргумент 2 ‘аргумент 3’ (компиляция: nasm -f elf lab8-1-3.asm , ld -m elf\_i386 -o lab8-1-3 lab8-1-3.o , запуск: ./lab8-1-3) (рис. ??)



Создание и запуск исполняемого файла

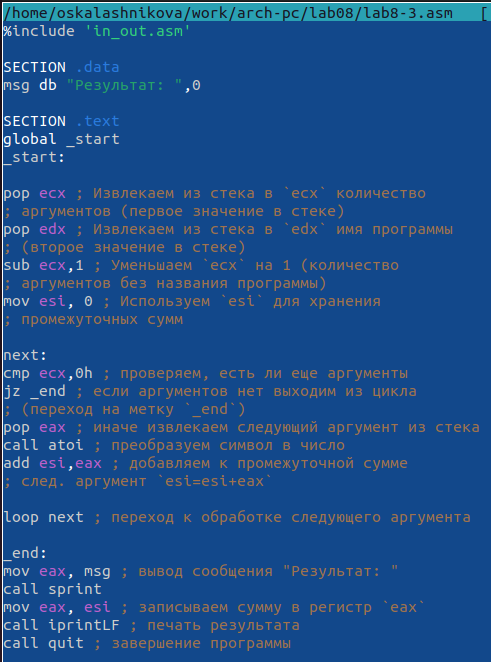
Сколько аргументов было обработано программой? - 4 аргумента, так как аргумент1 и ‘аргумент 3’ считаются по одному элементу, а аргумент и 2 это два разных (стоят через пробел без кавычек)

Создаём файл lab8-3.asm в каталоге ~/work/arch-pc/lab08 (при помощи команды touch lab8-3.asm) (рис. ??)

Создание файла

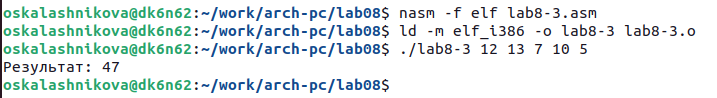
Создание файла

Введём в файл lab8-3.asm текст программы из листинга 8.3. (Программа вычисления суммы аргументов командной строки) (рис. ??)



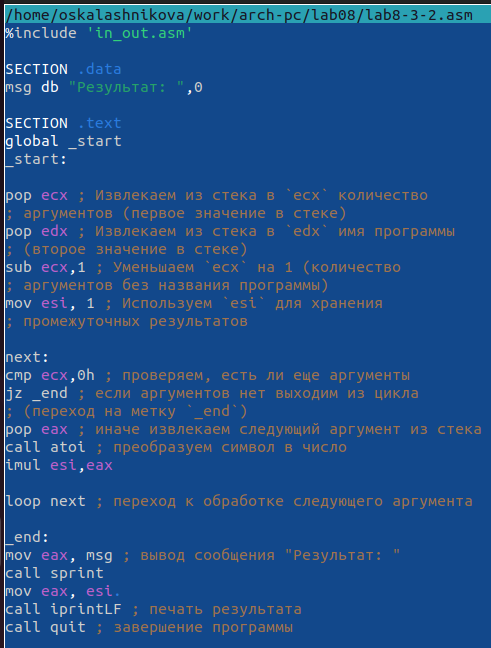
Текст программы

Создаём исполняемый файл и запускаем его, указав аргументы: 12 13 7 10 5 (компиляция: nasm -f elf lab8-3.asm , ld -m elf\_i386 -o lab8-3 lab8-3.o , запуск: ./lab8-3) (рис. ??)



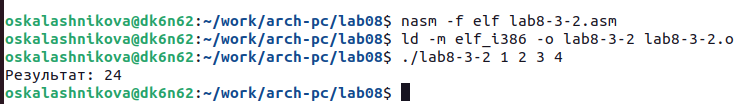
Создание и запуск исполняемого файла

Изменим текст программы из листинга 8.3 для вычисления произведения аргументов командной строки (рис. ??)



Изменённый текст программы

Создаём исполняемый файл и запускаем его, указав аргументы: 12 13 7 10 5 (компиляция: nasm -f elf lab8-3-2.asm , ld -m elf\_i386 -o lab8-3-2 lab8-3-2.o , запуск: ./lab8-3-2) (рис. ??)



Создание и запуск исполняемого файла

%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 результатов  
  
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  
imul esi,eax  
  
loop next ; переход к обработке следующего аргумента  
  
\_end:  
mov eax, msg ; вывод сообщения "Результат: "  
call sprint  
mov eax, esi   
call iprintLF ; печать результата  
call quit ; завершение программы

##Задание для самостоятельной работы

Задание 1: Напишем программу, которая находит сумму значений функци 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) возьмём из таблицы 8.1 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. (рис. ??), (рис. ??)

Вариант 7

Вариант 7

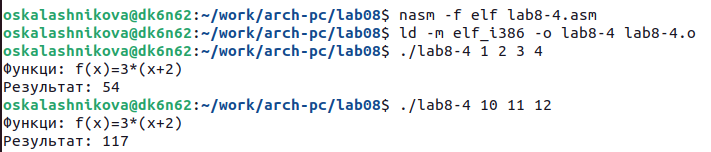


Текст программы

Текст программы:

%include 'in\_out.asm'  
  
SECTION .data  
msg db "Результат: ",0  
msg1 db "Функци: f(x)=3\*(x+2)",0h  
  
SECTION .text  
global \_start  
\_start:  
  
mov eax,msg1  
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,3  
mul ebx  
add esi,eax  
  
loop next  
  
\_end:  
mov eax, msg   
call sprint  
mov eax, esi   
call iprintLF   
call quit

Создаём исполняемый файл и проверяем его работу на нескольких наборах (я использовала 1,2,3,4 и 10,11,12) (компиляция: nasm -f elf lab8-4.asm , ld -m elf\_i386 -o lab8-4 lab8-4.o , запуск: ./lab8-4) (рис. ??)



Создание и запуск исполняемого файла

Посчитав значения сами мы понимаем, что программа работает верно.

# 3 Выводы

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