Отчёт по лабораторной работе №9

Дисциплина: Архитектура компьютера

Мишина Анастасия Алексеевна

Содержание

# 1 Цель работы

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

# 2 Выполнение лабораторной работы

Для начала создадим каталог для программ 9-ой лабораторной работы, перейдем в него и создадим файл lab9-1.asm (рис. 1).

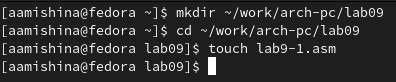


Рис. 1: Создание каталога и файла

Вводим текст программы из листинга 9.1 в наш файл. Создадим и запустим исполняемый файл, удостоверимся в его работе (рис. 2).

Программа lab9-1.asm:

;-----------------------------------------------------------------  
; Программа вывода значений регистра 'ecx'  
;-----------------------------------------------------------------  
%include 'in\_out.asm'  
SECTION .data  
msg1 db 'Введите N: ',0h  
SECTION .bss  
N: resb 10  
SECTION .text  
global \_start  
\_start:  
; ----- Вывод сообщения 'Введите N: '  
mov eax,msg1  
call sprint  
; ----- Ввод 'N'  
mov ecx, N  
mov edx, 10  
call sread  
; ----- Преобразование 'N' из символа в число  
mov eax,N  
call atoi  
mov [N],eax  
; ------ Организация цикла  
mov ecx,[N] ; Счетчик цикла, `ecx=N`  
label:  
mov [N],ecx  
mov eax,[N]  
call iprintLF ; Вывод значения `N`  
loop label ; `ecx=ecx-1` и если `ecx` не '0'  
; переход на `label`  
call quit

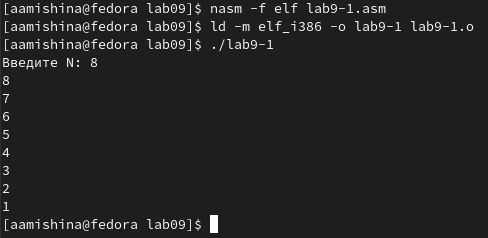


Рис. 2: Работа с файлом lab9-1.asm

Меняем текст программы. Создаем исполняемый файл, видим, что при вводе четного числа выводятся только нечетные. При вводе нечетного числа видим бесконечный вывод чисел (рис. 3).

Измененная программа lab9-1.asm:

;-----------------------------------------------------------------  
; Программа вывода значений регистра 'ecx'  
;-----------------------------------------------------------------  
%include 'in\_out.asm'  
SECTION .data  
msg1 db 'Введите N: ',0h  
SECTION .bss  
N: resb 10  
SECTION .text  
global \_start  
\_start:  
; ----- Вывод сообщения 'Введите N: '  
mov eax,msg1  
call sprint  
; ----- Ввод 'N'  
mov ecx, N  
mov edx, 10  
call sread  
; ----- Преобразование 'N' из символа в число  
mov eax,N  
call atoi  
mov [N],eax  
; ------ Организация цикла  
mov ecx,[N] ; Счетчик цикла, `ecx=N`  
label:  
sub ecx,1 ; `ecx=ecx-1`  
mov [N],ecx  
mov eax,[N]  
call iprintLF  
loop label  
; переход на `label`  
call quit

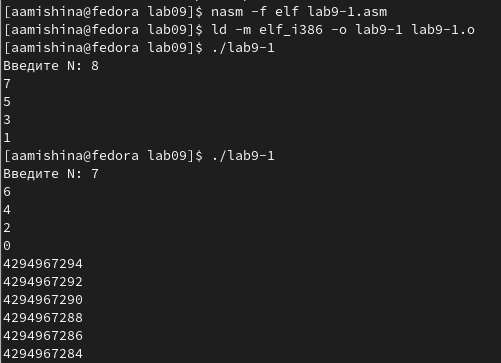


Рис. 3: Работа с измененным файлом lab9-1.asm

Теперь будем использовать стек для сохранения значения счетчика в цикле и корректности работы программы (рис. 4). Как видно, при вводе значения N выводятся числа от N-1 до 0. Число проходов цикла соответствует значению N.

Измененная программа lab9-1.asm:

;-----------------------------------------------------------------  
; Программа вывода значений регистра 'ecx'  
;-----------------------------------------------------------------  
%include 'in\_out.asm'  
SECTION .data  
msg1 db 'Введите N: ',0h  
SECTION .bss  
N: resb 10  
SECTION .text  
global \_start  
\_start:  
; ----- Вывод сообщения 'Введите N: '  
mov eax,msg1  
call sprint  
; ----- Ввод 'N'  
mov ecx, N  
mov edx, 10  
call sread  
; ----- Преобразование 'N' из символа в число  
mov eax,N  
call atoi  
mov [N],eax  
; ------ Организация цикла  
mov ecx,[N] ; Счетчик цикла, `ecx=N`  
label:  
push ecx ; добавление значения ecx в стек  
sub ecx,1  
mov [N],ecx  
mov eax,[N]  
call iprintLF  
pop ecx ; извлечение значения ecx из стека  
loop label  
; переход на `label`  
call quit

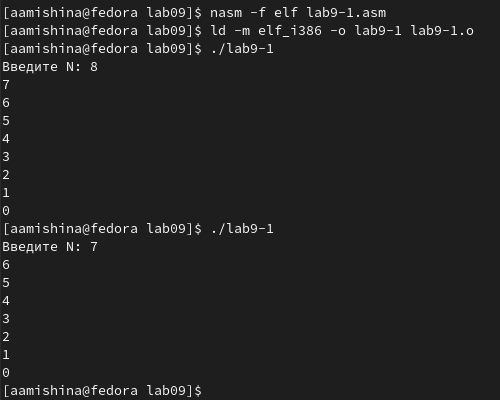


Рис. 4: Работа с измененным файлом lab9-1.asm

Создаем файл lab9-2.asm, вставляем в него текст программы из листинга 9.2 для обработки аргументов командной строки. Создаем и запускаем исполняемый файл, проверяем его работу (рис. 5). Программа обработала 5 аргументов.

Программа lab9-2.asm:

;-----------------------------------------------------------------  
; Обработка аргументов командной строки  
;-----------------------------------------------------------------  
%include 'in\_out.asm'  
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)  
next:  
cmp ecx, 0 ; проверяем, есть ли еще аргументы  
jz \_end ; если аргументов нет выходим из цикла  
; (переход на метку `\_end`)  
pop eax ; иначе извлекаем аргумент из стека  
call sprintLF ; вызываем функцию печати  
loop next ; переход к обработке следующего  
; аргумента (переход на метку `next`)  
\_end:  
call quit

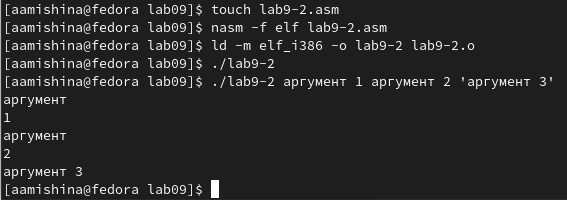


Рис. 5: Работа с файлом lab9-2.asm

Далее создадим файл lab9-3.asm и введем в него текст программы вычисления суммы аргументов командной строки из листинга 9.3. (рис. 6). Удостоверяемся в работе программы.

Программа lab9-3.asm:

%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, 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 esi,eax ; добавляем к промежуточной сумме  
; 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

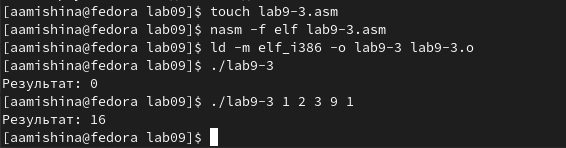


Рис. 6: Работа с файлом lab9-3.asm

Изменим текст файла lab9-3.asm, чтобы теперь программа считала произведение введенных аргументов (рис. 7).

Измененная программа lab9-3.asm:

%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  
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

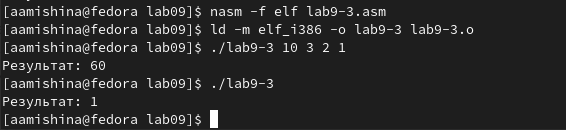


Рис. 7: Работа с измененным файлом lab9-3.asm

# 3 Выполнение заданий самостоятельной работы

Напишем программу, которая находит сумму значений функций для введенных аргументов. Согласно моему варианту (13) мне следовало использовать функцию f(x) = 12x - 7. Тестируем работу программы. Программа отработала успешно (рис. 8).

Программа lab9my.asm:

%include 'in\_out.asm'  
SECTION .data  
msg db "Результат: ", 0  
func: db 'f(x) = 12x - 7', 0  
  
SECTION .text  
global \_start  
\_start:  
mov eax, func  
call sprintLF  
  
pop ecx  
pop edx  
sub ecx, 1  
mov esi, 0  
  
next:  
cmp ecx, 0h  
jz \_end  
  
pop eax  
call atoi  
  
mov ebx, 12  
mul ebx  
  
sub eax, 7  
add esi, eax  
  
loop next  
  
\_end:  
mov eax, msg  
call sprint  
  
mov eax, esi  
call iprintLF  
  
call quit

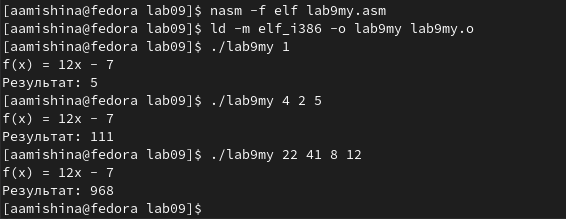


Рис. 8: Тестирование программы lab9my.asm

# 4 Выводы

В ходе выполнения данной лабораторной работы я приобрела навыки написания программ с использованием циклов и обработкой аргументов командной строки. Вся моя работа была записана и прокомментирована мной в данной лабораторной.