Отчёт по лабораторной работе №7

Архитектура компьютера

Морозова Мария Вячеславовна

Содержание

# 1 Цель работы

Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

# 2 Задание

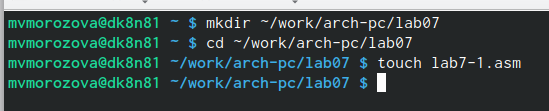
Написать программу для нахождения наименьшего из трёх чисел, программу для вычисления значения функции f(x).

# 3 Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов: • условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия. • безусловный переход – выполнение передачи управления в определенную точку про- граммы без каких-либо условий.

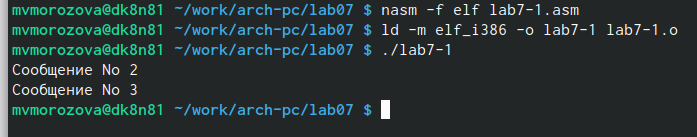
# 4 Выполнение лабораторной работы

Создала каталог для программ лабораторной работы № 7, перешла в него и со- здала файл lab7-1.asm: (рис. ??).



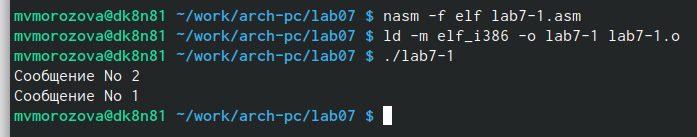
Создание файла

Создала исполняемый файл и запустила его с кодом листинга 7.1. (рис. ??).



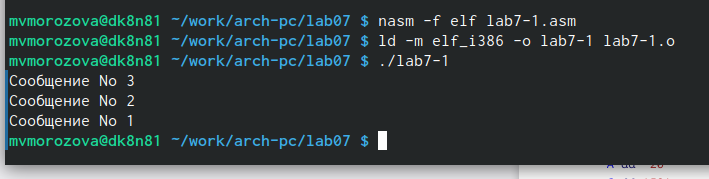
Создание файла, запуск

Создала исполняемый файл и запустила его с кодом листинга 7.2. (рис. ??).



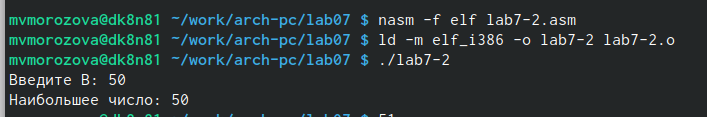
Создание файла, запуск

Создала исполняемый файл и запустила его с изменениями инструкции jmp. (рис. ??).



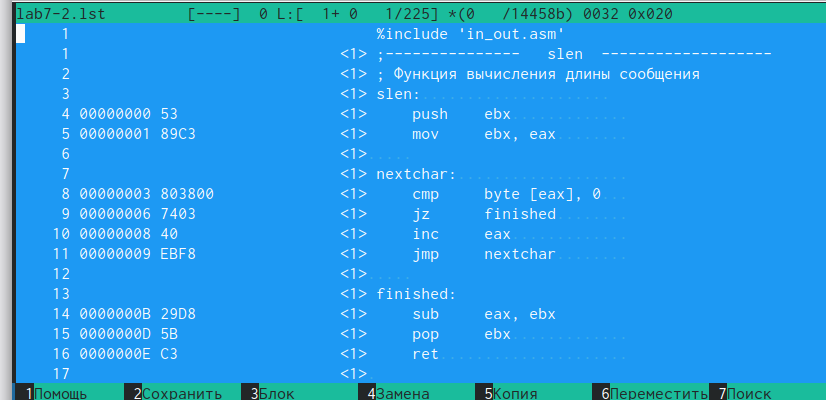
Результат работы программы

Создала файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Запустила с использованием программы из листинга 7.3. (рис. ??).



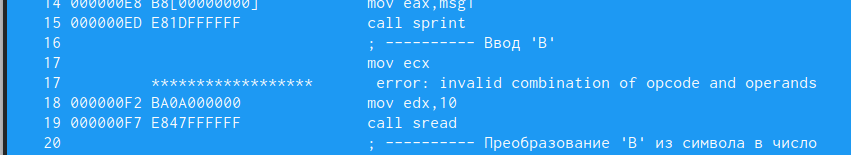
Создание файла, запуск

Создала файл листинга для программы из файла lab7-2.asm, открыла файл листинга lab7-2.lst с помощью текстового редактора. Рассмотрим несколько строк: строке 9 содержится номер сторки [9], адресс [00000003], машинный код [803800] и содержимое строки кода [cmp byte [eax], 0] в строке 11 содержится номер сторки [11], адресс [00000008], машинный код [40] и содержимое строки кода [inc eax] в строке 24 содержится номер сторки [24], адресс [0000000F], машинный код [52] и содержимое строки кода [push edx] (рис. ??).



Файл листинга

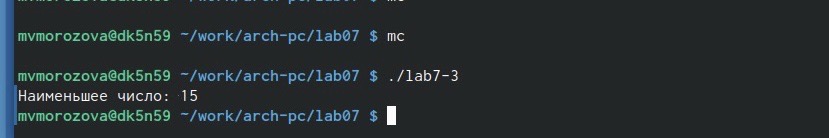
Удалила один операнд, чтобы посмотреть какая будет получена ошибка. (рис. ??).



Ошибка

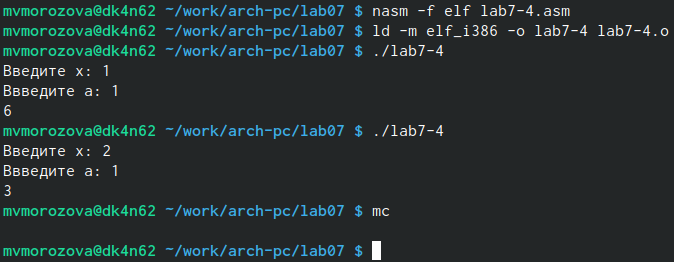
# 5 Выполнение самостоятельной работы

Запустила программу для вычисления наименьшего из трёх чисел. (рис. ??).



Результат работы программы

Создала исполняемый файл и запустила программу для нахождения f(x), проверила работу для двух пар x,a. (рис. ??).



Компоновка файла, запуск программы.

# 6 Выводы

Изучили команды условного и безусловного переходов. Приобрели навыки написания программ с использованием переходов. Познакомились с назначением и структурой файла листинга.

# 7 Листинги

lab7-3  
  
%include 'in\_out.asm'  
  
section .data  
 msg1 db "Наименьшее число:"  
 a dd 45  
 b dd 67  
 c dd 15  
  
section .bss  
 min resb 10  
  
section .text  
global \_start  
  
\_start:  
 mov eax, msg1  
 call sprint  
  
 mov ecx, [a]  
 mov [min], ecx ; 'min = A'  
 ; ---------- Сравниваем 'A' и 'С' (как числа)  
 cmp ecx, [c] ; Сравниваем 'A' и 'С'  
 jl check\_B ; если 'A<C', то переход на метку 'check\_B',  
 mov ecx, [c] ; иначе 'ecx = C'  
 mov [min], ecx ; 'min = C'  
; ---------- Преобразование 'min(A,C)' из символа в число  
  
check\_B:  
 ; ---------- Сравниваем 'min(A,C)' и 'B' (как числа)  
mov ecx, [min]  
 cmp ecx, [b] ; Сравниваем 'min(A,C)' и 'B'  
 jl fin ; если 'min(A,C)>B', то переход на 'fin',  
 mov ecx, [b] ; иначе 'ecx = B'  
  
 mov [min], ecx  
  
; ---------- Вывод результата  
fin:  
 mov eax, [min]  
 call iprintLF ; Вывод 'min(A,B,C)'  
 call quit ; Выход  
   
lab7-4  
  
%include 'in\_out.asm'  
  
SECTION .data  
input1 db "Введите х: ",0h  
input2 db "Ввведите а: ",0h  
  
SECTION .bss  
max resb 10  
x resb 10  
a resb 10  
  
SECTION .text  
GLOBAL \_start  
  
\_start:  
mov eax,input1  
call sprint  
  
mov ecx,x  
mov edx,10  
call sread  
  
mov eax,x  
call atoi   
mov [x],eax   
  
mov eax,input2  
call sprint  
  
mov ecx,a  
mov edx,10  
call sread  
  
mov eax,a  
call atoi   
mov [a],eax   
  
mov ebx, [x]  
cmp [a], ebx  
je check  
  
mov eax, [a]  
mov ebx, [x]  
add eax, ebx  
call iprintLF  
call quit  
  
check:  
mov eax, [a]  
mov ebx, 6  
mul ebx  
call iprintLF  
call quit