Отчёт по выполнению лабораторной работы №7

Дисциплина: архитектура компьютера

Лысенко Маргарита Олеговна

Содержание

Список иллюстраций

Список таблиц

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

Разобраться в командах условного и безусловного переходов.

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 программы без каких-либо условий.

4 Выполнение лабораторной работы

Создала каталог для программам лабораторной работы № 7, перешла в него и со- здала файл lab7-1.asm (рис. 4.1).

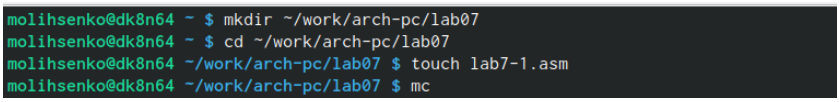

Рис. 4.1: Создание каталога и файла

Ввела программу из листинга 7.1. Создала исполняемый файл и запустила его. (рис. 4.2).

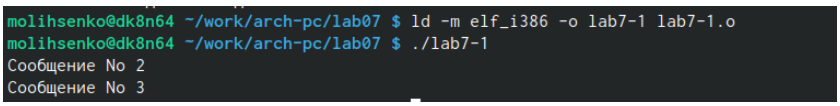

Рис. 4.2: Запуск файла

Изменила программу так, чтобы сначала выводилась 2, а потом 3 (рис. 4.3).

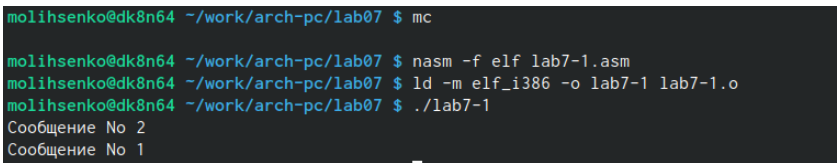

Рис. 4.3: Редактирование программы

Изменила текст программы в соответсвии с листингом 7.2. Создала исполняемый файл и проверила его работу. Изменила текст программы добавив или изменив инструкции jmp, чтобы вывод програм- мы был следующим: user@dk4n31:~$

./lab7-1 Сообщение № 3 Сообщение № 2 Сообщение № 1 user@dk4n31:~$ (рис. 4.4).

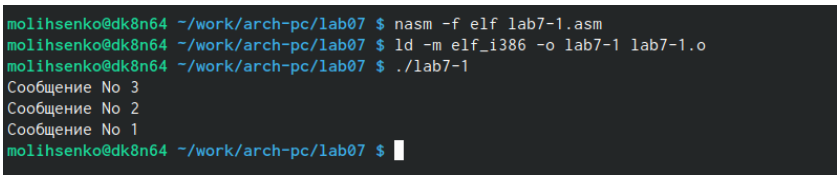

Рис. 4.4: Запуск файла

Создала файл lab7-2.asm в каталоге ~/work/arch-pc/lab07. Внимательно изучила текст программы из листинга 7.3 и ввела в lab7-2.asm. Создала исполняемый файл и проверила его работу для разных значений B. (рис. 4.5).

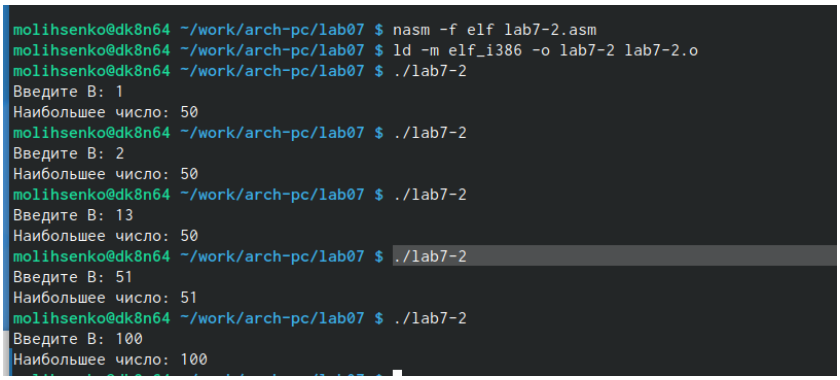

Рис. 4.5: Проверка файла

Открыла файл листинга lab7-2.lst с помощью текстового редактора. Внимательно ознакомилась с его форматом и содержимым. В строке 9 содержится собственно 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. 4.6).

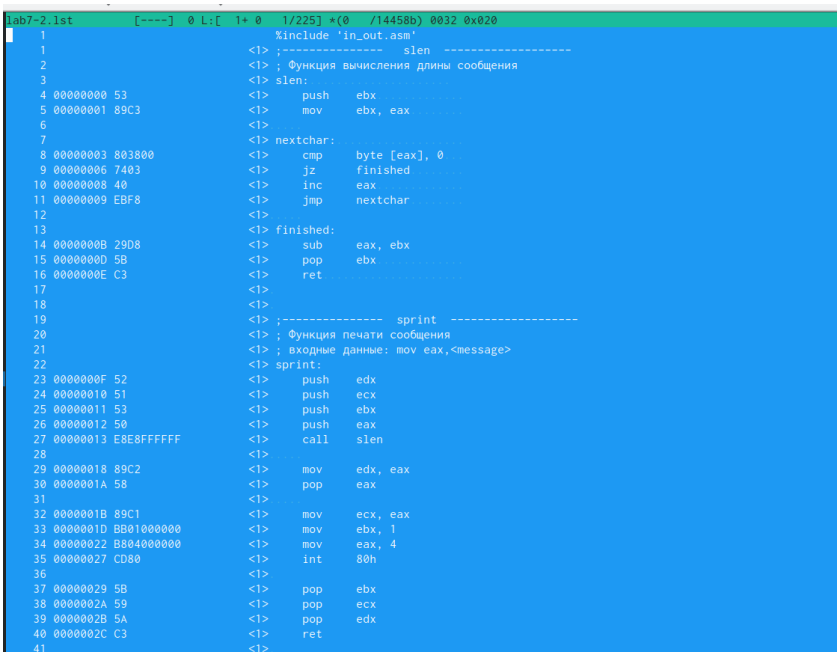

Рис. 4.6: Открытие файла листинга

В инструкции с двумя операндами удалила один операнд. Выполнила трансляцию с получением файла листинга: nasm -f elf -l lab7-2.lst lab7-2.asm Видно, что ели в коде появляется ошибка, то ее видно в листинге. (рис. 4.7).

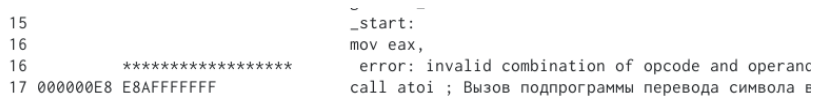

Рис. 4.7: Выявление ошибки

5 Выполнение самостоятельной работы

Написала программу нахождения наименьшей из 3 целочисленных переменных a,b и c. Значения переменных выбрать из табл. 7.5 в соответствии с вариантом 10. Создала исполняемый файл и проверила его работу. (рис. 5.1).

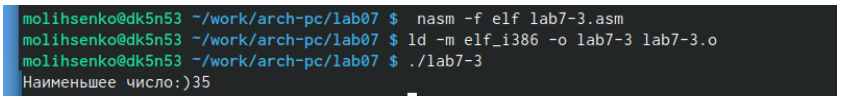

Рис. 5.1: Запуск файла

Написала программу, которая для введенных с клавиатуры значений ☒ и ☒ вычисляет значение заданной функции ☒(☒) и выводит результат вычислений. Вид функции ☒(☒) выбрала из таблицы 7.6 вариантов заданий в соответствии с вариантом 10. Создала исполняемый файл и проверила его работу для значений ☒ и ☒ . (рис. 5.2).

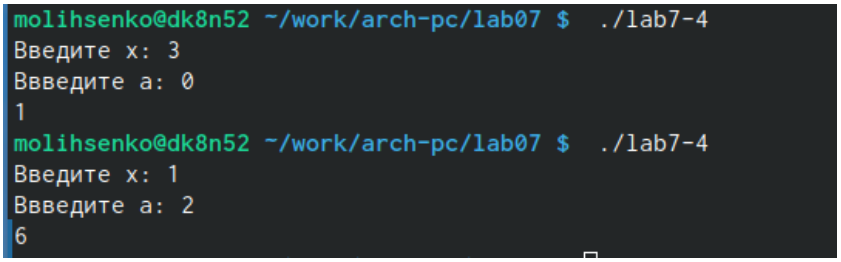

Рис. 5.2: Запуск файла

6 Листинги программ

```
%include 'in_out.asm'

section .data
    msg1 db "Наименьшее число:"
    a dd 41
    b dd 62
    c dd 35

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 'C' (как числа)
    cmp ecx, [c] ; Сравниваем 'A' и 'C'
```

```
    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

%include 'in_out.asm'

SECTION .data
input1 db "Введите x: ",0h
input2 db "Ввведите a: ",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, 2
cmp [x], ebx
```

```
jle check

mov eax, [x]
mov ebx, 2
sub eax, ebx
call iprintLF
call quit

check:
mov eax, [a]
mov ebx, 3
imul ebx
call iprintLF
call quit
```

7 Выводы

В ходе лабораторной и самостоятельной работ я изучила команды условного и безусловного переходов. Приобрела навыки написания программ с использованием переходов. Познакомилась с назначением и структурой файла листинга.

Список литературы