Отчёт по лабораторной работе №7

Дисциплина: архитектура компьютера

Булыгин Николай Александрович

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Целью работы является изучение команд условного и безусловного переходов, приобретение навыков написания программ с использованием переходов и знакомство с назначением и структурой файла листинга.

# 2 Задание

1. Реализация переходов в NASM
2. Изучение структуры файлы листинга
3. Выполнение самостоятельной работы

# 3 Выполнение лабораторной работы

## 3.1 Реализация переходов в NASM

Создаю каталог для программ в этой лабораторной работе, перехожу в него и создаю файл lab7-1.asm (рис. 1).

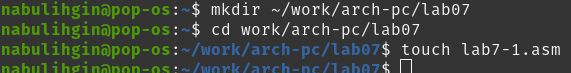


Рис. 1: lab7-1.asm

Ввожу в файл lab7-1.asm текст программы:

%include 'in\_out.asm'  
  
SECTION .data  
msg1: DB 'Сообщение № 1',0  
msg2: DB 'Сообщение № 2',0  
msg3: DB 'Сообщение № 3',0  
  
SECTION .text  
GLOBAL \_start  
\_start:  
  
jmp \_label2  
   
\_label1:  
 mov eax, msg1  
 call sprintLF  
   
\_label2:  
 mov eax, msg2  
 call sprintLF  
   
\_label3:  
 mov eax, msg3  
 call sprintLF  
   
\_end:  
 call quit

Создаю исполняемый файл и запускаю его. Выводится сначала Сообщение 2, потом Сообщение 3, так как jmp \_label2 осуществляет переход к \_label2, что пропускает \_label1 (рис. 2).

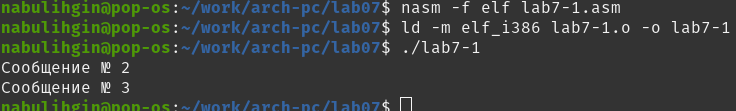


Рис. 2: Вывод программы

Меняю текст lab7-1.asm в соответствие с листингом 7.2:

%include 'in\_out.asm'  
  
SECTION .data  
msg1: DB 'Сообщение № 1',0  
msg2: DB 'Сообщение № 2',0  
msg3: DB 'Сообщение № 3',0  
  
SECTION .text  
GLOBAL \_start  
\_start:  
  
jmp \_label2  
   
\_label1:  
 mov eax, msg1  
 call sprintLF  
 jmp \_end  
   
\_label2:  
 mov eax, msg2  
 call sprintLF  
 jmp \_label1  
   
\_label3:  
 mov eax, msg3  
 call sprintLF  
   
\_end:  
 call quit

Создаю исполняемый файл и запускаю его. Выводится сначала Сообщение 2, потом Сообщение 1, так как jmp \_label2 осуществляет переход к \_label2, затем jmp \_label1 переходит на *label1 и jmp* end вызывает завершение (рис. 3).

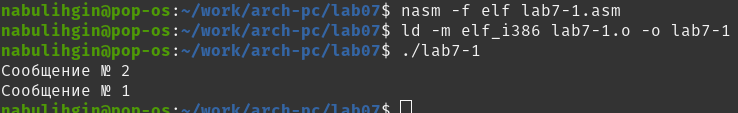


Рис. 3: Новый вывод программы

Снова меняю текст программы, чтобы сообщения выводились в порядке 3 2 1:

%include 'in\_out.asm'  
  
SECTION .data  
msg1: DB 'Сообщение № 1',0  
msg2: DB 'Сообщение № 2',0  
msg3: DB 'Сообщение № 3',0  
  
SECTION .text  
GLOBAL \_start  
\_start:  
  
jmp \_label3  
   
\_label1:  
 mov eax, msg1  
 call sprintLF  
 jmp \_end  
   
\_label2:  
 mov eax, msg2  
 call sprintLF  
 jmp \_label1  
   
\_label3:  
 mov eax, msg3  
 call sprintLF  
 jmp \_label2  
   
\_end:  
 call quit

Снова создаю исполняемый файл и запускаю его, вывод соответсвует условию задания (рис. 4).

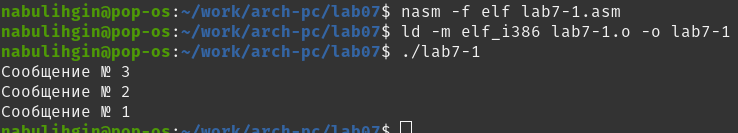


Рис. 4: Порядок вывода 3 2 1

Создаю файл lab7-2.asm (рис. 5).

Рис. 5: lab7-2.asm

Рис. 5: lab7-2.asm

Ввожу в него текст программы из листинга 7.3:

%include 'in\_out.asm'  
section .data  
 msg1 db 'Введите B: ',0h  
 msg2 db "Наибольшее число: ",0h  
 A dd '20'  
 C dd '50'  
section .bss  
max resb 10  
B resb 10  
section .text  
 global \_start  
\_start:  
 mov eax,msg1  
 call sprint  
  
 mov ecx,B  
 mov edx,10  
 call sread  
  
 mov eax,B  
 call atoi  
 mov [B],eax  
  
 mov ecx,[A]  
 mov [max],ecx  
  
 cmp ecx,[C]  
 jg check\_B  
 mov ecx,[C]  
 mov [max],ecx  
  
check\_B:  
 mov eax,max  
 call atoi  
 mov [max],eax  
  
 mov ecx,[max]  
 cmp ecx,[B]  
 jg fin  
 mov ecx,[B]  
 mov [max],ecx  
  
fin:  
 mov eax, msg2  
 call sprint  
 mov eax,[max]  
 call iprintLF  
 call quit

Создаю исполняемый файл и проверяю его работу для разных значений B (рис. 6).

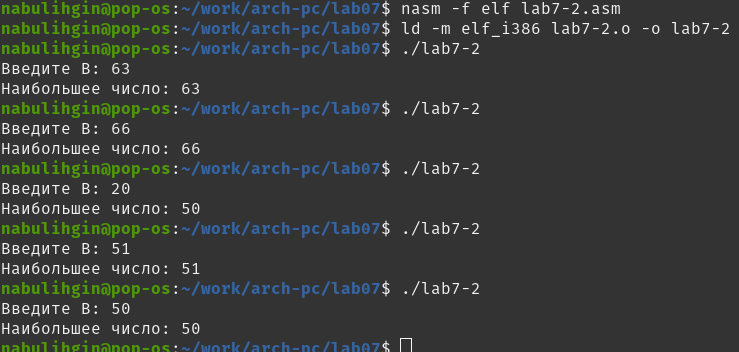


Рис. 6: lab7-2

## 3.2 Изучение структуры файлы листинга

Создаю файл листинга для программы из файла lab7-2.asm (рис. 7).

Рис. 7: Создание файла листинга

Рис. 7: Создание файла листинга

Открываю полученный файл листинга lab7-2.lst в mcedit (рис. 8).

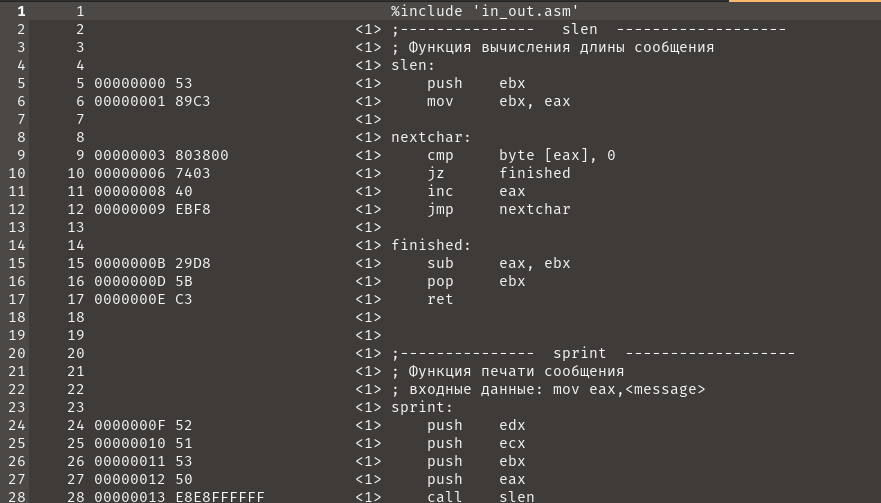


Рис. 8: lab7-2.lst

Объясняю содержимое трех строк файла по выбору:

Строка на 16 месте, адрес - 000000F2, машинный код - B9[0A000000], mov ecx,B - исходный текст программы, в регистр ecx вносится значение переменной B (рис. 9).

Рис. 9: Строка 16

Рис. 9: Строка 16

Строка на 21 месте, адрес - 00000106, машинный код - E891FFFFFF, call atoi - исходный текст программы, символ в строке выше переводится в число (рис. 10).

Рис. 10: Строка 21

Рис. 10: Строка 21

Строка на 30 месте, адрес - 0000012A, машинный код - 890D[00000000], mov [max],ecx - исходный текст программы, в регистр ecx вносится число, хранившееся в переменной max (рис. 11).

Рис. 11: Строка 30

Рис. 11: Строка 30

В строке 44 удаляю операнд msg2, выполняю трансляцию с получением файла листинга и получаю ошибку (рис. 12).

Рис. 12: Ошибка

Рис. 12: Ошибка

В файле листинга добавляется сообщение об ошибке (рис. 13).

Рис. 13: Сообщение об ошибке

Рис. 13: Сообщение об ошибке

## 3.3 Выполнение самостоятельной работы

Создаю файл task1.asm и пишу программу для нахождения наименьшего из 3 чисел. У меня второй вариант, поэтому в программе задаю числа A = 82, B = 59, C = 61. Программа выводит 59 (рис. 14).

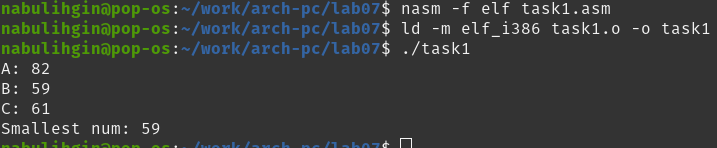


Рис. 14: Задание 1

Код программы:

%include 'in\_out.asm'  
section .data  
  
msga db 'A: ',0h  
msgb db 'B: ',0h  
msgc db 'C: ',0h  
msg1 db "Smallest num: ",0h  
section .bss  
min resb 10  
A resb 10  
B resb 10  
C resb 10  
section .text  
global \_start  
\_start:  
  
mov eax,msga  
call sprint  
mov ecx,A  
mov edx, 10  
call sread  
mov eax,msgb  
call sprint  
mov ecx,B  
call sread  
mov eax,msgc  
call sprint  
mov ecx,C  
call sread  
  
mov eax,A  
call atoi  
mov [A],eax  
mov eax,B  
call atoi  
mov [B],eax  
mov eax,C  
call atoi  
mov [C],eax  
  
mov ecx,[A]  
add ecx,[B]  
add ecx,[C]  
mov [min],ecx  
  
jmp cmp\_a\_b  
  
min\_below\_c:  
jmp fin  
  
c\_below\_min:  
mov eax,[C]  
mov [min],eax  
jmp fin  
  
a\_below\_b:  
mov eax,[A]  
mov [min],eax  
jmp cmp\_min\_c  
  
b\_below\_a:  
mov eax,[B]  
mov [min],eax  
jmp cmp\_min\_c  
  
cmp\_min\_c:  
mov eax,[C]  
cmp [min],eax  
jb min\_below\_c  
jg c\_below\_min  
  
cmp\_a\_b:  
mov eax,[B]  
cmp [A],eax  
jb a\_below\_b  
jg b\_below\_a  
  
fin:  
mov eax, msg1  
call sprint  
mov eax,[min]  
call iprintLF  
call quit

Создаю файл task2.asm и пишу программу для вычисления значения фунции второго варианта. При значениях x = 5, a = 7 она выводит a - 1 = 6, при значениях x = 6, a = 4 она выводит x - 1 = 5, следовательно, она работает корректно (рис. 15).

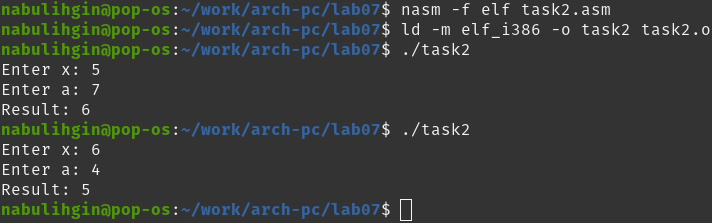


Рис. 15: Задание 2

Код программы:

%include 'in\_out.asm'  
section .data  
msgx db 'Enter x: ',0h  
msga db 'Enter a: ',0h  
msg1 db "Result: ",0h  
  
section .bss  
x resb 10  
a resb 10  
res resb 10  
  
section .text  
global \_start  
\_start:  
mov eax,msgx  
call sprint  
mov ecx,x  
mov edx,10  
call sread  
mov eax,msga  
call sprint  
mov ecx,a  
call sread  
mov eax,x  
call atoi  
mov [x],eax  
mov eax,a  
call atoi  
mov [a],eax  
  
mov eax, [a]  
mov ebx, [x]  
cmp ebx, eax  
jb x\_below\_a  
jae x\_above\_a  
  
x\_below\_a:  
mov ecx, [a]  
mov [res], ecx  
mov ecx, 1  
sub [res], ecx  
jmp fin  
  
x\_above\_a:  
mov ecx, [x]  
mov [res], ecx  
mov ecx, 1  
sub [res], ecx  
jmp fin  
  
fin:  
mov eax, msg1  
call sprint  
mov eax,[res]  
call iprintLF  
call quit

# 4 Выводы

Я изученил команды условного и безусловного переходов, приобрел навыки написания программ с использованием переходов и познакомился с назначением и структурой файла листинга.