Отчет по лабораторной работа №7

Дисциплина: архитектура компьютера

Ширинкин Т. Б.

Содержание

Цель работы

Изучение команд условного и безусловного переходов в Nasm, условий порядка и изучение структуры листинга.

Выполнение лабораторной работы

Создал каталог и первый файл (Рис. [-@fig:000])

```
tbshirinkin@Shiza:~$ mkdir ~/work/arch-pc/lab07
tbshirinkin@Shiza:~$ cd ~/work/arch-pc/lab07
tbshirinkin@Shiza:~/work/arch-pc/lab07$ touch lab7-1.asm
```

Рис. 0 Создал каталог и первый файл

Ввёл первый листинг (Рис. [-@fig:001])

```
%include 'in_out.asm' ; подключение внешнего файла
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 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 1'
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
_end:
call quit ; вызов подпрограммы завершения
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                    21,0-1        All
```

Рис. 1 Ввёл первый листинг

Создал исполняемый файл и запустил его (Рис. [-@fig:002])

```
tbshirinkin@Shiza:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
tbshirinkin@Shiza:~/work/arch-pc/lab07$
```

Рис. 2 Создал исполняемый файл и запустил его

Изменяю файл в соответствии со вторым листингом (Рис. [-@fig:003])

```
%include 'in_out.asm' ; подключение внешнего файла
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 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
_end:
call quit ; вызов подпрограммы завершения
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                    23,0-1        All
```

Рис. 3 Изменяю файл в соответствии со вторым листингом

Да, из-за переходов не показалась третья строка (Рис. [-@fig:004])

```
tbshirinkin@Shiza:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
tbshirinkin@Shiza:~/work/arch-pc/lab07$
```

Рис. 4 Да, из-за переходов не показалась третья строка

Изменил код в соответствии с заданием (Рис. [-@fig:005])

```
%include 'in_out.asm' ; подключение внешнего файла
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 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
jmp _label2
_end:
call quit ; вызов подпрограммы завершения
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                    24,1          All
```

Рис. 5 Изменил код в соответствии с заданием

Правда работает (Рис. [-@fig:006])

```
tbshirinkin@Shiza:~/work/arch-pc/lab07$ vi lab7-1.asm
tbshirinkin@Shiza:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
tbshirinkin@Shiza:~/work/arch-pc/lab07$
```

Рис. 6 Правда работает

Создал новый файл (Рис. [-@fig:007])

```
tbshirinkin@Shiza:~/work/arch-pc/lab07$ touch lab7-2.asm
```

Рис. 7 Создал новый файл

Ввёл в него новый листинг (Рис. [-@fig:008])

```
%include 'in_out.asm'
section .data
msg1 db 'Bведите 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:
; ---------- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ---------- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ---------- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ---------- Записываем 'A' в переменную 'max'
mov ecx,[A] ; 'ecx = A'
mov [max],ecx ; 'max = A'
; ---------- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[C] ; Сравниваем 'A' и 'C'
jg check_B ; если 'A>C', то переход на метку 'check_B',
mov ecx,[C] ; иначе 'ecx = C'
mov [max],ecx ; 'max = C'
; ---------- Преобразование 'max(A,C)' из символа в число
check_B:
mov eax,max
call atoi ; Вызов подпрограммы перевода символа в число
mov [max],eax ; запись преобразованного числа в `max`
; ---------- Сравниваем 'max(A,C)' и 'B' (как числа)
mov ecx,[max]
cmp ecx,[B] ; Сравниваем 'max(A,C)' и 'B'
jg fin ; если 'max(A,C)>B', то переход на 'fin',
mov ecx,[B] ; иначе 'ecx = B'
                                                    5,9           Top
```

Рис. 8 Ввёл в него новый листинг

Протестировал код на разных вводных значениях - работает в соответствии с написанным кодом. (Рис. [-@fig:009])

```
tbshirinkin@Shiza:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ./lab7-2
Введите B: -2
Наибольшее число: 50
tbshirinkin@Shiza:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 1
Наибольшее число: 50
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ./lab7-2
Введите B: s
Наибольшее число: 50
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 732
Наибольшее число: 732
```

Рис. 9 Протестировал код на разных вводных значениях - работает в соответствии с написанным кодом

Создвл файл lst (Рис. [-@fig:010])

```
tbshirinkin@Shiza:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
tbshirinkin@Shiza:~/work/arch-pc/lab07$ vi lab7-2.lst
```

Рис. 10 Создвл файл lst

Посмотрел (Рис. [-@fig:011])

```
    14 00000E8 B8[00000000]            mov eax,msg1
    15 000000ED E81DFFFFFF             call sprint
    16                                 ; ---------- Ввод 'B'
    17 000000F2 B9[0A000000]           mov ecx,B
    18 000000F7 BA0A000000             mov edx,10
    19 000000FC E842FFFFFF             call sread
    20                                 ; ---------- Преобразование 'B' из с
имвола в число
    21 00000101 B8[0A000000]           mov eax,B
    22 00000106 E891FFFFFF             call atoi ; Вызов подпрограммы перев
ода символа в число
    23 0000010B A3[0A000000]           mov [B],eax ; запись преобразованног
о числа в 'B'
    24                                 ; ---------- Записываем 'A' в переме
нную 'max'
    25 00000110 8B0D[35000000]         mov ecx,[A] ; 'ecx = A'
    26 00000116 890D[00000000]         mov [max],ecx ; 'max = A'
    27                                 ; ---------- Сравниваем 'A' и 'C' (к
ак символы)
    28 0000011C 3B0D[39000000]         cmp ecx,[C] ; Сравниваем 'A' и 'C'
    29 00000122 7F0C                   jg check_B ; если 'A>C', то переход
на метку 'check_B',
    30 00000124 8B0D[39000000]         mov ecx,[C] ; иначе 'ecx = C'
    31 0000012A 890D[00000000]         mov [max],ecx ; 'max = C'
    32                                 ; ---------- Преобразование 'max(A,C
)' из символа в число
    33                                 check_B:
    34 00000130 B8[00000000]           mov eax,max
    35 00000135 E862FFFFFF             call atoi ; Вызов подпрограммы перев
ода символа в число
    36 0000013A A3[00000000]           mov [max],eax ; запись преобразованн
ого числа в `max`
    37                                 ; ---------- Сравниваем 'max(A,C)' и
 'B' (как числа)
    38 0000013F 8B0D[00000000]         mov ecx,[max]
    39 00000145 3B0D[0A000000]         cmp ecx,[B] ; Сравниваем 'max(A,C)'
и 'B'
    40 0000014B 7F0C                   jg fin ; если 'max(A,C)>B', то перех
од на 'fin',
    41 0000014D 8B0D[0A000000]         mov ecx,[B] ; иначе 'ecx = B'
    42 00000153 890D[00000000]         mov [max],ecx
                                                          194,43        95%
```

Рис. 11 Посмотрел

Заменил в соответствии с заданием (Рис. [-@fig:012])

```
mov ecx,[C] ; иначе 'ecx = C'
mov [max],  ; 'max = C'
```

Рис. 12 Заменил в соответствии с заданием

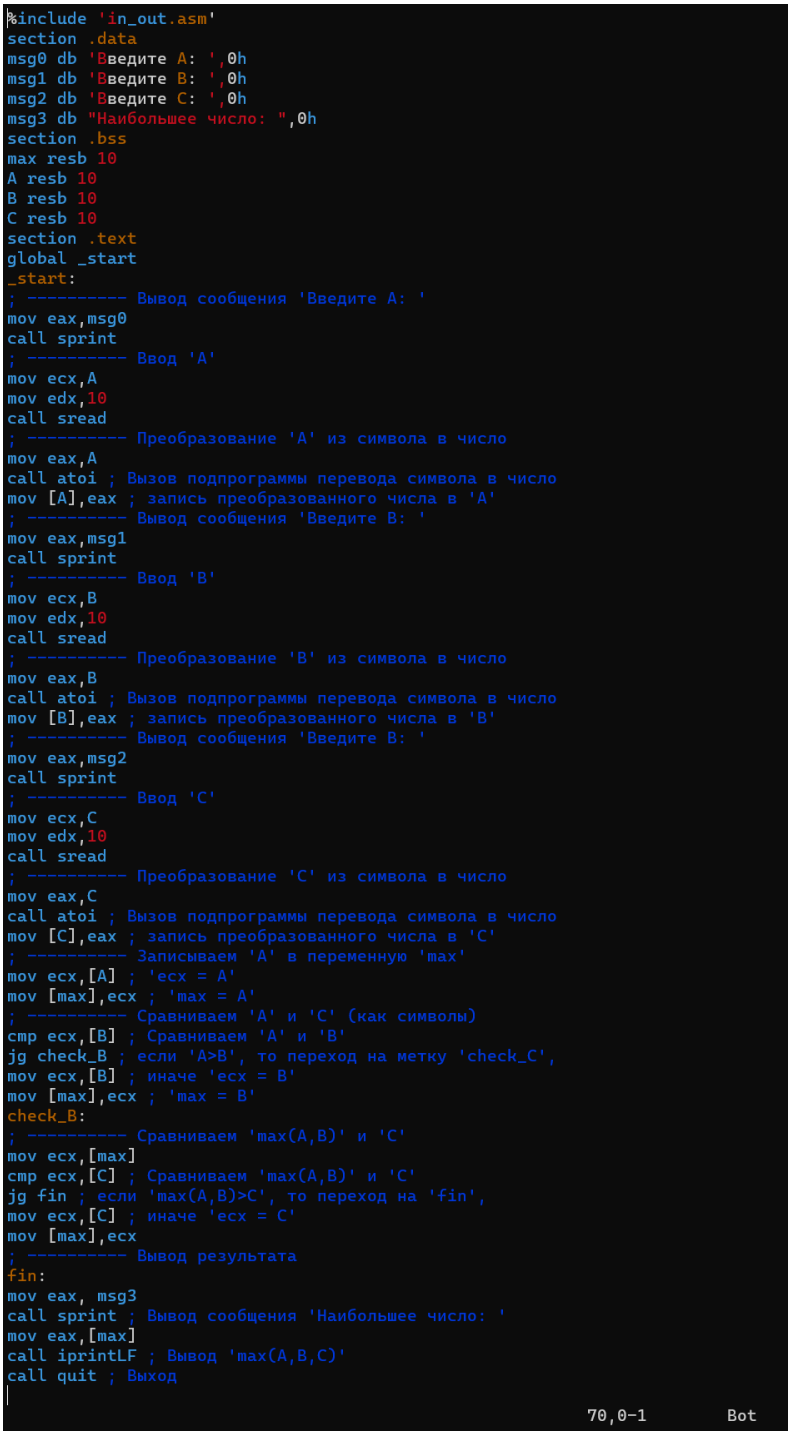
Ошибка (Рис. [-@fig:013])

```
tbshirinkin@Shiza:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
lab7-2.asm:31: error: invalid combination of opcode and operands
```

Рис. 13 Ошибка

Выполнение заданий для самостоятельной работы

Создал текст первой программы из программы lab7-2.asm (Рис. [-@fig:014])



```
%include 'in_out.asm'
section .data
msg0 db 'Введите A: ',0h
msg1 db 'Введите B: ',0h
msg2 db 'Введите C: ',0h
msg3 db "Наибольшее число: ",0h
section .bss
max resb 10
A resb 10
B resb 10
C resb 10
section .text
global _start
_start:
; ---------- Вывод сообщения 'Введите A: '
mov eax,msg0
call sprint
; ---------- Ввод 'A'
mov ecx,A
mov edx,10
call sread
; ---------- Преобразование 'A' из символа в число
mov eax,A
call atoi ; Вызов подпрограммы перевода символа в число
mov [A],eax ; запись преобразованного числа в 'A'
; ---------- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ---------- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ---------- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ---------- Вывод сообщения 'Введите B: '
mov eax,msg2
call sprint
; ---------- Ввод 'C'
mov ecx,C
mov edx,10
call sread
; ---------- Преобразование 'C' из символа в число
mov eax,C
call atoi ; Вызов подпрограммы перевода символа в число
mov [C],eax ; запись преобразованного числа в 'C'
; ---------- Записываем 'A' в переменную 'max'
mov ecx,[A] ; 'ecx = A'
mov [max],ecx ; 'max = A'
; ---------- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[B] ; Сравниваем 'A' и 'B'
jg check_B ; если 'A>B', то переход на метку 'check_C',
mov ecx,[B] ; иначе 'ecx = B'
mov [max],ecx ; 'max = B'
check_B:
; ---------- Сравниваем 'max(A,B)' и 'C'
mov ecx,[max]
cmp ecx,[C] ; Сравниваем 'max(A,B)' и 'C'
jg fin ; если 'max(A,B)>C', то переход на 'fin',
mov ecx,[C] ; иначе 'ecx = C'
mov [max],ecx
; ---------- Вывод результата
fin:
mov eax, msg3
call sprint ; Вывод сообщения 'Наибольшее число: '
mov eax,[max]
call iprintLF ; Вывод 'max(A,B,C)'
call quit ; Выход
                                                    70,0-1        Bot
```

Рис. 14 Создал текст программы

Собрал и ввёл данные своего номера (19) - работает как надо (после этого проверил и другие значения, тоже работает) (Рис. [-@fig:015])

```
tbshirinkin@Shiza:~/work/arch-pc/lab07$ nasm -f elf -l lab7-2.lst lab7-2.asm
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
tbshirinkin@Shiza:~/work/arch-pc/lab07$ ./lab7-2
Введите A: 46
Введите B: 32
Введите C: 74
Наибольшее число: 74
tbshirinkin@Shiza:~/work/arch-pc/lab07$
```

Рис. 15 Собрал и ввёл данные своего номера (19) - работает как надо (после этого проверил и другие значения, тоже работает)

Создал текст второй программы (для номера 19) опять из программы lab7-2.asm (Рис. [-@fig:016])

```
%include 'in_out.asm'
section .data
msg0 db 'Введите x: ',0h
msg1 db 'Введите a: ',0h
msg2 db "Вывод f(x): ",0h
section .bss
result resb 10
x resb 10
a resb 10
section .text
global _start
_start:



mov eax,msg0
call sprint

mov ecx,x
mov edx,10
call sread
mov eax,x

call atoi
mov [x],eax


mov eax,msg1
call sprint

mov ecx,a
mov edx,10
call sread

mov eax,a
call atoi
mov [a],eax


mov eax,[a]
mov ecx,[x]
add ecx,eax
mov [result],ecx

mov ecx,[x]
cmp ecx,eax
jg fin

mov [result],ecx

fin:

mov eax, msg2
call sprint
mov eax,[result]

call iprintLF



call quit
                                                    39,0-1        Bot
```

Рис. 16 Создал текст второй программы (для номера 19) опять из программы lab7-2.asm

Снова собрал и снова ввёл данные своего номера (19) - работает как надо (после этого как и в прошлый раз проверил и другие значения, опять работает) (Рис. [-@fig:017])

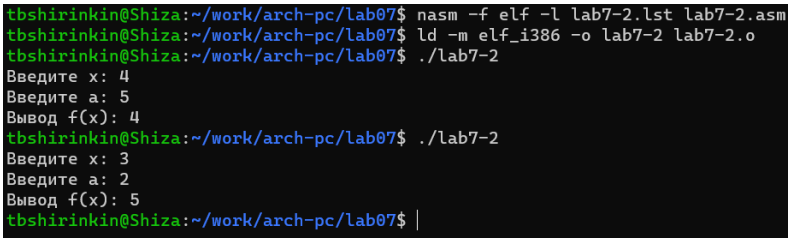


Рис. 17 Снова собрал и снова ввёл данные своего номера (19) - работает как надо (после этого как и в прошлый раз проверил и другие значения, опять работает)

Выводы

Цель достигнута: изучены команды условного и безусловного переходов в Nasm, условий порядка и изучены структуры листинга.