Отчёт по лабораторной работе 7

Архитектура компьютера

Бугерра Сухайеб

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Целью работы является изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Выполнение лабораторной работы

2.1 Реализация переходов в NASM

Создал каталог для программам лабораторной работы № 7 и файл lab7-1.asm

Инструкция jmp в NASM используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции jmp. Написал в файл lab7-1.asm текст программы из листинга 7.1.

lab7-1.asm

```
%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
```

Рис. 2.1: Программа в файле lab7-1.asm

Создал исполняемый файл и запустил его.

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

Рис. 2.2: Запуск программы lab7-1.asm

Инструкция jmp позволяет осуществлять переходы не только вперед но и назад. Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого в текст программы после вывода сообщения № 2 добавим инструкцию jmp с меткой _label1 (т.е. переход к инструкциям вывода сообщения № 1) и после вывода сообщения № 1 добавим инструкцию jmp с меткой _end (т.е. переход к инструкции call quit).

Изменил текст программы в соответствии с листингом 7.2.

lab7-1.asm

```
%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
```

Рис. 2.3: Программа в файле lab7-1.asm

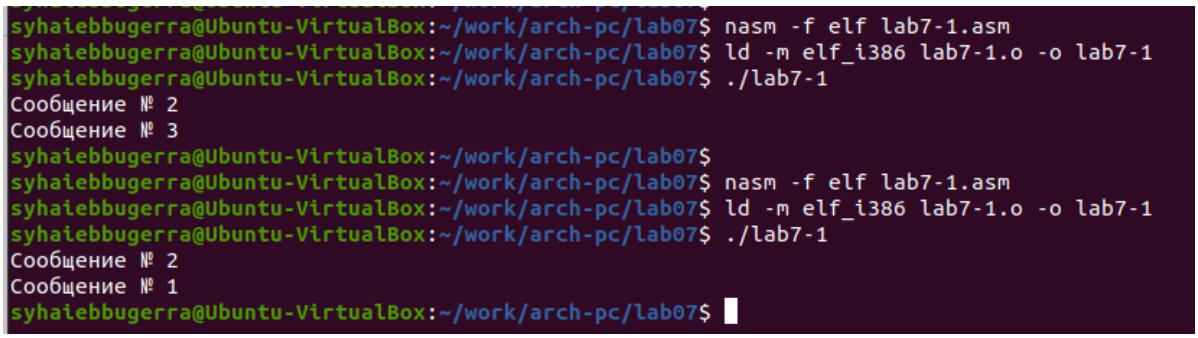

Рис. 2.4: Запуск программы lab7-1.asm

Изменил текст программы, изменив инструкции jmp, чтобы вывод программы был следующим:

```
Сообщение № 3
Сообщение № 2
Сообщение № 1
```

lab7-1.asm

```
%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
```

Рис. 2.5: Программа в файле lab7-1.asm

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

Рис. 2.6: Запуск программы lab7-1.asm

Использование инструкции jmp приводит к переходу в любом случае. Однако, часто при написании программ необходимо использовать условные переходы, т.е. переход должен происходить если выполнено какое-либо условие. В качестве примера рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C. Значения для A и C задаются в программе, значение B вводиться с клавиатуры.

Создал исполняемый файл и проверил его работу для разных значений B.

lab7-2.asm

```
7   section .bss
8   max resb 10
9   B resb 10
10  section .text
11  global _start
12  _start:
13  ; ---------- Вывод сообщения 'Введите B: '
14  mov eax,msg1
15  call sprint
16  ; ---------- Ввод 'B'
17  mov ecx,B
18  mov edx,10
19  call sread
20  ; ---------- Преобразование 'B' из символа в число
21  mov eax,B
22  call atoi
23  mov [B],eax
24  ; ---------- Записываем 'A' в переменную 'max'
25  mov ecx,[A]
26  mov [max],ecx
27  ; ---------- Сравниваем 'A' и 'C' (как символы)
28  cmp ecx,[C]
29  jg check_B
30  mov ecx,[C]
31  mov [max],ecx
32  ; ---------- Преобразование 'max(A,C)' из символа в число
33  check_B:
34  mov eax,max
35  call atoi
36  mov [max],eax
37  ; ---------- Сравниваем 'max(A,C)' и 'B' (как числа)
38  mov ecx,[max]
39  cmp ecx,[B]
40  jg fin
41  mov ecx,[B]
```

Рис. 2.7: Программа в файле lab7-2.asm

```
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 lab7-2.o -o lab7-2
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 20
Наибольшее число: 50
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 40
Наибольшее число: 50
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ ./lab7-2
Введите B: 60
Наибольшее число: 60
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 2.8: Запуск программы lab7-2.asm

2.2 Изучение структуры файлы листинга

Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ -l и задав имя файла листинга в командной строке.

Создал файл листинга для программы из файла lab7-2.asm

```
185    10                                  section .text
186    11                                  global _start
187    12                                  _start:
188    13                                  ; ---------- Вывод сообщения 'Введите B: '
189    14 000000E8 B8[00000000]            mov eax,msg1
190    15 000000ED E81DFFFFFF              call sprint
191    16                                  ; ---------- Ввод 'B'
192    17 000000F2 B9[0A000000]            mov ecx,B
193    18 000000F7 BA0A000000              mov edx,10
194    19 000000FC E842FFFFFF              call sread
195    20                                  ; ---------- Преобразование 'B' из символа в число
196    21 00000101 B8[0A000000]            mov eax,B
197    22 00000106 E891FFFFFF              call atoi
198    23 0000010B A3[0A000000]            mov [B],eax
199    24                                  ; ---------- Записываем 'A' в переменную 'max'
200    25 00000110 8B0D[35000000]          mov ecx,[A]
201    26 00000116 890D[00000000]          mov [max],ecx
202    27                                  ; ---------- Сравниваем 'A' и 'C' (как символы)
203    28 0000011C 3B0D[39000000]          cmp ecx,[C]
204    29 00000122 7F0C                    jg check_B
205    30 00000124 8B0D[39000000]          mov ecx,[C]
206    31 0000012A 890D[00000000]          mov [max],ecx
207    32                                  ; ---------- Преобразование 'max(A,C)' из символа в число
208    33                                  check_B:
209    34 00000130 B8[00000000]            mov eax,max
210    35 00000135 E862FFFFFF              call atoi
211    36 0000013A A3[00000000]            mov [max],eax
212    37                                  ; ---------- Сравниваем 'max(A,C)' и 'B' (как числа)
213    38 0000013F 8B0D[00000000]          mov ecx,[max]
214    39 00000145 3B0D[0A000000]          cmp ecx,[B]
215    40 0000014B 7F0C                    jg fin
```

Рис. 2.9: Файл листинга lab7-2

Внимательно ознакомился с его форматом и содержимым. Подробно объясню содержимое трёх строк файла листинга по выбору.

строка 192

- 17 - номер строки в подпрограмме
- 000000F2 - адрес
- B9[0A000000] - машинный код
- mov ecx,B - код программы

строка 193

- 18 - номер строки в подпрограмме
- 000000F7 - адрес
- BA0A000000 - машинный код
- mov edx,10 - код программы

строка 194

- 19 - номер строки в подпрограмме
- 000000FC - адрес
- E842FFFFFF - машинный код
- call sread - код программы

Открыл файл с программой lab7-2.asm и в инструкции с двумя операндами удалил один операнд. Выполнил трансляцию с получением файла листинга.

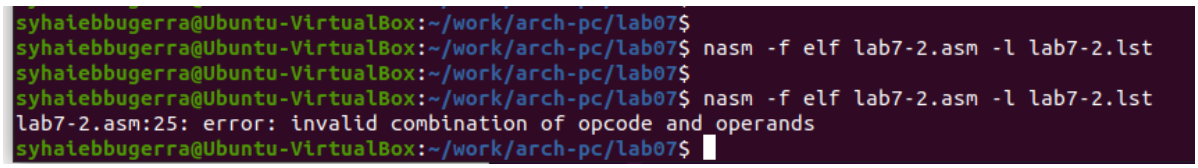

Рис. 2.10: Ошибка трансляции lab7-2

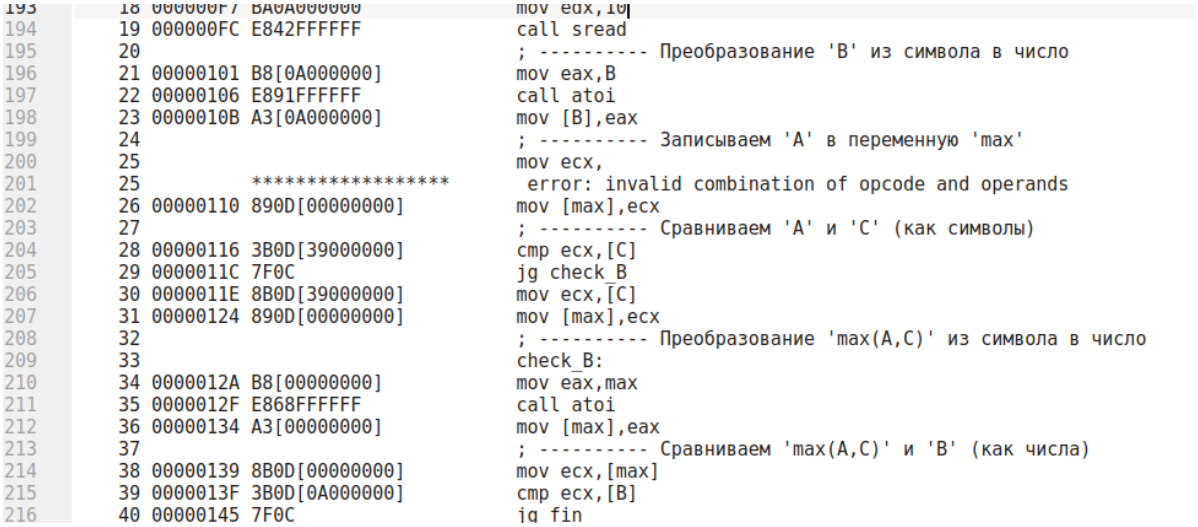

Рис. 2.11: Файл листинга с ошибкой lab7-2

Объектный файл не смог создаться из-за ошибки. Но получился листинг, где выделено место ошибки.

2.3 Задание для самостоятельной работы

Напишите программу нахождения наименьшей из 3 целочисленных переменных a,b и c. Значения переменных выбрать из табл. 7.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 6. Создайте исполняемый файл и проверьте его работу

для варианта 20 - 95,2,61

task7-1.asm

```
29      call sprint
30      mov ecx,B
31      mov edx,80
32      call sread
33      mov eax,B
34      call atoi
35      mov [B],eax
36
37      mov eax,msgC
38      call sprint
39      mov ecx,C
40      mov edx,80
41      call sread
42      mov eax,C
43      call atoi
44      mov [C],eax
45
46      mov ecx,[A]
47      mov [min],ecx
48
49      cmp ecx, [B]
50      jl check_C
51      mov ecx, [B]
52      mov [min], ecx
53
54  check_C:
55      cmp ecx, [C]
56      jl finish
57      mov ecx,[C]
58      mov [min],ecx
59
60  finish:
61      mov eax,answer
62      call sprint
63
64      mov eax, [min]
65      call iprintLF
```

Рис. 2.12: Программа в файле task7-1.asm

```
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf task7-1.asm
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 task7-1.o -o task7-1
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ ./task7-1
Input A: 95
Input B: 2
Input C: 61
Smallest: 2
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 2.13: Запуск программы task7-1.asm

Напишите программу, которая для введенных с клавиатуры значений x и a

вычисляет значение заданной функции f(x) и выводит результат вычислений. Вид функции f(x) выбрать из таблицы 7.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений X и a из 7.6.

для варианта 20

$$\begin{cases} x - a, x \geq a \\ 5, a < 0 \end{cases}$$

task7-2.asm

```
11  SECTION .text
12      GLOBAL _start
13
14  _start:
15      mov eax,msgA
16      call sprint
17      mov ecx,A
18      mov edx,80
19      call sread
20      mov eax,A
21      call atoi
22      mov [A],eax
23
24      mov eax,msgX
25      call sprint
26      mov ecx,X
27      mov edx,80
28      call sread
29      mov eax,X
30      call atoi
31      mov [X],eax
32
33      mov ebx, [X]
34      mov edx, [A]
35      cmp ebx, edx
36      jge first
37      jmp second
38
39  first:
40      mov eax,[X]
41      sub eax,[A]
42      call iprintLF
43      call quit
44  second:
45      mov eax,5
46      call iprintLF
47      call quit
```

Рис. 2.14: Программа в файле task7-2.asm

```
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ nasm -f elf task7-2.asm
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ ld -m elf_i386 task7-2.o -o task7-2
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ ./task7-2
Input A: 1
Input X: 2
1
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$ ./task7-2
Input A: 2
Input X: 1
5
syhaiebbugerra@Ubuntu-VirtualBox:~/work/arch-pc/lab07$
```

Рис. 2.15: Запуск программы task7-2.asm

3 Выводы

Изучили команды условного и безусловного переходов, познакомились с фалом листинга.