

Лабораторная работа №8.

Команды безусловного и условного переходов в Nasm.

Кучеренко София.

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Самостоятельная работа	9
4	Выводы	14

Список иллюстраций

2.1	Работа программы lab8-1	6
2.2	Работа программы lab8-2	6
2.3	Измененный код	7
2.4	Работа измененной программы	7
2.5	Результат выполнения программы	8
3.1	Код программы	10
3.2	Результат выполнения программы	11
3.3	Код программы	12
3.4	Результат выполнения программы	13

Список таблиц

1 Цель работы

Изучить команды условного и безусловного переходов и научиться писать программы с использованием этих переходов.

2 Выполнение лабораторной работы

1. Создадим файл lab8-1.asm, запишем код программы и проверим его работу:

```
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ touch lab8-1.asm
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ nasm -f elf lab8-1.asm
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ld -m elf_i386 -o lab8-1 lab8-1.o
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ./lab8-1
Сообщение No 2
Сообщение No 3
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $
```

Рис. 2.1: Работа программы lab8-1

2. Создадим файл lab8-2.asm, запишем код программы и также проверим его работу:

```
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ touch lab8-2.asm
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ nasm -f elf lab8-2.asm
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ./lab8-2
Сообщение No 2
Сообщение No 1
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $
```

Рис. 2.2: Работа программы lab8-2

3. Изменим текст программы, изменив инструкции jmp и получим следующее:

```

1  %include 'in_out.asm' ; подключение внешнего файла
2  SECTION .data
3  msg1: DB 'Сообщение No 1',0
4  msg2: DB 'Сообщение No 2',0
5  msg3: DB 'Сообщение No 3',0
6  SECTION .text
7  GLOBAL _start
8  _start:
9  jmp _label3
10 _label1:
11 mov eax, msg1 ; Вывод на экран строки
12 call sprintf ; 'Сообщение No 1'
13 jmp _end
14 _label2:
15 mov eax, msg2 ; Вывод на экран строки
16 call sprintf ; 'Сообщение No 2'
17 jmp _label1
18 _label3:
19 mov eax, msg3 ; Вывод на экран строки
20 call sprintf ; 'Сообщение No 3'
21 jmp _label2
22 _end:
23 call quit ; вызов подпрограммы завершения

```

Рис. 2.3: Измененный код

```

• smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ nasm -f elf lab8-2.asm
• smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ld -m elf_i386 -o lab8-2 lab8-2.o
• smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ./lab8-2
Сообщение No 3
Сообщение No 2
Сообщение No 1

```

Рис. 2.4: Работа измененной программы

4. Создадим файл lab8-3.asm, запишем код программы и проверим его работу на разных значениях В:

```
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ touch lab8-3.asm
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ nasm -f elf lab8-3.asm
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ./lab8-3
Введите В: 3
Наибольшее число: 50
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ./lab8-3
Введите В: 5
Наибольшее число: 50
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ./lab8-3
Введите В: 55
Наибольшее число: 55
```

Рис. 2.5: Результат выполнения программы

3 Самостоятельная работа

1. Напишем программу нахождения наименьшей из 3 целочисленных переменных и запустим её:

```

1  %include 'in_out.asm'
2
3  section .data
4      msg1 db "Наименьшее число:"
5      a dd 32
6      b dd 46
7      c dd 74
8
9  section .bss
10     min resb 10
11
12  section .text
13  global _start
14
15  _start:
16      mov eax, msg1
17      call sprint
18
19      mov ecx, [a]
20      mov [min], ecx ; 'min = A'
21      ; ----- Сравниваем 'A' и 'C' (как числа)
22      cmp ecx, [c] ; Сравниваем 'A' и 'C'
23      jl check_B ; если 'A<C', то переход на метку 'check_B',
24      mov ecx, [c] ; иначе 'ecx = C'
25      mov [min], ecx ; 'min = C'
26      ; ----- Преобразование 'min(A,C)' из символа в число
27
28  check_B:
29      ; ----- Сравниваем 'min(A,C)' и 'B' (как числа)
30      mov ecx, [min]
31      cmp ecx, [b] ; Сравниваем 'min(A,C)' и 'B'
32      jl fin ; если 'min(A,C)>B', то переход на 'fin',
33      mov ecx, [b] ; иначе 'ecx = B'
34
35      mov [min], ecx
36
37      ; ----- Вывод результата
38  fin:
39      mov eax, [min]
40      call iprintLF ; Вывод 'min(A,B,C)'
41      call quit ; Выход

```

Рис. 3.1: Код программы

```
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ nasm -f elf lab8-3.asm
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ld -m elf_i386 -o lab8-3 lab8-3.o
smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ./lab8-3
Наименьшее число: 32
```

Рис. 3.2: Результат выполнения программы

2. Напишем программу для нахождения значения заданной функции из введенных с клавиатуры значений:

```

_start:
; ----- Ввод 'X'
mov eax, msgX
call sprint
mov ecx, x
mov edx, 10
call sread

; ----- Ввод 'A'
mov eax, msgA
call sprint
mov ecx, a
mov edx, 10
call sread

; ----- Преобразование 'x' из символа в число
mov eax, x
call atoi
mov [x], eax

; ----- Преобразование 'a' из символа в число
mov eax, a
call atoi
mov [a], eax

mov ecx, [x]
cmp ecx, [a]

ja newfunc

mov eax, [x]
jmp fin

newfunc:
mov eax, [a]
mov ebx, [x]
add eax, ebx

fin:
call iprintLF
call quit

```

Рис. 3.3: Код программы

```
• smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ./lab8-4
x = 3
a = 2
5
• smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $ ./lab8-4
x = 4
a = 5
4
○ smkucherenko@dk3n66 ~/work/koshechki/study_2022-2023_arh-pc/labs/lab08 $
```

Рис. 3.4: Результат выполнения программы

4 Выводы

Я изучила команды условного и безусловного переходов и научилась писать программы с использованием этих переходов.