

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук  
Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина:   Архитектура компьютера

Студент: Мелкумян Арвин

Группа: НКАбд-04-23

МОСКВА

2023 г.

## КОМАНДЫ БЕЗУСЛОВНОГО И УСЛОВНОГО ПЕРЕХОДОВ В NASM. ПРОГРАММИРОВАНИЕ ВЕТВЛЕНИЙ.

**Цель работы:** Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

### Ход работы.

Создадим программу для реализации безусловного перехода, исходный код которой показан на рисунке 1.

```
1  include 'in_out.asm'
2  SECTION .data
3  msg1: DB 'Сообщение №1',0
4  msg2: DB 'Сообщение №2',0
5  msg3: DB 'Сообщение №3',0
6
7  SECTION .text
8  GLOBAL _start
9  _start:
10     jmp _label2
11
12     _label1:
13     mov eax, msg1
14     call sprintf
15
16     _label2:
17     mov eax, msg2
18     call sprintf
19
20     _label3:
21     mov eax, msg3
22     call sprintf
23
24     _end:
25     call quit
```

Рисунок 1 — Исходный код программы

Как видно из кода, после старта выполнение с помощью команды `jmp` переходит к метке `label2` и оттуда продолжается до завершения. Результат работы программы показан на рисунке 2.

```
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ touch lab7-1.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ nvim lab7-1.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ nasm -f elf lab7-1.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ./lab7-1
Сообщение №2
Сообщение №3
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$
```

Рисунок 2 — Результат работы программы

Изменим программу таким образом, чтобы сначала выводила второе сообщение, потом первое, потом завершала работу. Исходный код такой программы показан на рисунке 3, а результат выполнения на рисунке 4.

```
20 %include 'in_out.asm'
19
18 SECTION .data
17 msg1: DB 'Сообщение №1',0
16 msg2: DB 'Сообщение №2',0
15 msg3: DB 'Сообщение №3',0
14
13 SECTION .text
12 GLOBAL _start
11 _start:
10     jmp _label2
9
8     _label1:
7     mov eax, msg1
6     call sprintfLF
5     jmp _end
4
3     _label2:
2     mov eax, msg2
1     call sprintfLF
21    jmp _label1
1
2     _label3:
3     mov eax, msg3
4     call sprintfLF
5
6     _end:
7     call quit
~
```

Рисунок 3 — Исходный код второй версии программы

```
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ nasm -f elf lab7-1.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ./lab7-1
Сообщение №2
Сообщение №1
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$
```

Рисунок 4 — Результат выполнения программы

Наконец, изменим программу, чтобы она выводила сообщения в обратном

порядке. Скорректированный исходный код показан на рисунке 5. Результат выполнения на рисунке 6.

```
17 %include 'in_out.asm'
16
15 SECTION .data
14 msg1: DB 'Сообщение №1',0
13 msg2: DB 'Сообщение №2',0
12 msg3: DB 'Сообщение №3',0
11
10 SECTION .text
9 GLOBAL _start
8 _start:
7 jmp _label3
6
5 _label1:
4 mov eax, msg1
3 call sprintf
2 jmp _end
1
18 _label2:
1 mov eax, msg2
2 call sprintf
3 jmp _label1
4
5 _label3:
6 mov eax, msg3
7 call sprintf
8 jmp _label2
9
10 _end:
11 call quit
```

Рисунок 5 — Исходный файл третьей версии программы

```
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ nasm -f elf lab7-1.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ld -m elf_i386 lab7-1.o -o lab7-1
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ./lab7-1
Сообщение №3
Сообщение №2
Сообщение №1
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ █
```

Рисунок 6 — Результат выполнения программы

Далее создадим программу, которая из трех чисел (двух явно заданных в программе и одного вводимого пользователем) найдем наибольшее. Исходный код программы показан на рисунке 7, результат выполнения на рисунке 8

```

44 %include 'in_out.asm'
43
42 section .data
41     msg1 db 'Введите B: ',0h
40     msg2 db 'Наибольшее число: ',0h
39     A dd '20'
38     C dd '50'
37
36 section .bss
35     max resb 10
34     B resb 10
33
32 section .text
31     global _start
30
29 _start:
28     mov eax, msg1
27     call sprintf
26
25     mov ecx, B
24     mov edx, 10
23     call sread
22
21     mov eax, B
20     call atoi
19     mov [B], eax
18
17     mov ecx, [A]
16     mov [max], ecx
15
14     cmp ecx, [C]
13     jg check_B
12     mov ecx, [C]
11     mov [max], ecx
10
9  check_B:
8     mov eax, max
7     call atoi
6     mov [max], eax
5
4     mov ecx, [max]
3     cmp ecx, [B]
2     jg fin
1     mov ecx, [B]
45    mov [max], ecx
1
2  fin:
3     mov eax, msg2
4     call sprintf
5     mov eax, [max]
6     call iprintLF
7     call quit

```

Рисунок 7 — Исходный код программы lab7-2

```

infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ touch lab7-2.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ nvim lab7-2.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ nasm -f elf lab7-2.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ld -m elf_i386 lab7-2.o -o lab7-2
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ./lab7-2
Введите B: 100
Наибольшее число: 100
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ █

```

Рисунок 8 — Результат выполнения программы lab7-2

Указав при компиляции ключ `l` и имя файла получим файл листинга

программы, пример которого для программы lab7-2 представлен на рисунке 9.

```

139 000000B1 80FB30      <l>    cmp     bl, 48
140 000000B4 7C14      <l>    jl     .finished
141 000000B6 80FB39      <l>    cmp     bl, 57
142 000000B9 7F0F      <l>    jg     .finished
143          <l>
144 000000BB 80EB30      <l>    sub     bl, 48
145 000000BE 01D8      <l>    add     eax, ebx
146 000000C0 B80A000000 <l>    mov     ebx, 10
147 000000C5 F7E3      <l>    mul     ebx
148 000000C7 41        <l>    inc     ecx
149 000000C8 EBE2      <l>    jmp     .multiplyLoop
150          <l>
151          <l> .finished:
152 000000CA 83F900      <l>    cmp     ecx, 0
153 000000CD 7407      <l>    je     .restore
154 000000CF B80A000000 <l>    mov     ebx, 10
155 000000D4 F7F3      <l>    div     ebx
156          <l>
157          <l> .restore:
158 000000D6 5E        <l>    pop     esi
159 000000D7 5A        <l>    pop     edx
160 000000D8 59        <l>    pop     ecx
161 000000D9 5B        <l>    pop     ebx
162 000000DA C3        <l>    ret
163          <l>
164          <l>
165          <l> ;----- quit -----
166          <l> ; Функция завершения программы
167          <l> quit:
168 000000DB B800000000 <l>    mov     ebx, 0
169 000000E0 B801000000 <l>    mov     eax, 1
170 000000E5 CD80      <l>    int     80h
171 000000E7 C3        <l>    ret
2
3
4 00000000 D092D0B2D0B5D0B4D0-      section .data
4 00000009 B8D182D0B520423A20-      msg1 db 'Введите B: ',0h
4 00000012 00
5 00000013 D09DD0B0D0B8D0B1D0-      msg2 db 'Наибольшее число: ',0h
5 0000001C BE00BBD18CD188D0B5-
5 00000025 D0B520D187D0B8D181-
5 0000002E D0BBD0BE3A2000
6 00000035 32300000      A dd '20'
7 00000039 35300000      C dd '50'
8
9
10 00000000 <res Ah>      section .bss
11 0000000A <res Ah>      max resb 10
12                                B resb 10
13
14                                section .text
15                                global _start
16
17 000000E8 B8[00000000]      _start:
18 000000ED E81DFFFFFF      mov eax, msg1
19                                call sprint

```

Рисунок 9 — Файл листинга программы

Данный файл содержит машинный код, адрес памяти и исходный код программы. К примеру, в строках 3-5 содержится объявление данных для выводимых сообщений. В случае ошибки в исходном коде в файле листинга

отображается ошибка в соответствующем месте файла.

**Задание 1.** Напишите программу нахождения наименьшей из 3 целочисленных переменных  $a, b$  и  $c$ . Значения переменных соответствии с вариантом 17 равны 26; 12 и 68. Создайте исполняемый файл и проверьте его работу.

Решение.

Исходный код программы показан на рисунке 10

```
1  include 'in_out.asm'
2  section .data
3      msg2 db 'Наименьшее число: ',0h
4      A dd 26
5      B dd 12
6      C dd 68
7
8  section .bss
9      min resb 10
10
11 section .text
12     global _start
13
14 _start:
15     mov ecx, [A]
16     mov [min], ecx
17
18     cmp ecx, [C]
19     jl check_B
20     mov ecx, [C]
21     mov [min], ecx
22
23 check_B:
24     mov ecx, [min]
25     cmp ecx, [B]
26     jl fin
27     mov ecx, [B]
28     mov [min], ecx
29
30 fin:
31     mov eax, msg2
32     call sprint
33     mov eax, [min]
34     call iprintLF
35     call quit
```

Рисунок 10 — Исходный код программы lab7-3

В отличие от представленной выше программы 7-2, здесь мы сразу работаем с числами и не вводим данные из консоли, что несколько упрощает исходный код.

Результат выполнения программы показан на рисунке 11.

```
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ nvim lab7-3.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ nasm -f elf lab7-3.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ld -m elf_i386 lab7-3.o -o lab7-3
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ./lab7-3
Наименьшее число: 12
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ █
```

Рисунок 11 — Результат выполнения программы lab7-3

**Задание 2.** 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 (вариант 17). Создайте исполняемый файл и проверьте его работу для значений  $x$  и  $a$  из 7.6.

**Решение.**

Исходный код программы показан на рисунке 12.



```

1  #include 'in_out.asm'
2  section .data
3      msg1 db 'Введите x: ',0h
4      msg2 db 'Введите a: ',0h
5      msg3 db 'Результат: ',0h
6
7  section .bss
8      result resb 10
9      X resb 10
10     A resb 10
11
12 section .text
13     global _start
14
15 _start:
16     mov eax, msg1
17     call sprintf
18     mov ecx, X
19     mov edx, 10
20     call sread
21     mov eax, X
22     call atoi
23     mov [X], eax
24
25     mov eax, msg2
26     call sprintf
27     mov ecx, A
28     mov edx, 10
29     call sread
30     mov eax, A
31     call atoi
32     mov [A], eax
33
34     cmp eax, 8
35     jl _calc_first
36
37     mov eax, [X]
38     mov ebx, [A]
39     mul ebx
40     mov [result], eax
41
42     jmp _end
43
44 _calc_first:
45     mov eax, [A]
46     add eax, 8
47     mov [result], eax
48
49     jmp _end
50
51 _end:
52     mov eax, msg3
53     call sprintf
54     mov eax, [result]
55     call iprintLF
56
57     call quit
58

```

Рисунок 12 — Исходный код программы 7-4

На рисунке 13 показан результат выполнения программы с предложенными, согласно варианту, исходными данными.

```
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ nasm -f elf lab7-4.asm
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ld -m elf_i386 lab7-4.o -o lab7-4
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ./lab7-4
Введите x: 3
Введите a: 4
Результат: 12
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$ ./lab7-4
Введите x: 2
Введите a: 9
Результат: 18
infer@Cameron:~/work/study/2023-2024/Архитектура компьютера/study_2023-2024_arh-pc/labs/lab07$
```

Рисунок 13 — Результат выполнения программы

Как нетрудно заметить, результат соответствует правильному ответу.

**Выводы:** В ходе лабораторной работы были изучены структуры безусловного и условного перехода в NASM.