

Лабараторная работа №07.

НПИбд-03-24

Подготовил:

Гелдиев Ыхлас. Студенческий номер: 1032249184

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
2.1	Реализация переходов в NASM	6
2.2	Изучение структуры файлы листинга	12
3	Задание для самостоятельной работы	15
4	Выводы	20

Список иллюстраций

2.1	Создал lab7-1.asm	6
2.2	Заполнил lab7-1	7
2.3	Запуск lab7-1	7
2.4	Изменил текст lab7-1	8
2.5	Запуск измененного lab7-1	8
2.6	Выполнение мини задания	9
2.7	Проверка выполнения	9
2.8	Создание lab7-2	10
2.9	Заполнение lab7-2	11
2.10	Запуск lab7-2	12
2.11	Компиляция с листингом	13
2.12	Ошибка в lab7-2	14
2.13	Измененный листинг	14
3.1	Программа hw1	16
3.2	Запуск hw1	17
3.3	Программа hw2	18
3.4	Запуск hw2	19

Список таблиц

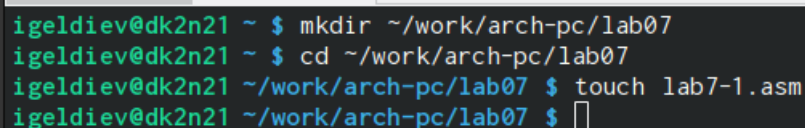
1 Цель работы

Изучение команд для условных и безусловных переходов. Освоение навыков программирования с применением переходов. Ознакомление с назначением и структурой файла с листингом.

2 Выполнение лабораторной работы

2.1 Реализация переходов в NASM

1. Создал каталог для программ, перешел в него и создал файл lab7-1.asm (рис. 2.1)



```
igeldiev@dk2n21 ~ $ mkdir ~/work/arch-pc/lab07
igeldiev@dk2n21 ~ $ cd ~/work/arch-pc/lab07
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ touch lab7-1.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $
```

Рис. 2.1: Создал lab7-1.asm

2. Заполнил lab7-1.asm (рис. 2.2)

```

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

```

Рис. 2.2: Заполнил lab7-1

3. Создал исполняемый файл и запустил его (рис. 2.3)

```

igeldiev@dk2n21 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 3
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ █

```

Рис. 2.3: Запуск lab7-1

4. Изменил текст программы (рис. 2.4)

```

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1: DB 'Сообщение № 1', 0
5 msg2: DB 'Сообщение № 2', 0
6 msg3: DB 'Сообщение № 3', 0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11     jmp _label2
12
13     _label1:
14         mov eax, msg1
15         call sprintLF
16         jmp _end
17
18     _label2:
19         mov eax, msg2
20         call sprintLF
21         jmp _label1
22
23     _label3:
24         mov eax, msg3
25         call sprintLF
26
27     _end:
28         call quit

```

Рис. 2.4: Изменил текст lab7-1

5. Создал исполняемый файл и запустил его (рис. 2.5)

```

igeldiev@dk2n21 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 1
igeldiev@dk2n21 ~/work/arch-pc/lab07 $

```

Рис. 2.5: Запуск измененного lab7-1

6. Изменил текст программы так, что бы он соответствовал выводу (рис. 2.6)


```

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1: DB 'Сообщение № 1', 0
5 msg2: DB 'Сообщение № 2', 0
6 msg3: DB 'Сообщение № 3', 0
7
8 SECTION .text
9 GLOBAL _start
10 _start:
11     jmp _label3
12
13     _label1:
14         mov eax, msg1
15         call sprintf
16         jmp _end
17
18     _label2:
19         mov eax, msg2
20         call sprintf
21         jmp _label1
22
23     _label3:
24         mov eax, msg3
25         call sprintf
26         jmp _label2
27
28     _end:
29         call quit

```

Рис. 2.6: Выполнение мини задания

7. Проверка выполнения (рис. 2.7)

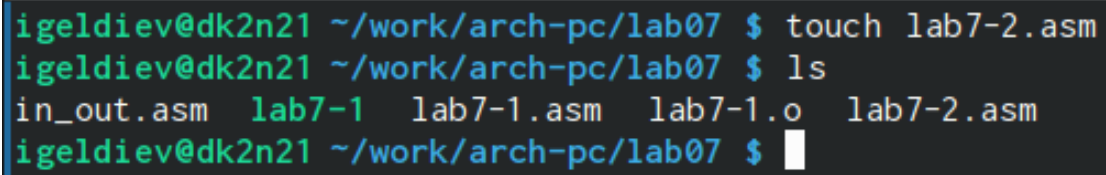
```

igeldiev@dk2n21 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
igeldiev@dk2n21 ~/work/arch-pc/lab07 $

```

Рис. 2.7: Проверка выполнения

8. Создал файл lab7-2.asm (рис. 2.8)

A terminal window with a dark background and light-colored text. The prompt is 'igeldiev@dk2n21' and the current directory is '~/work/arch-pc/lab07'. The user enters the command 'touch lab7-2.asm'. The next line shows the command 'ls' and the output listing files: 'in_out.asm', 'lab7-1', 'lab7-1.asm', 'lab7-1.o', and 'lab7-2.asm'. The prompt is shown again at the end of the line.

```
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ touch lab7-2.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ls
in_out.asm  lab7-1  lab7-1.asm  lab7-1.o  lab7-2.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $
```

Рис. 2.8: Создание lab7-2

9. Заполнил lab7-2.asm (рис. 2.9)

```

1 %include 'in_out.asm'
2 section .data
3     msg1 db 'Введите B: ',0h
4     msg2 db '', 0h
5     A dd '20'
6     C dd '50'
7
8 section .bss
9     max resb 10
10    B resb 10
11
12 section .text
13 global _start
14 _start:
15 ; ----- mov Вывод сообщения 'Введите B: '
16 mov eax,msg1
17 call sprint
18 ; ----- Ввод 'B'
19 mov ecx,B
20 mov edx,10
21 call sread
22 ; ----- Преобразование 'B' из символа в число
23 mov eax,B
24 call atoi ; Вызов подпрограммы перевода символа в число
25 mov [B],eax ; запись преобразованного числа в 'B'
26 ; ----- Записываем 'A' в переменную 'max'
27 mov ecx,[A] ; 'ecx = A'
28 mov [max],ecx ; 'max = A'
29 ; ----- Сравниваем 'A' и 'C' (как символы)
30 cmp ecx,[C]; Сравниваем 'A' и 'C'
31 jg check_B; если 'A>C', то переход на метку 'check_B',
32 mov ecx,[C]; иначе 'ecx = C'
33 mov [max],ecx ; 'max = C'
34 ; ----- Преобразование 'max(A,C)' из символа в число
35 check_B:
36 mov eax,max
37 call atoi ; Вызов подпрограммы перевода символа в число
38 mov [max],eax ; запись преобразованного числа в 'max'
39 ; ----- Сравниваем 'max(A,C)' и 'B' (как числа)
40 mov ecx,[max]
41 cmp ecx,[B]; Сравниваем 'max(A,C)' и 'B'
42 jg fin; если 'max(A,C)>B', то переход на 'fin',
43 mov ecx,[B]; иначе 'ecx = B'
44 mov [max],ecx
45 ; ----- Вывод результата
46 fin:
47 mov eax,msg2

```

Рис. 2.9: Заполнение lab7-2

10. Создал исполняемый файл и запустил его (рис. 2.10)

```
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ nasm -f elf lab7-2.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-2 lab7-2.o
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ./lab7-2
Введите B: 12
50
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ./lab7-2
Введите B: 56
56
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ./lab7-2
Введите B: -12
50
igeldiev@dk2n21 ~/work/arch-pc/lab07 $
```

Рис. 2.10: Запуск lab7-2

2.2 Изучение структуры файлы листинга

11. Создал файл листинга для программы из файла lab7-2.asm и открыл его (рис. 2.11)

```

igeldiev@dk2n21 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ tail -n 100000000 lab7-2.lst
1          %include 'in_out.asm'
2          <1> ;----- slen -----
3          <1> ; Функция вычисления длины сообщения
4          <1> slen:
5          4 00000000 53          <1>     push    ebx
6          5 00000001 89C3       <1>     mov     ebx, eax
7          6          <1>
8          7          <1> nextchar:
9          8 00000003 803800     <1>     cmp     byte [eax], 0
10         9 00000006 7403       <1>     jz      finished
11        10 00000008 40         <1>     inc     eax
12        11 00000009 EBF8       <1>     jmp     nextchar
13        12          <1>
14        13          <1> finished:
15        14 0000000B 29D8       <1>     sub     eax, ebx
16        15 0000000D 5B         <1>     pop     ebx
17        16 0000000E C3         <1>     ret
18        17          <1>
19        18          <1>
20        19          <1> ;----- sprint -----
21        20          <1> ; Функция печати сообщения
22        21          <1> ; входные данные: mov eax,<message>
23        22          <1> sprint:
24        23 0000000F 52         <1>     push    edx
25        24 00000010 51         <1>     push    ecx
26        25 00000011 53         <1>     push    ebx
27        26 00000012 50         <1>     push    eax
28        27 00000013 E8E8FFFF   <1>     call    slen
29        28          <1>
30        29 00000018 89C2       <1>     mov     edx, eax
31        30 0000001A 58         <1>     pop     eax
32        31          <1>
33        32 0000001B 89C1       <1>     mov     ecx, eax
34        33 0000001D BB01000000 <1>     mov     ebx, 1
35        34 00000022 B804000000 <1>     mov     eax, 4
36        35 00000027 CD80       <1>     int     80h
37        36          <1>
38        37 00000029 5B         <1>     pop     ebx
39        38 0000002A 59         <1>     pop     ecx
40        39 0000002B 5A         <1>     pop     edx
41        40 0000002C C3         <1>     ret
42        41          <1>
43        42          <1>
44        43          <1> ;----- sprintf -----
45        44          <1> ; Функция печати сообщения с переводом строки
46        45          <1> ; входные данные: mov eax,<message>
47        46          <1> sprintf:

```

Рис. 2.11: Компиляция с листингом

12. Изменил файл lab7-2 (рис. 2.12)

```

7
8 section .bss
9     max resb 10
10    B resb 10
11
12 section .text
13 global _start
14 _start:
15 ; ----- mov Вывод сообщения 'Введите B: '
16 mov eax
17 call sprint
18 ; ----- Ввод 'B'
19 mov ecx,B
20 mov edx,10
21 call sread
22 ; ----- Преобразование 'B' из символа в число
23 mov eax,B
24 call atoi ; Вызов подпрограммы перевода символа в число

```

Рис. 2.12: Ошибка в lab7-2

13. Создание файла листинга для измененной программы и нахождение изменений (рис. 2.13)

```

7
8                                section .bss
9 00000000 <res Ah>              max resb 10
10 0000000A <res Ah>             B resb 10
11
12                                section .text
13                                global _start
14                                _start:
15                                ; ----- mov Вывод сообщения 'Введите B: '
16                                mov eax
16                                ***** error: invalid combination of opcode and operands
17 000000E8 E822FFFFFF            call sprint
18                                ; ----- Ввод 'B'
19 000000ED B9[0A000000]          mov ecx,B
20 000000F2 BA0A000000            mov edx,10
21 000000F7 E847FFFFFF            call sread

```

Рис. 2.13: Измененный листинг

3 Задание для самостоятельной работы

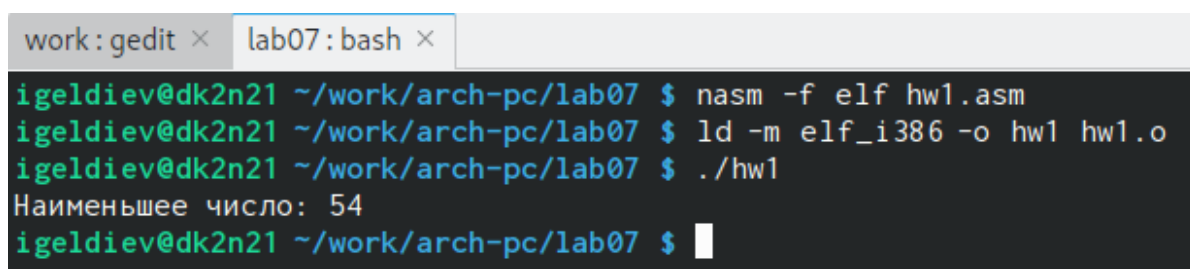
1. Программа для нахождения наименьшего из 3 целочисленных переменных (рис. 3.1) (рис. 3.2)

```

1 %include 'in_out.asm'
2 section .data
3     msg db 'Наименьшее число: ', 0h
4     A dd 54
5     B dd 62
6     C dd 87
7
8 section .bss
9     min resb 10
10
11 section .text
12 global _start
13 _start:
14     ; ----- Записываем 'A' в переменную 'min'
15     mov ecx,[A]      ; 'ecx = A'
16     mov [min],ecx    ; 'min = A'
17     ; ----- Сравниваем 'A' и 'C' (как символы)
18     cmp ecx,[C]      ; Сравниваем 'A' и 'C'
19     jnl check_B      ; если 'A<C', то переход на метку 'check_B'
20     mov ecx,[C]      ; иначе 'ecx = C'
21     mov [min],ecx    ; 'min = C'
22 check_B:
23     ; ----- Сравниваем 'min(A,C)' и 'B'
24     mov ecx,[min]
25     cmp ecx,[B]      ; Сравниваем 'min(A,C)' и 'B'
26     jnl fin          ; если 'min(A,C)<B', то переход на 'fin'
27     mov ecx,[B]      ; иначе 'ecx = B'
28     mov [min],ecx
29     ; ----- Вывод результата
30 fin:
31     mov eax, msg
32     call sprint      ; Вывод сообщения 'Наименьшее число: '
33     mov eax, [min]
34     call iprintLF    ; Вывод 'min(A,B,C)'
35     call quit        ; Выход
36

```

Рис. 3.1: Программа hw1



```
work: gedit × lab07: bash ×
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ nasm -f elf hw1.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o hw1 hw1.o
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ./hw1
Наименьшее число: 54
igeldiev@dk2n21 ~/work/arch-pc/lab07 $
```

Рис. 3.2: Запуск hw1

2. Программа для вычислений по формулам $2(x-a)$ при $x > a$ и 15, при $x \leq a$ (Вариант №5) (рис. 3.3) (рис. 3.4)

```

2 SECTION .data
3 msg1 db "Результат: ",0h
4 msg2 db "Введите x: ",0h
5 msg3 db "Введите a: ",0h
6 SECTION .bss
7 X resb 10
8 Y resb 10
9 result resb 10
10 SECTION .text
11 global _start
12 _start:
13     mov eax,msg2
14     call sprint
15     mov ecx,X
16     mov edx,10
17     call sread
18     mov eax,X
19     call atoi
20     mov [X],eax
21     mov eax,msg3
22     call sprint
23     mov ecx,Y
24     mov edx,10
25     call sread
26     mov eax,Y
27     call atoi
28     mov [Y],eax
29     cmp [X], eax
30     jg xgreatery
31     mov eax, 15
32     mov [result], eax
33     jmp printresult
34 xgreatery:
35     mov eax, [X]
36     mov ebx, [Y]
37     sub eax,ebx
38     mov [X], eax
39     mov eax, [X]
40     mov edx, 2
41     mul edx
42     mov [result], eax
43 printresult:
44     mov eax, msg1
45     call sprint
46     mov eax, [result]
47     call iprintLF
48     call quit

```

Рис. 3.3: Программма hw2

```
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ nasm -f elf hw2.asm
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o hw2 hw2.o
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ./hw2
Введите x: 1
Введите a: 2
Результат: 15
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ ./hw2
Введите x: 2
Введите a: 1
Результат: 2
igeldiev@dk2n21 ~/work/arch-pc/lab07 $ █
```

Рис. 3.4: Запуск hw2

4 Выводы

Я изучил команды для условных и безусловных переходов. Я освоил навыки программирования с применением переходов и ознакомился с назначением и структурой файла с листингом.