

Отчёт по Лабораторной работе №7

Дисциплина: Архитектура компьютера

Гозенко А.С.

Содержание

1	Цель работы	4
2	Выполнение лабораторной работы	5
3	Выполнение задания для самостоятельной работы. Вариант 13.	11
4	Выводы	14

Список иллюстраций

2.1	Создание файла	5
2.2	Программа	5
2.3	Работа программы	6
2.4	Программа	6
2.5	Работа программы	6
2.6	Программа	7
2.7	Работа программы	7
2.8	Программа	8
2.9	Работа программы	8
2.10	Создание файла листинга	9
2.11	Ошибка	9
2.12	Описание ошибки	10
3.1	Код программы	11
3.2	Работа программы	12
3.3	Код программы	12
3.4	Работа программы	13

1 Цель работы

Изучение команд условного и безусловного переходов ассемблера NASM. Приобретение навыков написания программ с использованием переходов.

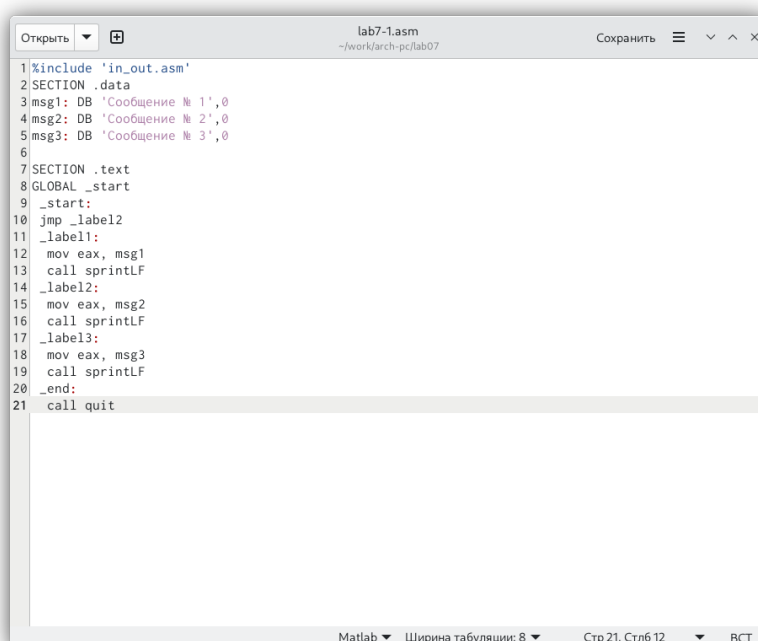
2 Выполнение лабораторной работы

Создание каталога для лабораторной работы, переход в него и создание файла lab7-1.asm (рис. 2.1).

```
asgozenko@dk8n81 ~ $ mkdir ~/work/arch-pc/lab07
asgozenko@dk8n81 ~ $ cd ~/work/arch-pc/lab07
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ touch lab7-1.asm
asgozenko@dk8n81 ~/work/arch-pc/lab07 $
```

Рис. 2.1: Создание файла

Написание программы из листинга 7.1 (рис. 2.2).



```
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 _label1:
12 mov eax, msg1
13 call sprintf
14 _label2:
15 mov eax, msg2
16 call sprintf
17 _label3:
18 mov eax, msg3
19 call sprintf
20 _end:
21 call quit
```

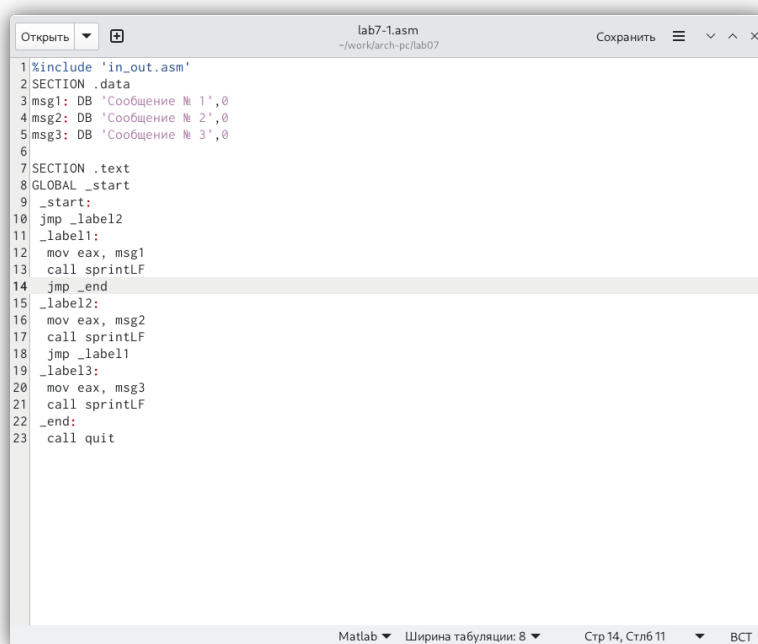
Рис. 2.2: Программа

Работа программы из листинга 7.1(рис. 2.3).

```
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 3
asgozenko@dk8n81 ~/work/arch-pc/lab07 $
```

Рис. 2.3: Работа программы

Изменение программы из листинга 7.1(рис. 2.4).



```
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 _label1:
12 mov eax, msg1
13 call sprintf
14 jmp _end
15 _label2:
16 mov eax, msg2
17 call sprintf
18 jmp _label1
19 _label3:
20 mov eax, msg3
21 call sprintf
22 _end:
23 call quit
```

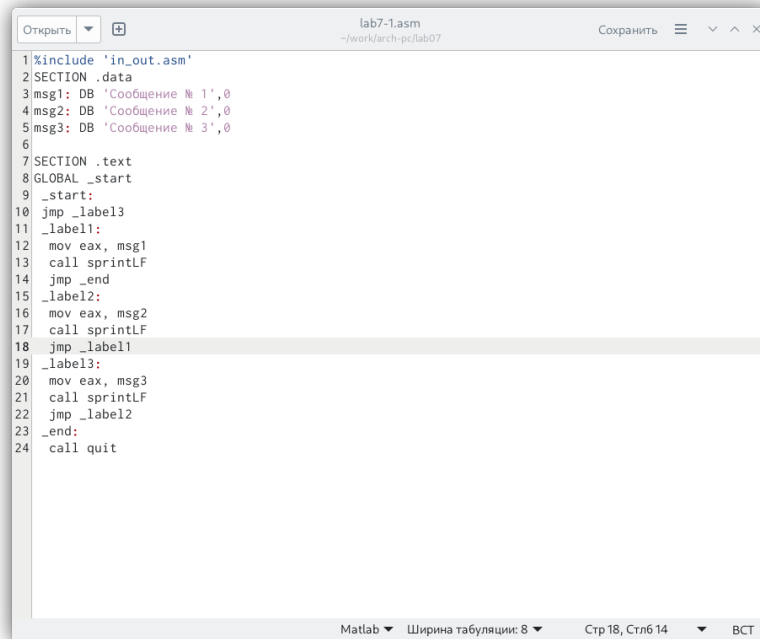
Рис. 2.4: Программа

Работа изменённой программы(рис. 2.5).

```
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 1
asgozenko@dk8n81 ~/work/arch-pc/lab07 $
```

Рис. 2.5: Работа программы

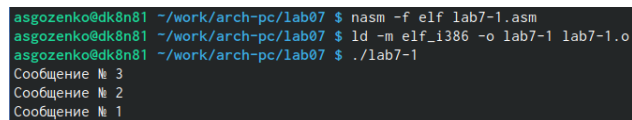
Написание программы из листинга 7.2(рис. 2.6).



```
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 _label3
11 _label1:
12 mov eax, msg1
13 call sprintf
14 jmp _end
15 _label2:
16 mov eax, msg2
17 call sprintf
18 jmp _label1
19 _label3:
20 mov eax, msg3
21 call sprintf
22 jmp _label2
23 _end:
24 call quit
```

Рис. 2.6: Программа

Работа программы из листинга 7.2(рис. 2.7).



```
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
```

Рис. 2.7: Работа программы

Написание программы lab7-2.asm из Листинга 7.3(рис. 2.8).

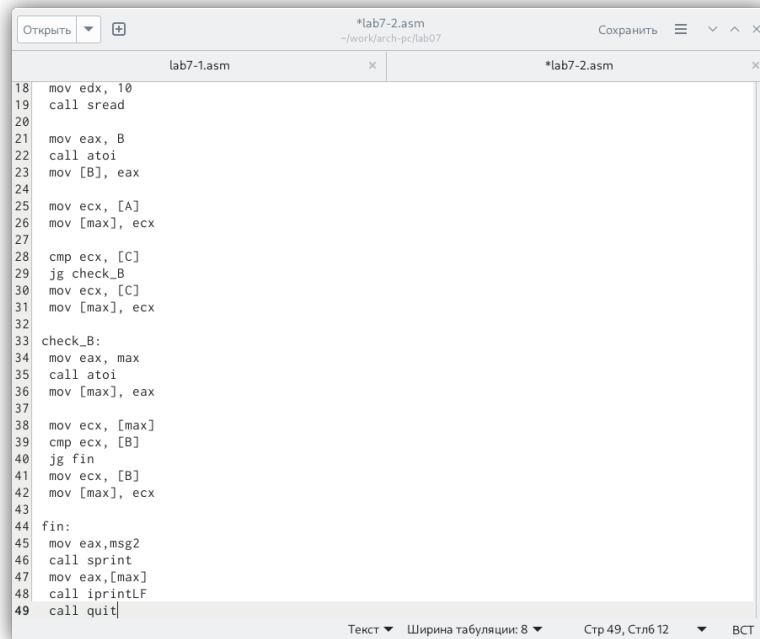


Рис. 2.8: Программа

Создание файла и работа программы(рис. 2.9).

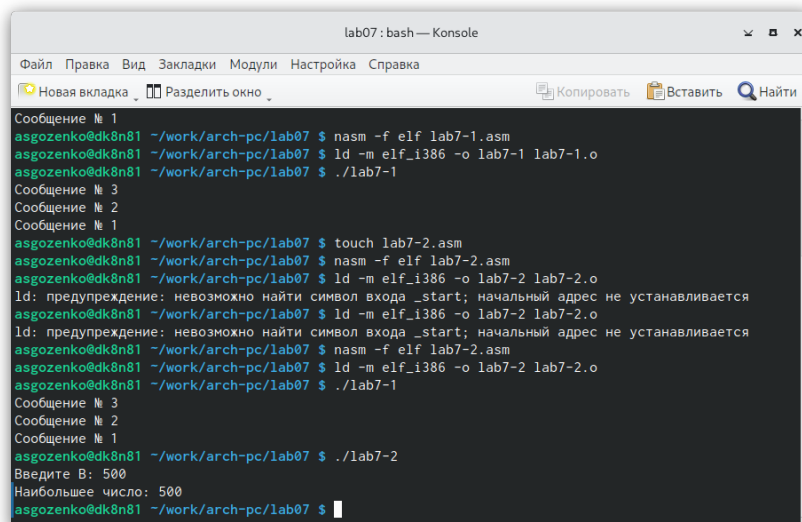


Рис. 2.9: Работа программы

Создание файла листинга(рис. 2.10).

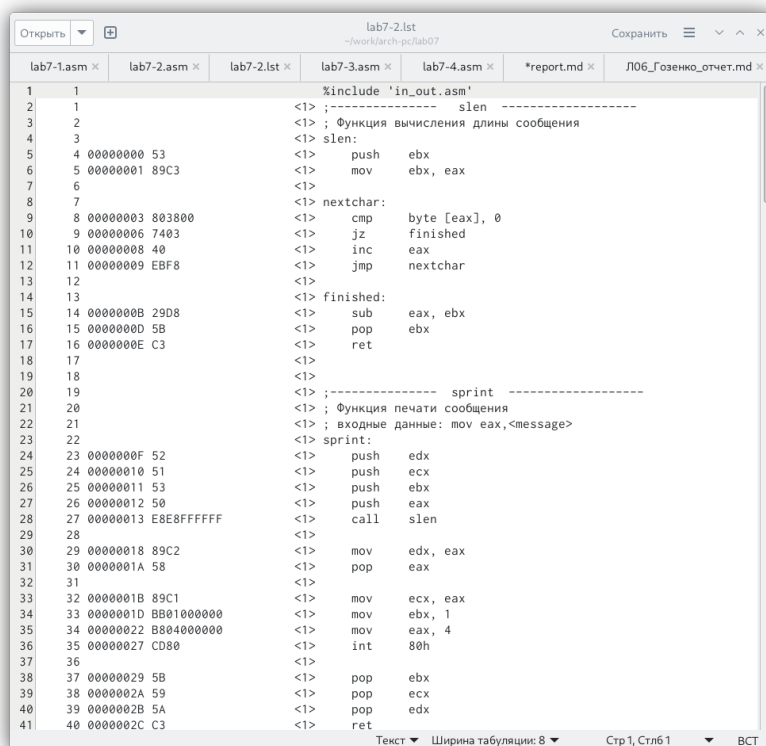

```

asgozenko@dk8n81 ~/work/arch-pc/lab07 $ nasm -f elf -l lab7-2.lst lab7-2.asm
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ mcedit lab7-2.lst
asgozenko@dk8n81 ~/work/arch-pc/lab07 $

```

Рис. 2.10: Создание файла листинга

Файл листинга(рис. ??).



Рассмотрим файл листинга одной из программ. В строке 8 содержится номер строки [8], адрес[00000003], машинный код[803800] и содержимое строки кода[cmp byte [eax], 0]. В строке 10 содержится номер строки [10], адрес[00000008], машинный код[40] и содержимое строки кода[inc eax]. В строке 23 содержится номер строки [23], адрес[0000000F], машинный код[52] и содержимое строки кода[push edx].

Специально допущенная ошибка в коде программы(рис. 2.11).

```

13
14 mov eax, |
15 call sprint
16
17 mov ecx, B

```

Рис. 2.11: Ошибка

Реакция листинга на допущенную ошибку, добавляется описание ошибке в файле листинга(рис. 2.12).

The screenshot shows a debugger window with three tabs: lab7-1.asm, lab7-2.asm, and lab7-2.lst. The lab7-2.lst tab is active, displaying assembly code with addresses and hex values in the first column, and assembly instructions in the second column. An error message is visible in the third column, indicating an invalid combination of opcode and operands for the instruction at address 14.

```

170 169 000000E5 CD80      <1>      int      80h
171 170 000000E7 C3      <1>      ret
172      2                      SECTION .data
173      3 00000000 D092D0B2D0B5D0B4D0-    msg1 db 'Введите B: ', 0h
174      3 00000009 B8D182D0B520423A20-
175      3 00000012 00
176      4 00000013 D09DD0B0D0B8D0B1D0-    msg2 db "Наибольшее число: ", 0h
177      4 0000001C BED0BBD18CD188D0B5-
178      4 00000025 D0B520D187D0B8D181-
179      4 0000002E D0BBD0BE3A2000
180      5 00000035 32300000      A dd '20'
181      6 00000039 35300000      C dd '50'
182      7                      SECTION .bss
183      8 00000000 <res Ah>      max resb 10
184      9 0000000A <res Ah>      B resb 10
185      10                     SECTION .text
186      11                     GLOBAL _start
187      12                     _start:
188      13
189      14                     mov eax,
190      14                     *****
191      15 000000E8 E822FFFFFF      error: invalid combination of opcode and operands
192      16                     call sprint
193      17 000000ED B9[0A000000]      mov ecx, B
194      18 000000F2 BA0A000000      mov edx, 10
195      19 000000F7 E847FFFFFF      call sread
196      20
197      21 000000FC B8[0A000000]      mov eax, B
198      22 00000101 E896FFFFFF      call atoi
199      23 00000106 A3[0A000000]      mov [B], eax
200      24
201      25 0000010B 8B0D[35000000]      mov ecx, [A]

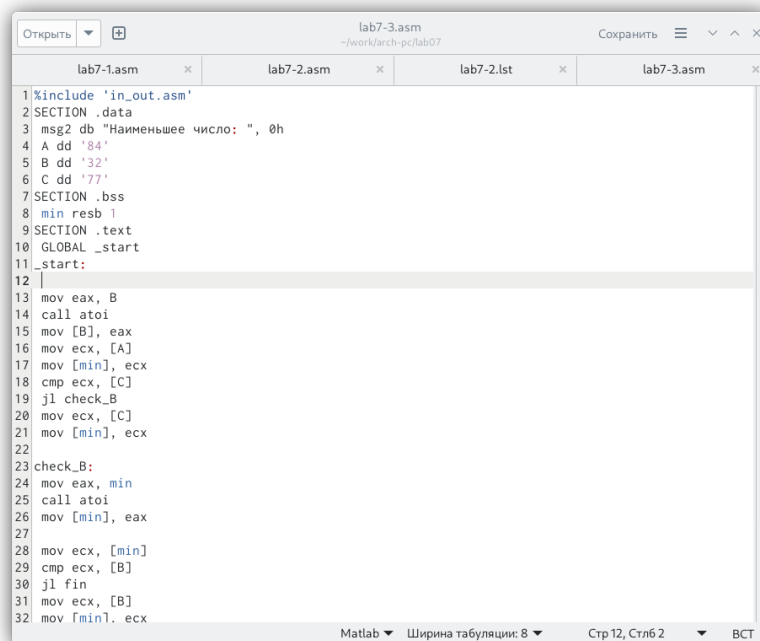
```

At the bottom of the window, there is a status bar with the text: Текст, Ширина табуляции: 8, Стр 1, Стлб 1, ВСТ.

Рис. 2.12: Описание ошибки

3 Выполнение задания для самостоятельной работы. Вариант 13.

Написание программы для пункта 1 самостоятельной работы(рис. 3.1).



```
lab7-3.asm
~\work\arch-pc\lab07

lab7-1.asm x lab7-2.asm x lab7-2.lst x lab7-3.asm x

1 %include 'in_out.asm'
2 SECTION .data
3 msg2 db "Наименьшее число: ", 0h
4 A dd '84'
5 B dd '32'
6 C dd '77'
7 SECTION .bss
8 min resb 1
9 SECTION .text
10 GLOBAL _start
11 _start:
12
13 mov eax, B
14 call atoi
15 mov [B], eax
16 mov ecx, [A]
17 mov [min], ecx
18 cmp ecx, [C]
19 jl check_B
20 mov ecx, [C]
21 mov [min], ecx
22
23 check_B:
24 mov eax, min
25 call atoi
26 mov [min], eax
27
28 mov ecx, [min]
29 cmp ecx, [B]
30 jl fin
31 mov ecx, [B]
32 mov [min], ecx

Matlab Ширина табуляции: 8 Стр 12, Стлб 2 ВСТ
```

Рис. 3.1: Код программы

Проверка работы программы(рис. 3.2).

```

asgozenko@dk8n81 ~/work/arch-pc/lab07 $ nasm -f elf lab7-3.asm
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-3 lab7-3.o
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ./lab7-3
Наименьшее число: 32
asgozenko@dk8n81 ~/work/arch-pc/lab07 $

```

Рис. 3.2: Работа программы

Код Программы из пункта 1:

```

#include 'in_out.asm' SECTION .data msg2 db "Наименьшее число:", 0h A dd '84'
B dd '32' C dd '77' SECTION .bss min resb 1 SECTION .text GLOBAL _start _start:
    mov eax, B call atoi mov [B], eax mov ecx, [A] mov [min], ecx cmp ecx, [C] jl check_B
mov ecx, [C] mov [min], ecx
check_B: mov eax, min call atoi mov [min], eax
mov ecx, [min] cmp ecx, [B] jl fin mov ecx, [B] mov [min], ecx
fin: mov eax, msg2 call sprint mov eax, [min] call iprintLF call quit

```

Написание программы для пункта 2 самостоятельной работы(рис. 3.3).

```

1 %include 'in_out.asm'
2
3 SECTION .data
4 msg1: db 'Введите x: ', 0h
5 msg2: db 'Введите a: ', 0h
6
7 SECTION .bss
8 max 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 sprint
18
19 mov ecx, x
20 mov edx, 10
21 call sread
22
23 mov eax, x
24 call atoi
25 mov [x], eax
26
27 mov eax, msg2
28 call sprint
29
30 mov ecx, a
31 call sread
32

```

Рис. 3.3: Код программы

Проверка работы программы(рис. 3.4).

```
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ nasm -f elf lab7-4.asm
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-4 lab7-4.o
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ./lab7-4
Введите x: 3
Введите a: 9
2
asgozenko@dk8n81 ~/work/arch-pc/lab07 $ ./lab7-4
Введите x: 6
Введите a: 4
24
asgozenko@dk8n81 ~/work/arch-pc/lab07 $
```

Рис. 3.4: Работа программы

4 Выводы

Я изучил команды условного и безусловного переходов ассемблера NASM. Приобрёл навыки написания программ с использованием переходов.