

Отчет по лабораторной работе №7

Гиршфельд Александр Евгеньевич

Содержание

1	Цель работы	5
2	Выполнение лабораторной работы	6
3	Выводы	12

Список иллюстраций

2.1	создание папки и файла	6
2.2	запуск файла	6
2.3	работа программы	8

Список таблиц

1 Цель работы

Изучить основы работы команд усовного и безусловного перехода в assembler.

2 Выполнение лабораторной работы

Создадим рабочую папку и рабочий файл (рис. 2.1).

```
aegirshfeljd@dk3n38 ~ $ mkdir ~/work/arch-pc/lab07
aegirshfeljd@dk3n38 ~ $ cd ~/work/arch-pc/lab07
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ touch lab7-1.asm
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $
```

Рис. 2.1: создание папки и файла

Запишем в файл код, проассемблируем его, запустим (рис. 2.2)

```
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ nasm -f elf lab7-1.asm
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ ld -m elf_i386 -o lab7-1 lab7-1.o
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 3
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $
```

Рис. 2.2: запуск файла

зменим программу так, чтобы она выводила второе, затем первое сообщение и завершала работу. Её код (рис. ??) и работа (рис. ??)

```

#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 sprintf
jmp _end

_label2:
mov eax, msg2
call sprintf
jmp _label1

_label3:
mov eax, msg3
call sprintf

_end:
call quit

```

```

aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 2
Сообщение № 1
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $

```

Еще напишем программу, выводящую сообщения в обратном порядке(рис. ??)
(рис. ??)

```

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
12 jmp _label3
13
14 _label1:
15 mov eax, msg1
16 call sprintf
17 jmp _end
18
19 _label2:
20 mov eax, msg2
21 call sprintf
22 jmp _label1
23
24 _label3:
25 mov eax, msg3
26 call sprintf
27 jmp _label2
28
29 _end:
30 call quit

```

```

aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ nasm -f
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ ld -m el
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $

```

Напишем программу с условным переходом(рис. 2.3)

```

aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ ./lab7-2
Введите B: 12
Наибольшее число: 50
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $ ./lab7-2
Введите B: 45
Наибольшее число: 50
aegirshfeljd@dk3n38 ~/work/arch-pc/lab07 $

```

Рис. 2.3: работа программы

```

lab7-2.lst [----] 0 L: [ 1+ 0 1/220] *(0 /13
1 %include 'in_out.
2 <1> ;-----
3 <1> ; Функция вычисле
4 00000000 53 <1> push ebx
5 00000001 89C3 <1> mov ebx,
6 <1>
7 <1> nextchar:
8 00000003 803800 <1> cmp byte
9 00000006 7403 <1> jz finis
10 00000008 40 <1> inc eax
11 00000009 EBF8 <1> jmp nextc
12 <1>
13 <1> finished:
14 0000000B 29D8 <1> sub eax,
15 0000000D 5B <1> pop ebx
16 0000000E C3 <1> ret
17 <1>
18 <1>
19 <1> ;-----
20 <1> ; Функция печати
21 <1> ; входные данные:
22 <1> sprint:
23 0000000F 52 <1> push edx
24 00000010 51 <1> push ecx
25 00000011 53 <1> push ebx
26 00000012 50 <1> push eax
27 00000013 E8E8FFFFFF <1> call slen
28 <1>
29 00000018 89C2 <1> mov edx,
30 0000001A 58 <1> pop eax
31 <1>
32 0000001B 89C1 <1> mov ecx,
33 0000001D BB01000000 <1> mov ebx,
34 00000022 B804000000 <1> mov eax,

```

Рассмотрим файл листинга одной из программ(рис. ??)

в строке 9 содержится собственно номер строчки [9], адресс [00000003], машинный код [803800] и содержимое строки кода [cmp byte [eax], 0] в строке 11 содержится номер строчки [11], адресс [00000008], машинный код [40] и содержимое строки кода [inc eax] в строке 24 содержится номер строчки [24], адресс [0000000F], машинный код [52] и содержимое строки кода [push edx]

Если в коде появляется ошибка, то ее описание появится в файле листинга

Задания для самостоятельной работы

вариант 16

программа для сравнения трех заранее известных чисел(рис. ??) и ее работа(рис. ??)


```
Открыть + lab7-3.asm ~\work\arch-pc\lab07 Сохранить ≡ v
1 %include 'in_out.asm'
2 section .data
3 msg2 db "Наибольшее число: ",0h
4 A dd '44'
5 B dd '74'
6 C dd '17'
7
8 section .bss
9 max resb 10
10
11 section .text
12
13 global _start
14 _start:
15 mov eax, B
16 call atoi
17 mov [B], eax
18
19 mov ecx, [A]
20 mov [max],ecx
21
22 cmp ecx, [C]
23 jg check_B
24 mov ecx,[C]
25 mov [max], ecx
26
27
28 check_B:
29 mov eax,max
30 call atoi
31 mov [max],eax
32
33 mov ecx,[max]
34 cmp ecx,[B]
35 jg fin
36 mov ecx,[B]
37
38 mov [max],ecx
39
40
41 fin:
42 mov eax, msg2
43 call sprint
44 mov eax,[max]
45 call iprintLF
```

```
aegirshfeljd@dk3n38 ~\work\arch-pc\lab07 $ nasm -f elf lab7-3.asm
aegirshfeljd@dk3n38 ~\work\arch-pc\lab07 $ ld -m elf_i386 -o lab7-3
aegirshfeljd@dk3n38 ~\work\arch-pc\lab07 $ ./lab7-3
Наибольшее число: 74
Ошибка сегментирования (стек памяти сброшен на диск)
aegirshfeljd@dk3n38 ~\work\arch-pc\lab07 $
```

Программа для вычисления выражения в зависимости от условия на одну из вводимых переменных(рис. ??) (рис. ??)и ее работа(рис. ??)

```

%include 'in_out.asm'

SECTION .data
input1 db "Введите x: ",0h
input2 db "Введите a: ",0h

SECTION .bss
max resb 10
x resb 10
a resb 10

SECTION .text
GLOBAL _start

_start:
mov eax,input1
call sprint

mov ecx,a
mov edx,10
call sread

mov eax,a
call atoi
mov [a],eax

mov eax,input2
call sprint

mov ecx,x
mov edx,10
call sread

mov eax,x
call atoi
mov [x],eax

mov ebx, 4
cmp [x], ebx
jge check

mov eax, [x]
mov ebx, 4
add eax, ebx
call iprintLF
call quit

```

```

8 mov ebx, 4
9 cmp [x], ebx
0 jge check
1
2 mov eax, [x]
3 mov ebx, 4
4 add eax, ebx
5 call iprintLF
6 call quit
7
8 check:
9 mov eax, [x]
0 mov ebx, [a]
1 mul ebx
2 call iprintLF
3 call quit

```

```
aegirshfeljd@dk8n68 ~/work/arch-pc/lab07 $ ./lab7-4
Введите x: 1
Введите a: 1
5
aegirshfeljd@dk8n68 ~/work/arch-pc/lab07 $ ./lab7-4
Введите x: 7
Введите a: 1
5
aegirshfeljd@dk8n68 ~/work/arch-pc/lab07 $ █
```

3 Выводы

Были изучены основные принципы работы с условным и безусловным переходом в assembler и изучены основы чтения файлов листинга.