

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

дисциплина: Архитектура компьютера

Лисовская Арина Валерьевна

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Символьные и численные данные в NASM	6
3.2	Задание для самостоятельной работы	8
4	Выводы	10

Список иллюстраций

3.1	файл	6
3.2	jmr работает	6
3.3	Работа программы	6
3.4	ошибка в файле листинга	8

1 Цель работы

Изучить основы работы команд усовного и безусловного перехода в assembler.

2 Задание

- 1.Символьные и численные данные в NASM
- 2.Выполнение арифметических операций в NASM
- 3.Выполнение заданий для самостоятельной работы

3 Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

Создадим рабочую папку и рабочий файл (рис. [3.1]).

Левая панель		Файл	Команда	Настройки	Правая панель	
~ /work/arch-pc/lab07			[^]>	~	[^]>	
.и	Имя	Размер	Время правки	.и	Имя	Размер
./..	-ВВЕРХ-	ноя 21 13:34		./..	-ВВЕРХ-	окт 14 14:44
in out.asm	3942	ноя 16 13:54		./cache	4096	ноя 18 21:53
*lab7-1	9200	ноя 21 13:59		./config	4096	ноя 18 22:26
lab7-1.asm	650	ноя 21 13:56		./git	4096	окт 28 19:37
lab7-1.o	1440	ноя 21 13:58		./gnupg	4096	ноя 18 20:30

Рис. 3.1: файл

Запишем в файл код, проасSEMBЛИруем его, запустим(рис. [3.2])

```
[2] ~ установлен ПК
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$
```

Рис. 3.2: jmp работает

Напишем программу с условным переходом(рис. [3.3])

```
[3] ~ установлен ПК
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ ./lab7-2
Сообщение № 2
Сообщение № 1
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$
```

Рис. 3.3: Работа программы

1		%inc
2		<1> ;---
3		<1> ; Фy
4		<1> slen
5	00000000 53	<1>
6	00000001 89C3	<1>
7		<1>
8		<1> next
9	00000003 803800	<1>
10	00000006 7403	<1>
11	00000008 40	<1>
12	00000009 EBF8	<1>
13		<1>
14		<1> fini
15	0000000B 29D8	<1>
16	0000000D 5B	<1>
17	0000000E C3	<1>
18		<1>
19		<1>
20		<1> ;---
21		<1> ; Фy
22		<1> ; vx
23		<1> spri
24	0000000F 52	<1>
25	00000010 51	<1>
26	00000011 53	<1>
27	00000012 50	<1>
28	00000013 E8E8FFFFFF	<1>
29		<1>
30	00000018 89C2	<1>
31	0000001A 58	<1>
32		<1>
33	0000001B 89C1	<1>
34	0000001D BB01000000	<1>

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

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

Если в коде появляется ошибка, то ее описание появится в файле листинга(рис. [3.4])

```

  9
10 00000000 <res Ah>      section .bss
11 0000000A <res Ah>      max resb 10
12                                B resb 10
13
14                                section .text
15
16                                global _start
17                                _start:
18                                mov eax
19                                *****
20 000000E8 E822FFFFFF      call sprint
21                                error: invalid combination of opcode and operands
22 000000ED B9[0A000000]    mov ecx, B
23                                mov edx, 10
24 000000F2 BA0A000000      call sread
25                                call sread
26 000000F7 E847FFFFFF      mov eax, B
27                                call atoi
28 00000106 A3[0A000000]    mov [B], eax
                                mov ecx, [A]

```

Рис. 3.4: ошибка в файле листинга

3.2 Задание для самостоятельной работы

(Вар 15)

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

```

GNU nano 6.2 /home/avlisovskaya/work/arch-pc/lab07/lab7-3.asm *
#include 'in_out.asm'

section .data
    msg1 db "Наименьшее число:"
    a dd 32
    b dd 6
    c dd 54

section .bss
    min resb 10

section .text
    global _start

_start:
    mov eax, msg1
    call sprint

    mov ecx, [a]
    mov [min], ecx ; 'min = A'

```

```

avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ nas
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ d -
d: команда не найдена
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ ld
o
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ ./l
Наименьшее число: 6
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$

```

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


```

GNU nano 6.2 /home/avlisovskaya/work/arch-pc/lab07/lab
%include 'in_out.asm'
SECTION .data
msg: DB 'Введите x: ',0
msg1: DB 'Введите a: ',0
msg2: DB 'Результат: ',0
SECTION .bss
x: RESB 80
a: RESB 80
otv: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, msg
call sprintLF
mov ecx, x
mov edx, 80
call sread
mov eax, x
call atoi
mov [x], eax
mov eax, msg1
call sprintLF
mov ecx, a
mov edx, 80
call sread
mov eax, a
call atoi
mov [a], eax
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ ./lab7-4
Введите x:
4
Введите a:
2
Результат:
20
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$

```

```

[5] 10:07:07 PM
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ nc
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ l
0
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$ .
Введите x:
2
Введите a:
3
Результат:
30
avlisovskaya@avlisovskaya:~/work/arch-pc/lab07$

```

4 Выводы

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