

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 5

дисциплина: Архитектура компьютера

Студент: Маслова Анастасия Сергеевна

Группа: НКНбд-01-21

МОСКВА

2021 г.

Цель работы: знакомство со структурой и предназначением файла листинга, получение навыков использования команд условного и безусловного переходов, изучение принципов написания программ с использованием переходов.

Ход работы:

1. В каталоге Architecture_PC создала каталог lab05 с помощью команды mkdir и файлы lab5-1.asm, lab5-2.asm с помощью команды touch (рис.1).

```
asmaslova@ubuntu2104:~$ ls
Architecture_PC Desktop Documents Downloads Music Pictures Public Templates Videos
asmaslova@ubuntu2104:~$ cd Architecture_PC/
asmaslova@ubuntu2104:~/Architecture_PC$ mkdir lab05
asmaslova@ubuntu2104:~/Architecture_PC$ ls
lab01 lab02 lab03 lab04 lab05
asmaslova@ubuntu2104:~/Architecture_PC$ cd lab05
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ touch lab5-1.asm lab5-2.asm
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ ls
lab5-1.asm lab5-2.asm
```

рис. 1. Создание каталога и файлов

2. С помощью mcedit я написала программу для нахождения наименьшей из 3 целочисленных переменных a, b и c. В 4-й лабораторной работе у меня был 16 вариант, поэтому мне достались числа 44, 74 и 17. Создав исполняемый файл, я убедилась, что все работает верно (рис.2).

```
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ mcedit lab5-1.asm

asmaslova@ubuntu2104:~/Architecture_PC/lab05$ nasm -f elf lab5-1.asm
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ ld -m elf_i386 lab5-1.o -o lab5-1
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ ./lab5-1
Введите b:74
Наименьшее число:17
```

рис. 2. Создание исполняемого файла и проверка работы программы

3. У меня 16 вариант, и поэтому мне нужно было написать программу, которая при $x < 4$ вычисляла бы $x+4$, а при x больше или равно 4 вычисляла бы $a*x$. Написав программу, я проверила её работу на первой паре чисел: $a=1$ и $x=1$ (рис.3)

```
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ nasm -f elf lab5-2.asm
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ ld -m elf_i386 lab5-2.o -o lab5-2
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ ./lab5-2
Введите x:
1
Введите a:
1
Результат:5
```

рис. 3. Проверка работы программы на первой паре чисел

При $x < 4$ программа работала правильно, поэтому я проверила её работу на второй паре чисел: $x=7$ и $a=1$ (рис.4)

```
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ ./lab5-2
Введите x:
7
Введите a:
1
Результат:7
```

рис. 4. Проверка работы программы на второй паре чисел

4. Чтобы получить файл листинга, я в функцию `nasm` добавила `-l` и название файла, после чего открыла его с помощью `mcedit` (рис.5).

```
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ nasm -f elf -l lab5-2.lst lab5-2.asm
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ ls
in_out.asm lab5-1 lab5-1.asm lab5-1.o lab5-2 lab5-2.asm lab5-2.lst lab5-2.o
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ mcedit lab5-2.lst
```

рис. 5. Создание и открытие файла листинга

Я попробовала удалить один операнд во второй программе в инструкции сложения, но не смогла проассемблировать программу с файлом листинга, потому что ассемблер выдал ошибку с номером именно той строки, где я совершила намеренную ошибку (рис. 6). Тем не менее, в файле листинга появилась строка с описанием ошибки (рис.7).

```
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ mcedit lab5-2.asm
asmaslova@ubuntu2104:~/Architecture_PC/lab05$ nasm -f elf -l lab5-2.lst lab5-2.asm
lab5-2.asm:48: error: invalid combination of opcode and operands
```

рис. 6. Ошибка при ассемблировании

```
/home/asmaslova/Ar~/lab05/lab5-2.lst [----] 51 L:[222+22 244/272] *(14432/15559b) 0114 0x072 [*][X]
46 less:
47 00000140 A1[00000000] mov eax,[x]
48 add eax
48 ***** error: invalid combination of opcode and operands
49 00000145 B8[A0000000] mov eax,rez
50 0000014A E840FFFFFF call atoi
51 0000014F A3[A0000000] mov [rez],eax
52 00000154 EB1E jmp fin
53 .....
54 greaterequal:
55 00000156 891D[50000000] mov [a],ebx
56 0000015C A1[00000000] mov eax,[x]
57 00000161 F7E3 mul ebx
58 00000163 B8[A0000000] mov eax,rez
59 00000168 E82FFFFFFF call atoi
60 0000016D A3[A0000000] mov [rez],eax
61 00000172 EB00 jmp fin
```

рис. 7. Строка с ошибкой в файле листинга

Для объяснений содержимого файла листинга я решила взять строки из второй программы (рис.8).

```
/home/asmaslova/Ar~/lab05/lab5-2.lst [----] 53 L:[172+18 190/271] *(11560/15470b) 0010 0x00A [*][X]
2 .....
3 ..... SECTION .data
4 00000000 D092D0B2D0B5D0B4D0- msg1: DB 'Введите x:',0
4 00000009 B8D182D0B520D1853A-
4 00000012 00.....
5 00000013 D092D0B2D0B5D0B4D0- msg2: DB 'Введите a:',0
5 0000001C B8D182D0B520D0B03A-
5 00000025 00.....
6 00000026 D0A0D0B5D0B7D183D0- div: DB 'Результат:',0
6 0000002F BBD18CD182D0B0D182-
6 00000038 3A00.....
7 .....
8 ..... SECTION .bss
9 00000000 <res 50h> x: RESB 80
10 00000050 <res 50h> a: RESB 80
11 000000A0 <res 50h> rez: RESB 80
12 .....
13 ..... SECTION .text
14 ..... GLOBAL _start
15 .....
16 ..... _start:
17 ..... ;ввод x
18 000000E8 B8[00000000] mov eax,msg1
19 000000ED E83BFFFFFF call sprintf
20 .....
21 000000F2 B9[00000000] mov ecx,x
22 000000F7 BA50000000 mov edx,80
23 000000FC E842FFFFFF call sread
24 .....
25 00000101 B8[00000000] mov eax,x
26 00000106 E891FFFFFF call atoi
27 0000010B A3[00000000] mov [x],eax
28 .....
```

рис. 8. Файл листинга второй программы

Строка 6 состоит из номера строки (6), адреса (00000026), машинного кода (D0A0D0B5D0B7D183D0BBD18CD182D0B0D1823A00) и исходного текста программы.

Строка 18 состоит так же из номера строки (18), адреса (000000E8), машинного кода (B8[00000000]) и исходного текста программы.

Строка 26 состоит из номера строки (26), адреса (00000106), машинного кода (E891FFFFFF) и исходного текста программы.

Вывод: в ходе работы я научилась создавать файл листинга и узнала, для чего он необходим. Кроме того, я изучила команды условного и безусловного перехода и применила эти знания на практике.