

# РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

## ОТЧЕТ

### ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4

*дисциплина:* Архитектура компьютера

Студент: Бешкуров Михаил

Группа: НКНбд-01-18

МОСКВА

2018 г.

## Цель работы

Знакомство с назначением и структурой файла листинга. Изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов.

## Ход работы

1. Я написал программу, работающую по следующему алгоритму:

- (a) вывести на экран запрос о времени дня, например, «Полдень прошел?»;
- (b) принять с клавиатуры ответ (Y/N);
- (c) если было введено N выдать сообщение «Доброе утро», в противном случае — «Добрый день».

```
/home/mihail/Рабочий стол/rudn/lab4/[----] 13 L:[ 1+ 0 1/ 54] *(13 / 761b) 0010 0x00A [*][X]
SECTION .data
    text:<-----DB 'Полдень прошел? Y/N',10
    textLen<---->EQU $-text
    utro:<-----DB 'Доброе утро!',10
    utroLen<---->EQU $-utro
    den:<-----DB 'Добрый день!',10
    denLen<---->EQU $-den

SECTION .bss
    input:<-----RESB 80

SECTION .text
    GLOBAL _start

_start:
    mov eax,4
    mov ebx,1
    mov ecx,text
    mov edx,textLen
    int 80h

    mov eax,3
    mov ebx,0
    mov ecx,input
    mov edx,80
    int 80h

    cmp byte[input],'Y'
    je dden
    cmp byte[input],'N'
    je utr
    jmp _start

utr:
    mov eax,4
    mov ebx,1
    mov ecx,utro
    mov edx,utroLen
    int 80h
    jmp end

dden:
    mov eax,4
    mov ebx,1
    mov ecx,den
    mov edx,denLen
    int 80h
    jmp end

end:
    mov eax,1
    mov ebx,0
    int 80h
```

```
mihail@mihail-beshkurov ~/Рабочий стол/rudn/lab4 $ ./lab
Полдень прошел? Y/N
Y
Добрый день!
mihail@mihail-beshkurov ~/Рабочий стол/rudn/lab4 $ ./lab
Полдень прошел? Y/N
N
Доброе утро!
mihail@mihail-beshkurov ~/Рабочий стол/rudn/lab4 $
```

Программа работает.

2. Получил файл листинга при помощи команды `nasm -l lab.lst lab4.asm` и ознакомился с его форматом и содержимым.

```
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $ nasm -l lab.lst lab4.asm
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $ ls
lab lab4 lab4.asm lab.lst lab.o makefile
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $
```

Файл листинга:

```
1          SECTION .data
2 00000000 D09FD0BED0BBD0B4D0-      text: DB 'Полдень прошел? Y/N',10
3 00000009 B5D0BDD18C20D0BFD1-
4 00000012 80D0BED188D0B5D0BB-
5 0000001B 3F20592F4E0A
6
7 00000021 D094D0BED0B1D180D0-      textLen EQU $-text
8 0000002A BED0B520D183D182D1-      utro: DB 'Доброе утро!',10
9 00000033 80D0BE210A
10
11 00000038 D094D0BED0B1D180D1-      utroLen EQU $-utro
12 00000041 8BD0B920D0B4D0B5D0-      den: DB 'Добрый день!',10
13 0000004A BDD18C210A
14
15
16          SECTION .bss
17 00000000 <res 00000050>          input: RESB 80
18
19          SECTION .text
20          GLOBAL _start
21
22          _start:
23 00000000 66B804000000          mov eax,4
24 00000006 66BB01000000          mov ebx,1
25 0000000C 66B9[00000000]      mov ecx,text
26 00000012 66BA21000000          mov edx,textLen
27 00000018 CD80              int 80h
28
29 0000001A 66B803000000          mov eax,3
30 00000020 66BB00000000          mov ebx,0
31 00000026 66B9[00000000]      mov ecx,input
32 0000002C 66BA50000000          mov edx,80
33 00000032 CD80              int 80h
34
35 00000034 803E[0000]59          cmp byte[input],'Y'
36 00000039 7425              je dden
37 0000003B 803E[0000]4E          cmp byte[input],'N'
38 00000040 7402              je utr
39 00000042 EBBC              jmp _start
40
41          utr:
42 00000044 66B804000000          mov eax,4
43 0000004A 66BB01000000          mov ebx,1
44 00000050 66B9[21000000]      mov ecx,utro
45 00000056 66BA17000000          mov edx,utroLen
46 0000005C CD80              int 80h
47 0000005E EB1C              jmp end
48
49          dden:
50 00000060 66B804000000          mov eax,4
51 00000066 66BB01000000          mov ebx,1
52 0000006C 66B9[38000000]      mov ecx,den
53 00000072 66BA17000000          mov edx,denLen
54 00000078 CD80              int 80h
55 0000007A EB00              jmp end
56
57          end:
58 0000007C 66B801000000          mov eax,1
59 00000082 66BB00000000          mov ebx,0
60 00000088 CD80              int 80h
```

3. В инструкции с двумя операндами удалил один операнд и проассемблировал программу с получением файла листинга.

```
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $ nasm -f elf32 -l lab.lst lab4.asm
lab4.asm:28: error: invalid combination of opcode and operands
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $
```

Создался файл листинга, в котором добавилась строка об ошибке (№36).

```
28
29 00000016 B803000000          mov eax,3
30 0000001B BB00000000          mov ebx,0
31 00000020 B9[00000000]      mov ecx,input
32 00000025 BA50000000          mov edx,80
33 0000002A CD80              int 80h
34
35          cmp byte[input]
36          *****      error: invalid combination of opcode and operands
37 0000002C 7423              je dden
38 0000002E 803D[00000000]4E          cmp byte[input],'N'
39 00000035 7402              je utr
40 00000037 EBC7              jmp _start
41
```

#### 4. Объяснить три строки из файла листинга.

```
28
29 00000016 B803000000      mov eax,3
30 0000001B BB00000000      mov ebx,0
31 00000020 B9[00000000]    mov ecx,input
32 00000025 BA50000000      mov edx,80
33 0000002A CD80          int 80h
34
35                          cmp byte[input]
36          *****      error: invalid combination of opcode and operands
37 0000002C 7423          je dden
38 0000002E 803D[00000000]4E      cmp byte[input],'N'
39 00000035 7402          je utr
40 00000037 EBC7          jmp _start
41
```

а) 29 — номер строки файла листинга. 00000016 — смещение машинного кода от начала текущего сегмента (адрес). B803000000 — машинный код в шестнадцатеричном представлении, который означает системный вызов read. mov eax,3 — код программы.

б) 30 — номер строки файла листинга. 0000001B — смещение машинного кода от начала текущего сегмента (адрес). BB00000000 — машинный код, который означает стандартный ввод. mov ebx,0 — код программы.

в) 31 — номер строки файла листинга. 00000020 — смещение машинного кода от начала текущего сегмента (адрес). B9[00000000] — машинный код, который означает адрес переменной input в ecx.

#### 5. При помощи команды **ls -l** получим имя владельца и права доступа.

```
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $ ls -l
total 20
-rwxr-xr-x 1 mi hail mi hail 1044 окт 14 22:49 lab
-rw-r--r-- 1 mi hail mi hail 761 окт 14 22:49 lab4.asm
-rw-r--r-- 1 mi hail mi hail 3133 окт 14 22:49 lab4.lst
-rw-r--r-- 1 mi hail mi hail 1040 окт 14 22:49 lab.o
-rw-r--r-- 1 mi hail mi hail 101 окт 14 22:48 makefile
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $
```

Для всех файлов установлены мои права, как владельца, на чтение(r) и запись(w), а для исполнимого файла lab — ещё и на выполнение (x); для членов группы владельца и остальных разрешено только чтение файлов.

#### 6. При помощи команды **chmod 444 programm** изменим права доступа на файл lab (запретим выполнение файла) и попробуем запустить его.

```
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $ chmod 444 lab
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $ ls -l
total 20
-r--r--r-- 1 mi hail mi hail 1044 окт 14 22:49 lab
-rw-r--r-- 1 mi hail mi hail 761 окт 14 22:49 lab4.asm
-rw-r--r-- 1 mi hail mi hail 3133 окт 14 22:49 lab4.lst
-rw-r--r-- 1 mi hail mi hail 1040 окт 14 22:49 lab.o
-rw-r--r-- 1 mi hail mi hail 101 окт 14 22:48 makefile
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $ ./lab
bash: ./lab: Permission denied
mi hail@mi hail-beshkurov ~/Рабочий стол/rudn/lab4 $
```

7. Разрешим выполнение исходного текста программы как исполняемого файла при помощи команды **chmod 744 lab4.asm** и попробуем запустить его.

```
mihail@mihail-beshkurov ~/Рабочий стол/rudn/lab4 $ chmod 744 lab4.asm
mihail@mihail-beshkurov ~/Рабочий стол/rudn/lab4 $ ls -l
total 20
-rwxr-xr-x 1 mihail mihail 1044 окт 14 22:49 lab
-rwxr--r-- 1 mihail mihail 761 окт 14 22:49 lab4.asm
-rw-r--r-- 1 mihail mihail 3133 окт 14 22:49 lab4.lst
-rw-r--r-- 1 mihail mihail 1040 окт 14 22:49 lab.o
-rw-r--r-- 1 mihail mihail 101 окт 14 22:48 makefile
mihail@mihail-beshkurov ~/Рабочий стол/rudn/lab4 $ ./lab4.asm
./lab4.asm: line 1: SECTION: команда не найдена
./lab4.asm: line 2: text:: команда не найдена
./lab4.asm: line 3: textLen: команда не найдена
./lab4.asm: line 4: utro:: команда не найдена
./lab4.asm: line 5: utroLen: команда не найдена
./lab4.asm: line 6: den:: команда не найдена
./lab4.asm: line 7: denLen: команда не найдена
./lab4.asm: line 9: SECTION: команда не найдена
./lab4.asm: line 10: input:: команда не найдена
./lab4.asm: line 12: SECTION: команда не найдена
./lab4.asm: line 13: GLOBAL: команда не найдена
./lab4.asm: line 15: _start:: команда не найдена
./lab4.asm: line 16: mov: команда не найдена
./lab4.asm: line 17: mov: команда не найдена
./lab4.asm: line 18: mov: команда не найдена
./lab4.asm: line 19: mov: команда не найдена
./lab4.asm: line 20: int: команда не найдена
./lab4.asm: line 22: mov: команда не найдена
./lab4.asm: line 23: mov: команда не найдена
```

Командная строка думает, что в файле lab4.asm находятся команды для неё, но это не так. В файле находится текст программы на языке ассемблера.

### Вывод

В ходе работы я познакомился со структурой файла листинга, изучил команды условного и безусловного перехода, приобрел навыки написания программы с использованием переходов.

### Контрольные вопросы

1. В Unix-подобных системах набор прав доступа к файлу различаются в зависимости от категории пользователя (владелец, член группы владельца, все остальные). Это позволяет защитить файлы от нежелательного вмешательства других пользователей.
2. Исполняемый файл - тот, для которого определен атрибут "X" (право на выполнение). Права на чтение и запись можно изменять консольной командой **chmod**.
3. Набор прав доступа задается тройками битов и состоит из прав на чтение, запись и исполнение файла. В символьном представлении он выглядит как строка gwx, где вместо любого символа может стоять дефис. Эти биты могут быть записаны как восьмеричное число.
4. Листинг — это один из выходных файлов, создаваемых транслятором. Он имеет текстовый вид и нужен при отладке программы, т. к. кроме строк самой программы содержит дополнительную информацию.
5. Формат файла листинга — текстовый, расширение — **lst**. Он состоит из «номера строки», «адреса», «машинного кода» и «исходного текста программы». «Номер строки» представляет собой номер строки файла листинга. Номера строк особенно полезны при работе с перекрестными ссылками.

6. С помощью команд условного и безусловного перехода (команда сравнения `cmp` и `jmp`-команда), с помощью этих команд можно переходить к другим блокам программы.