

Отчёт по лабораторной работе Архитектура компьютера

Лабораторная работа №7

Дмитрий Алексеевич Митяков

Содержание

1 Цель работы

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

2 Выполнение лабораторной работы

Я создал нужную директиву и необходимый файл (рис. ??).

```
[damityakov@fedora ~]$ mkdir ~/work/arch-pc/lab07  
[damityakov@fedora ~]$ cd ~/work/arch-pc/lab07  
[damityakov@fedora lab07]$ touch lab7-1.asm  
[damityakov@fedora lab07]$
```

В

этот файл вставил код из листинга (рис. ??).

```

lab7-1.asm      [----]  8 L:[  1+15  16/ 20]  *(47
%include 'in_out.asm' ; подключение внешнего файла
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 ; Вывод на экран строки
call sprintf ; 'Сообщение № 1'
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintf ; 'Сообщение № 2'
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintf ; 'Сообщение № 3'
_end:
call quit ; вызов подпрограммы завершения

```

И получил такой результат (рис. ??).

```

[damityakov@fedora lab07]$ nasm -f elf lab7-1.asm
[damityakov@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[damityakov@fedora lab07]$ ./lab7-1
Сообщение № 2
Сообщение № 3
[damityakov@fedora lab07]$

```

Вставил код из другого листинга (рис. ??).

```

lab7-1.asm      [-----] 41 L:[ 1+21 22/ 22] *(670 / 670b) <EOF>
%include 'in_out.asm' ; подключение внешнего файла
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 ; Вывод на экран строки
call sprintfLF ; 'Сообщение № 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintfLF ; 'Сообщение № 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintfLF ; 'Сообщение № 3'
_end:
call quit ; вызов подпрограммы завершения

```

И получил новый результат (рис. ??).

```

[damityakov@fedora lab07]$ nasm -f elf lab7-1.asm
[damityakov@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[damityakov@fedora lab07]$ ./lab7-1
Сообщение № 2
Сообщение № 1
[damityakov@fedora lab07]$ 

```

Поменял код на этот (рис. ??).

```

lab7-1.asm      [----]  8 L:[ 1+12 13/ 23] *(367 / 682b) 0010 0x00A  [*][X]
%include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg1: DB 'Сообщение № 1',0
msg2: DB 'Сообщение № 2',0
msg3: DB 'Сообщение № 3',0
SECTION .text
GLOBAL _start
_start:
jmp _label3
_label1:
mov eax, msg1 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 1'
jmp _end
_label2:
mov eax, msg2 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 2'
jmp _label1
_label3:
mov eax, msg3 ; Вывод на экран строки
call sprintLF ; 'Сообщение № 3'
jmp _label2
_end:
call quit ; вызов подпрограммы завершения

```

Получив необходимый результат (рис. ??).

```

[damityakov@fedora lab07]$ nasm -f elf lab7-1.asm
[damityakov@fedora lab07]$ ld -m elf_i386 -o lab7-1 lab7-1.o
[damityakov@fedora lab07]$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
[damityakov@fedora lab07]$

```

Создал новый файл и вставил туда новый код (рис. ??).

```

lab7-2.asm      [-----]  9 L:[  1+ 4   5/ 49]  *(124 /1743b) 0010
#include 'in_out.asm'
section .data
msg1 db 'Введите B: ',0h
msg2 db "Наибольшее число: ",0h
A dd '20'
C dd '50'
section .bss
max resb 10
B resb 10
section .text
global _start
_start:
; ----- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ----- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'

```

Получил такие результаты попробовав несколько вводимых значений (рис. ??).

```

[damityakov@fedora lab07]$ touch lab7-2.asm
s
bash: s: команда не найдена...
[damityakov@fedora lab07]$ nasm -f elf lab7-2.asm
[damityakov@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[damityakov@fedora lab07]$ ./lab7-2
Введите B: 1
Наибольшее число: 50
[damityakov@fedora lab07]$ ./lab7-2
Введите B: 52
Наибольшее число: 52
[damityakov@fedora lab07]$ ./lab7-2
Введите B: 100
Наибольшее число: 100
[damityakov@fedora lab07]$

```

Открыл файл листинга и обнаружил там следующие 3 строки: 1 - номер строки 2 - адрес в памяти 3 - машинный код 4 - исходный текст программы 5 - комментарий (часть исходного текста команды)

(рис. ??).

```
46 0000015E E8ACFEFFFF    call sprint ; Вывод сообщения 'Наибольш
47 00000163 A1[00000000]    mov eax,[max]
48 00000168 E819FFFFF    call iprintLF ; Вывод 'max(A,B,C)'
49 0000016D E869FFFFF    call quit ; Выход
```

#fig:010 width=70%} Убрал один операнд из команды (рис. ??).

```
call sprint
; ----- Ввод 'B'
mov ecx
mov edx,10
call sread
```

И получил такой результат в листинге (рис. ??).

```
16                                ; ----- Ввод 'B'
17                                mov ecx
17                                ***** error: invalid combination of opcode an
18 000000F2 BA0A000000    mov edx,10
19 000000F7 F847FFFFF    call sread
```

Поменял проверку флага jg на jl в коде нахождения максимального числа и некоторые строки и получил программу на нахождение минимального числа (рис. ??).

```
[damityakov@fedora lab07]$ nasm -f elf lab7-2.asm
[damityakov@fedora lab07]$ ld -m elf_i386 -o lab7-2 lab7-2.o
[damityakov@fedora lab07]$ ./lab7-2
Введите B: 33
Наименьшее число: 33
[damityakov@fedora lab07]$
```

(рис. ??).


```

lab7-2.asm      [----]  7 L:[  2+10  12/ 49] *(205 /1743b) 0010 0x0
section .data
msg1 db 'Введите B: ',0h
msg2 db "Наименьшее число: ",0h
A dd '52'
C dd '40'
section .bss
min resb 10
B resb 10
section .text
global _start
_start:
; ----- Вывод сообщения 'Введите B: '
mov eax,msg1
call sprint
; ----- Ввод 'B'
mov ecx,B
mov edx,10
call sread
; ----- Преобразование 'B' из символа в число
mov eax,B
call atoi ; Вызов подпрограммы перевода символа в число
mov [B],eax ; запись преобразованного числа в 'B'
; ----- Записываем 'A' в переменную 'max'
mov ecx,[A] ; 'ecx = A'
mov [min],ecx ; 'max = A'
; ----- Сравниваем 'A' и 'C' (как символы)
cmp ecx,[C] ; Сравниваем 'A' и 'C'
jnl check_B ; если 'A>C', то переход на метку 'check_B',
mov ecx,[C] ; иначе 'ecx = C'
mov [min],ecx ; 'min = C'
; ----- Преобразование 'min(A,C)' из символа в число
check_B:
mov eax,min
call atoi ; Вызов подпрограммы перевода символа в число
mov [min],eax ; запись преобразованного числа в 'max'
; ----- Сравниваем 'min(A,C)' и 'B' (как числа)
mov ecx,[min]
cmp ecx,[B] ; Сравниваем 'min(A,C)' и 'B'
jnl fin ; если 'min(A,C)>B', то переход на 'fin',
mov ecx,[B] ; иначе 'ecx = B'
mov [min],ecx
; ----- Вывод результата
fin:
mov eax, msg2
call sprint ; Вывод сообщения 'Наибольшее число: '
mov eax,[min]
call iprintLF ; Вывод 'min(A,B,C)'
call quit ; Выход

```

Написал программу нахождения результата функции (рис. ??).

```
[damityakov@fedora lab07]$ nasm -f elf lab7-2t2.asm
[damityakov@fedora lab07]$ ld -m elf_i386 -o lab7-2t2 lab7-2t2.o
[damityakov@fedora lab07]$ ./lab7-2t2
Введите x: 1
Введите a: 4
Ответ: 2
[damityakov@fedora lab07]$ ./lab7-2t2
Введите x: 1
Введите a: 2
Ответ: 2
[damityakov@fedora lab07]$
```


(рис. ??).

```
%include 'in_out.asm'
section .data
msg1 db 'Введите x: ',0h
msg2 db 'Введите a: ',0h
msg3 db 'Ответ: ',0h
section .bss
x resb 10
a resb 10
res resb 10
section .text
global _start
_start:
mov eax,msg1
call sprint
; ----- Ввод 'x'
mov ecx,x
mov edx,10
call sread

; ----- Преобразование 'x' из символа в число
mov eax,x
call atoi ; Вызов подпрограммы перевода символа в число
mov [x],eax ; запись преобразованного числа в 'x'
; ----- Вывод сообщения 'Введите a: '
mov eax,msg2
call sprint
; ----- Ввод 'a'
mov ecx,a
mov edx,10
call sread

; ----- Преобразование 'a' из символа в число
mov eax,a
call atoi ; Вызов подпрограммы перевода символа в число
mov [a],eax ; запись преобразованного числа в 'a'
mov eax,[a]
mov ecx, 3
mul ecx
mov [res],eax
cmp eax,3
jl fin
mov eax,[x]
inc eax
mov [res], eax
; ----- Вывод результата
```

3 Выводы

Я изучил команды условного и безусловного переходов. Приобретел навыки написания программ с использованием переходов. Познакомился с назначением и структурой файла листинга.

Список литературы