

Отчет по лабораторной работе №7

Дисциплина: архитектура компьютера

Абуков Ислам Ренатович

Содержание

1 Цель работы	5
2 Задание	6
3 Теоретическое введение	7
4 Выполнение лабораторной работы	8
4.1 Реализация переходов в <i>NASM</i>	8
4.2 Изучение структуры файла листинга	1
	5

4.3 Задания для самостоятельной работы **18**

5 Выводы

Список иллюстраций

4.1 Создание каталога и файла для программы
4.2 Сохранение программы
4.3 Запуск программы
4.4 Изменение программы
4.5 Запуск изменененной программы
4.6 Изменение программы
4.7 Проверка изменений
4.8 Сохранение новой программы
4.9 Проверка программы из листинга
4.10 Проверка файла листинга
4.11 Удаление операнда из программы
4.12 Просмотр ошибки в файле листинга
4.13 Первая программа самостоятельной работы
4.14 Проверка работы первой программы
4.15 Вторая программа самостоятельной работы
4.16 Проверка работы второй программы

Цель работы

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

Задание

1. Реализация переходов в NASM
2. Изучение структуры файлов листинга
3. Самостоятельное написание программ по материалам лабораторной работы

Теоретическое введение

Для реализации ветвлений в ассемблере используются так называемые команды передачи управления или команды перехода. Можно выделить 2 типа переходов:

- условный переход – выполнение или не выполнение перехода в определенную точку программы в зависимости от проверки условия.
- безусловный переход – выполнение передачи управления в определенную точку программы без каких-либо условий.

Выполнение лабораторной работы

Реализация переходов в NASM

Создаю каталог для программ лабораторной работы №7

```
irabukov1@islamAbukov:~$ cd ~/work/arch-pc/lab07
irabukov1@islamAbukov:~/work/arch-pc/lab07$ mc
```

Копирую код из листинга в файл будущей программы.

The screenshot shows a terminal window with the title bar "mc [irabukov1@islamAbukov]:~/work/arch-pc/lab07". The main area displays assembly code for a program named "lab7-1.asm". The code includes sections for data and text, global entry point, and three labels for printing messages. The keyboard shortcut bar at the bottom provides various editing functions.

```
GNU nano 7.2 /home/irabukov1/work/arch-pc/lab07/lab7-1.asm
%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 sprintLF

_label2:
    mov eax, msg2
    call sprintLF

_label3:
    mov eax, msg3
    call sprintLF

_end:
    call quit


```

Keyboard shortcuts:

- Справка (F1)
- Записать (Ctrl+S)
- Поиск (Ctrl+F)
- Вырезать (Ctrl+X)
- Выполнить (Ctrl+T)
- Позиция (Ctrl+C)
- Выход (Ctrl+Q)
- ЧитФайл (Ctrl+R)
- Замена (Ctrl+H)
- Вставить (Ctrl+V)
- Выровнять (Ctrl+J)
- К строке (Ctrl+L)

При запуске программы я убедился в том, что неусловный переход действительно изменяет порядок выполнения инструкций.

The screenshot shows a terminal window with two tabs. The active tab displays the following command-line session:

```
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-1
bash: ./lab7-1: Нет такого файла или каталога
irabukov1@islamAbukov:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
irabukov1@islamAbukov:~/work/arch-pc/lab07$
```

Изменяю программу таким образом, чтобы поменялся порядок выполнения функций.

```
mc [irabukov1@islamAbukov]:~/work/arch-pc/lab07          irabukov1@islamAbukov: ~/work/arch-pc/lab07
GNU nano 7.2          /home/irabukov1/work/arch-pc/lab07/lab7-1.asm *
%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 sprintLF
    jmp _end

_label2:
    mov eax, msg2
    call sprintLF
    jmp _label1

_label3:
    mov eax, msg3
    call sprintLF

_end:
    call quit

^G Справка ^O Записать ^W Поиск      ^K Вырезать ^T Выполнить ^C Позиция
^X Выход ^R ЧитФайл ^V Замена      ^U Вставить ^J Выровнять ^/ К строке
```

Запускаю программу и проверяю, что примененные изменения верны.

```
нение из буфера обмена. irabukov1@islamAbukov: ~/work/arch-pc/lab07
mc [irabukov1@islamAbukov]:~/work/arch-pc... ×      irabukov1@islamAbukov: ~/work/arch-pc/lab07 ×      v
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-1
bash: ./lab7-1: Нет такого файла или каталога
irabukov1@islamAbukov:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 3
irabukov1@islamAbukov:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 2
Сообщение № 1
irabukov1@islamAbukov:~/work/arch-pc/lab07$
```

Теперь изменяю текст программы так, чтобы все три сообщения вывелись в обратном порядке.

```
ение из буфера обмена. : [irabukov1@islamAbukov]:~/work/arch-pc/lab07   Q  ⌂  ×
mc [irabukov1@islamAbukov]:~/work/arch-...  ×  irabukov1@islamAbukov: ~/work/arch-pc/la...  ×  ↻
GNU nano 7.2      /home/irabukov1/work/arch-pc/lab07/lab7-1.asm
%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 _label3

_label1:
    mov eax, msg1
    call sprintLF
    jmp _end

_label2:
    mov eax, msg2
    call sprintLF
    jmp _label1

_label3:
    mov eax, msg3
    call sprintLF
    jmp _label2

_end:
    call quit
[ Прочитано 30 строк ]
^G Справка  ^O Записать  ^W Поиск  ^K Вырезать  ^T Выполнить
^X Выход  ^R ЧитФайл  ^\ Замена  ^U Вставить  ^J Выровнять
```

Работа выполнена корректно, программа в нужном мне порядке выводит сообщения.

```
irabukov1@islamAbukov:~/work/arch-pc/lab07$ nasm -f elf lab7-1.asm
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab6-1 lab6-1.o
ld: невозможно найти lab6-1.o: Нет такого файла или каталога
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-1 lab7-1.o
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
irabukov1@islamAbukov:~/work/arch-pc/lab07$
```

Создаю новый рабочий файл и вставляю в него код из следующего листинга.

```
GNU nano 7.2      /home/irabukov1/work/arch-pc/lab07/lab7-2.asm
%include 'in_out.asm'

section .data
msg1 db 'Введите В: ',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:

    mov eax,msg1
    call sprint
    mov ecx,B
    mov edx,10
    call sread

    mov eax,B
    call atoi
    mov [B],eax

    mov ecx,[A]
    mov [max],ecx

    cmp ecx,[C]
    jg check_B
```

[Прочитано 50 строк]

^G Справка **^O Записать** **^W Поиск** **^K Вырезать** **^T Выполнить**
^X Выход **^R ЧитФайл** **^V Замена** **^U Вставить** **^J Выровнять**

Программа выводит значение переменной с максимальным значением, проверяю работу программы с разными входными данными.

```
irabukov1@islamAbukov:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 25
Наибольшее число: 50
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab-2
bash: ./lab-2: Нет такого файла или каталога
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 60
Наибольшее число: 60
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 15
Наибольшее число: 50
irabukov1@islamAbukov:~/work/arch-pc/lab07$
```

Задания для самостоятельной работы

Искренне не понимаю, какой вариант я должен был получить во время 7 лабораторной работы, поэтому буду использовать свой вариант – пятнадцатый - из предыдущей лабораторной работы. Возвращаю операнд к функции в программе и изменяю ее так, чтобы она выводила переменную с наименьшим значением.

```

GNU nano 7.2      /home/irabukov1/work/arch-pc/lab07/lab7-2.asm *
%include "in_out.asm"

SECTION .data
    msg1 db 'Введите В: ', 0
    msg2 db 'Наибольшее число: ', 0

    A    dd 32
    C    dd 54

SECTION .bss
    max resd 1      ; максимум хранится тут (4 байта)
    B    resd 1      ; тут число из ввода

SECTION .text
    global _start

_start:
    ; вывод приглашения и ввод В (как строка)
    mov eax, msg1
    call sprint

    mov ecx, B          ; адрес В для строки
    mov edx, 10         ; максимум 10 символов
    call sread

    ; перевод строки в число
    mov eax, B
    call atoi
    mov [B], eax

```

$\wedge\text{G}$ Справка $\wedge\text{O}$ Записать $\wedge\text{W}$ Поиск $\wedge\text{K}$ Вырезать $\wedge\text{T}$ Выполнить
 $\wedge\text{X}$ Выход $\wedge\text{R}$ ЧитФайл $\wedge\backslash$ Замена $\wedge\text{U}$ Вставить $\wedge\text{J}$ Выровнять

Проверяю корректность написания первой программы.

```

irabukov1@islamAbukov:~/work/arch-pc/lab07$ nasm -f elf lab7-2.asm
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-2 lab7-2.o
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-2
Введите В: 6
Наибольшее число: 54
irabukov1@islamAbukov:~/work/arch-pc/lab07$

```

Пишу программу, которая будет вычислять значение заданной функции согласно моему варианту для введенных с клавиатурых переменных а и х.

```

GNU nano 7.2      /home/irabukov1/work/arch-pc/lab07/lab7-3.asm *
%include "in_out.asm"

SECTION .data
msg_x: db "Vvedite znachenie peremennoj x: ",0
msg_a: db "Vvedite znachenie peremennoj a: ",0
msg_res: db "Rezultat: ",0

SECTION .bss
x: resb 80
a: resb 80

SECTION .text
global _start

_start:
; ввод x
    mov eax, msg_x
    call sprint
    mov ecx, x
    mov edx, 80
    call sread
    mov eax, x
    call atoi
    mov ebx, eax        ; ebx = x

; ввод a
    mov eax, msg_a
    call sprint
    mov ecx, a
    mov edx, 80

```

^{^G} Справка ^{^O} Записать ^{^W} Поиск ^{^K} Вырезать ^{^T} Выполнить
^{^X} Выход ^{^R} ЧитФайл ^{^V} Замена ^{^U} Вставить ^{^J} Выровнять

Транслирую и компоную файл, запускаю и проверяю работу программы для различных значений а и х.

```

irabukov1@islamAbukov:~/work/arch-pc/lab07$ nasm -f elf lab7-3.asm
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ld -m elf_i386 -o lab7-3 lab7-3.o
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-3
Vvedite znachenie peremennoj x: 2
Vvedite znachenie peremennoj a: 3
Rezultat: 42
irabukov1@islamAbukov:~/work/arch-pc/lab07$ ./lab7-3
Vvedite znachenie peremennoj x: 4
Vvedite znachenie peremennoj a: 2
Rezultat: 42
irabukov1@islamAbukov:~/work/arch-pc/lab07$ 

```

Выводы

При выполнении лабораторной работы я изучил команды условных и безусловных переходов, а также приобрел навыки написания программ с использованием переходов, познакомился с назначением и структурой файлов листинга.