

Лабораторная работа №8

**Команды безусловного и условного переходов в Nasm.
Программирование ветвлений.**

Лушин Артем Андреевич

Содержание

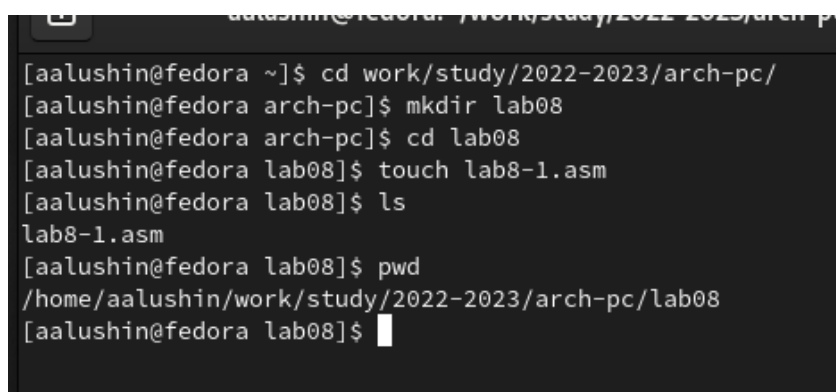
1	Цель работы	3
2	Выполнение лабораторной работы	4
3	Самостоятельная работа.	14
4	Вывод	19

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

- 1) Я создал каталог lab8 и внутри создал файл lab8-1.asm



```
[aalushin@fedora ~]$ cd work/study/2022-2023/arch-pc/  
[aalushin@fedora arch-pc]$ mkdir lab08  
[aalushin@fedora arch-pc]$ cd lab08  
[aalushin@fedora lab08]$ touch lab8-1.asm  
[aalushin@fedora lab08]$ ls  
lab8-1.asm  
[aalushin@fedora lab08]$ pwd  
/home/aalushin/work/study/2022-2023/arch-pc/lab08  
[aalushin@fedora lab08]$
```

Рис. 2.1: Создание файла lab8-1.asm

- 2) Я ввел в файл текст программы и запустил его.

```
/home/aalushin/work/study/2022-2023/asm/lab8-1.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

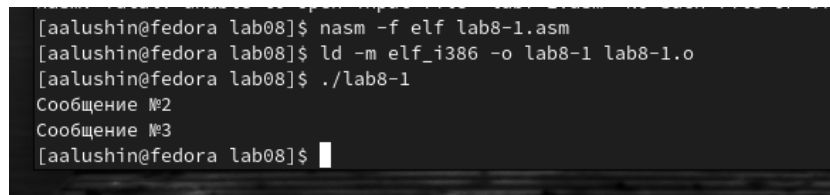
_label2:
mov eax,msg2
call sprintLF

_label3:
mov eax,msg3
call sprintLF

_end:
call quit
```

Рис. 2.2: Текст в файле lab8-1.asm

3) Я создал исполняемый файл и запустил его. Результат соответствовал нужному.

A terminal window with a dark background and light-colored text. The text shows a user named 'aalushin' at a 'fedora' machine in the 'lab08' directory. They run 'nasm -f elf lab8-1.asm', then 'ld -m elf_i386 -o lab8-1 lab8-1.o', and finally './lab8-1'. The output shows two messages: 'Сообщение №2' and 'Сообщение №3'. The prompt returns to '[aalushin@fedora lab08]\$' with a cursor.

```
[aalushin@fedora lab08]$ nasm -f elf lab8-1.asm
[aalushin@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[aalushin@fedora lab08]$ ./lab8-1
Сообщение №2
Сообщение №3
[aalushin@fedora lab08]$
```

Рис. 2.3: Запуск программы lab8-1

4) Я изменил текст программы чтобы выводился нужный ответ и создал исполняемый файл.

```

#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

```

Рис. 2.4: Изменение текста

```
[aalushin@fedora lab08]$ nasm -f elf lab8-1.asm
[aalushin@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[aalushin@fedora lab08]$ ./lab8-1
Сообщение №2
Сообщение №1
```

Рис. 2.5: Проверка работы программы

5) Я изменил текст программы чтобы сначала выводило сообщение 3, затем 2, затем 1.


```

#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 sprintf
jmp _end

_label2:
mov eax,msg2
call sprintf
jmp _label1

_label3:
mov eax,msg3
call sprintf
jmp _label2
_end:
call quit

```

Рис. 2.6: Изменение текста

6) Запустил программу и проверил ее работу.

```
[aalushin@fedora lab08]$ nasm -f elf lab8-1.asm
[aalushin@fedora lab08]$ ld -m elf_i386 -o lab8-1 lab8-1.o
[aalushin@fedora lab08]$ ./lab8-1
Сообщение №3
Сообщение №2
Сообщение №1
```

Рис. 2.7: Запуск программы

7) Я создал файл lab8-2.asm и написал текст программы.

```

%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:

mov eax,msg1
call sprint

mov ecx,B
mov edx,10
call sread

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

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

cmp ecx,[C]
jg check_B
mov ecx,[C]
mov [max],ecx

check_B:
mov eax,max
call atoi
mov [max],eax

mov ecx,[max]
cmp ecx,[B]
jg fin

```

Рис. 2.8: Текст программы для сравнения чисел

8) Я ввел два разных числа чтобы проверить как работает программа.

```
[aalushin@fedora lab08]$ nasm -f elf lab8-2.asm
[aalushin@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
ld: невозможно найти lab8-2.0: Нет такого файла или каталога
[aalushin@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[aalushin@fedora lab08]$ ./lab8-2
Введите B65
Наибольшее число:65
[aalushin@fedora lab08]$ ./lab8-2
Введите B44
Наибольшее число:50
```

Рис. 2.9: Программа для сравнения чисел

9) Я создал файл листинга lab8-2.lst и открыл его.

```
[aalushin@fedora lab08]$ nasm -f elf -l lab8-2.lst lab8-2.asm
[aalushin@fedora lab08]$ mcedit lab8-2.lst
[aalushin@fedora lab08]$
```

Рис. 2.10: Файл листинга lab8-2.lst

10) Проанализировав файл, я понял как он работает и какие значения выводит.

11) Эта строка находится на 24 месте, ее адрес “00000101”, Машинный код - B8 [0A000000], а mov eax,B - исходный текст программы, означающий что в регистр eax мы вносим значения переменной B.

```
24 00000101 B8[0A000000]          mov eax,B
```

Рис. 2.11: Объяснения первой строки

2) Эта строка находится на 38 месте, ее адрес “00000134”, Машинный код - E863FFFFFF, а call atoi - исходный текст программы, означающий что символ лежащий в строке выше переводится в число.

```
37 0000012F B8[00000000]          mov eax,max
38 00000134 E863FFFFFF          call atoi
```

Рис. 2.12: Объяснения второй строки

- 3) Эта строка находится на 50 месте, ее адрес “00000162”, Машинный код - A1[00000000], а `mov eax,[max]` - исходный текст программы, означающий что число хранившееся в переменной `max` записывается в регистр `eax`.



Рис. 2.13: Объяснения третьей строки

- 11) В строке `mov eax,max` я убрал `max` и попробовал создать файл. Выдало ошибку, так как для программы нужно два операнда.

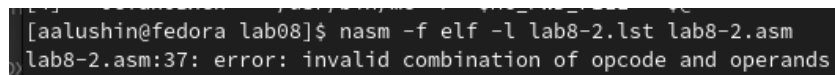


Рис. 2.14: Создание файла без одного операнда

- 12) В файле листинга показывает где именно ошибка и с чем она связана.

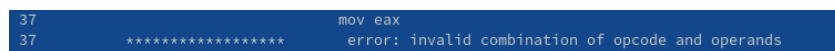


Рис. 2.15: Файл листинга без одного операнда

3 Самостоятельная работа.

- 1) Я написал программу для нахождения меньшего из трех чисел. Для большего удобства я сделал ввод чисел с клавиатуры. У меня первый вариант поэтому числа были :17,23,45. Программа вывела меньшее из этих чисел.

```

%include 'in_out.asm'

SECTION .data
A1 DB 'Введите число A: ',0h
B1 DB 'Введите число B: ',0h
C1 DB 'Введите число C: ',0h
otv DB 'Наименьшее число: ',0h
SECTION .bss
min RESB 20
A RESB 20
B RESB 20
C RESB 20

SECTION .text
GLOBAL _start
_start:

mov eax,A1
call sprint

mov ecx,A
mov edx,20
call sread

mov eax, A
call atoi
mov [A],eax

xor eax,eax

mov eax,B1
call sprint

mov ecx,B
mov edx,20
call sread

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

xor eax,eax

mov ecx, [A]

```

Рис. 3.1: Текст программы


```

#include 'in_out.asm'

SECTION .data
prim1 DB '2a-x ,x<a' ,0
prim2 DB '8, x=>a',0
X1 DB 'Введите значение X:',0
A1 DB 'Введите значение a:',0
otv DB 'Ответ: ',0

SECTION .bss
X RESB 20
A RESB 20
F RESB 20
SECTION .text
GLOBAL _start
_start:

mov eax,prim1
call sprintf
mov eax,prim2
call sprintf

mov eax,X1
call sprintf

mov ecx,X
mov edx,10
call sread

mov eax,X
call atoi
mov [X],eax

mov eax,A1
call sprintf

mov ecx,A
mov edx,10
call sread

mov eax,A
call atoi
mov [A],eax

```

Рис. 3.3: Текст программы

```
[aalushin@fedora lab08]$ nasm -f elf 3.asm
[aalushin@fedora lab08]$ ld -m elf_i386 -o 3 3.o
[aalushin@fedora lab08]$ ./3
2a-x ,x<a
8, x=>a
Введите значение X:1
Введите значение a:2
Ответ: 3
[aalushin@fedora lab08]$ ./3
2a-x ,x<a
8, x=>a
Введите значение X:2
Введите значение a:1
Ответ: 8
```

Рис. 3.4: Проверка работы программы

4 Вывод

Я изучил команды условного и безусловного перехода. Приобрел навыки написания программ с переходами.