

Лабораторная работа № 8

Команды безусловного и условного переходов в Nasm. Программирование ветвлений.

Шулуужук Айраана В.

Содержание

1	Цель работы	1
2	Выполнение лабораторной работы	1
2.1	Реализация переходов в NASM	1
2.2	Изучение структуры файлы листинга	3
3	Задания для самостоятельной работы	3
3.1	Выполнение самостоятельной работы.....	4
4	Выводы.....	6

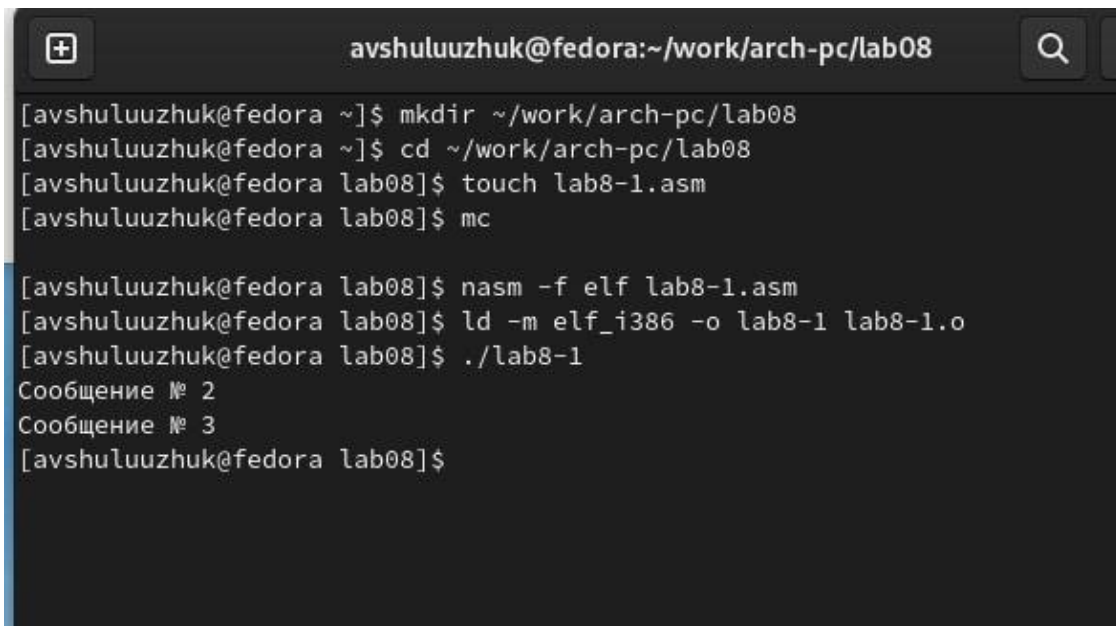
1 Цель работы

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

2 Выполнение лабораторной работы

2.1 Реализация переходов в NASM

1. Создадим файл lab8-1.asm и введем текст программы из листинга. Проверим его работу (рис. 1)



```
avshuluuzhuk@fedora:~/work/arch-pc/lab08
[avshuluuzhuk@fedora ~]$ mkdir ~/work/arch-pc/lab08
[avshuluuzhuk@fedora ~]$ cd ~/work/arch-pc/lab08
[avshuluuzhuk@fedora lab08]$ touch lab8-1.asm
[avshuluuzhuk@fedora lab08]$ mc

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

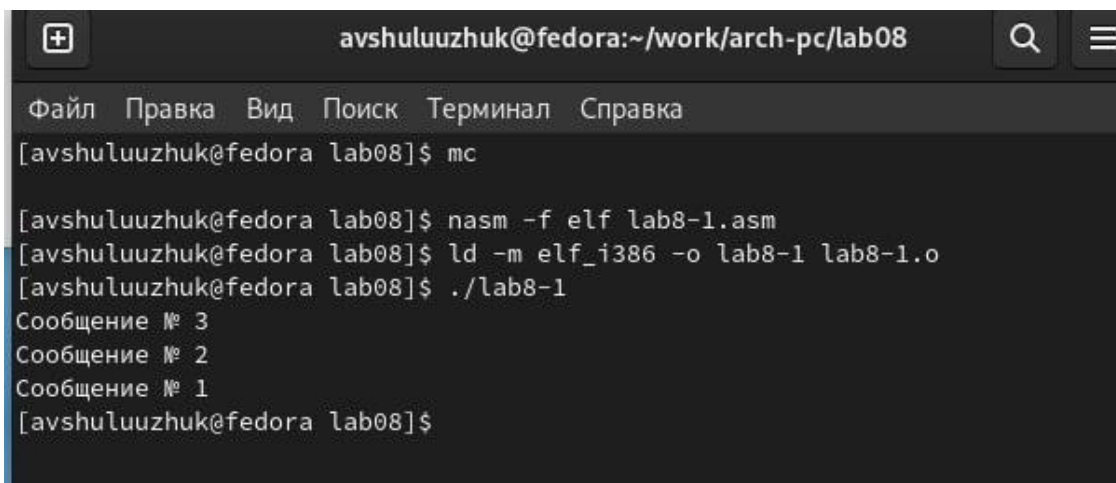
Рис. 1: результат работы программы

2. Изменим текст программы так, чтобы вывод программы был: (рис. 2)

Сообщение No 3

Сообщение No 2

Сообщение No 1



```
avshuluuzhuk@fedora:~/work/arch-pc/lab08
Файл  Правка  Вид  Поиск  Терминал  Справка
[avshuluuzhuk@fedora lab08]$ mc

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

Рис. 2: результат работы программы

3. Создадим новый файл и введем в него из листинга текст программы, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных (рис. 3)

```

[avshuluuzhuk@fedora lab08]$ mc

[avshuluuzhuk@fedora lab08]$ ./lab8-2
введите B: 40
наибольшее число: 50
[avshuluuzhuk@fedora lab08]$

```

Рис. 3: результат работы программы

2.2 Изучение структуры файла листинга

1. Создадим файл листинга для программы из файла lab8-2.asm. Откроем этот файл с помощью текстового редактора
2. Удалим в любой инструкции с двумя операндами один его операнд. Далее выполним трансляцию с получением файла листинга.

В итоге команда выводит ошибку. А в листинге появляется строка с ошибкой (рис. 4)

```

lab8-2.lst [-----] 49 L: [202+10 212/229] *(12724/13600b) 0010 0x00A[*] [X]
27.....
28 00000111 8B0D[36000000]      mov ecx,[A]
29 00000117 890D[00000000]      mov [max],ecx
30.....
31 0000011D 3B0D[3A000000]      cmp ecx,[C]
32 00000123 7F0C                      jg check_B
33 00000125 8B0D[3A000000]      mov ecx,[C]
34 0000012B 890D[00000000]      mov [max],ecx
35.....
36                      check_B:
37                      mov eax
37          ***** error: invalid combination of opcode and operand
38 00000131 E866FFFFFF      call atoi
39 00000136 A3[00000000]      mov [max],eax
40.....
41 0000013B 8B0D[00000000]      mov ecx,[max]
42 00000141 3B0D[0A000000]      cmp ecx,[B]
43 00000147 7F0C                      jg fin
44 00000149 8B0D[0A000000]      mov ecx,[B]
45 0000014F 890D[00000000]      mov [max],ecx
46.....
47                      fin:

```

Рис. 4: результат работы программы

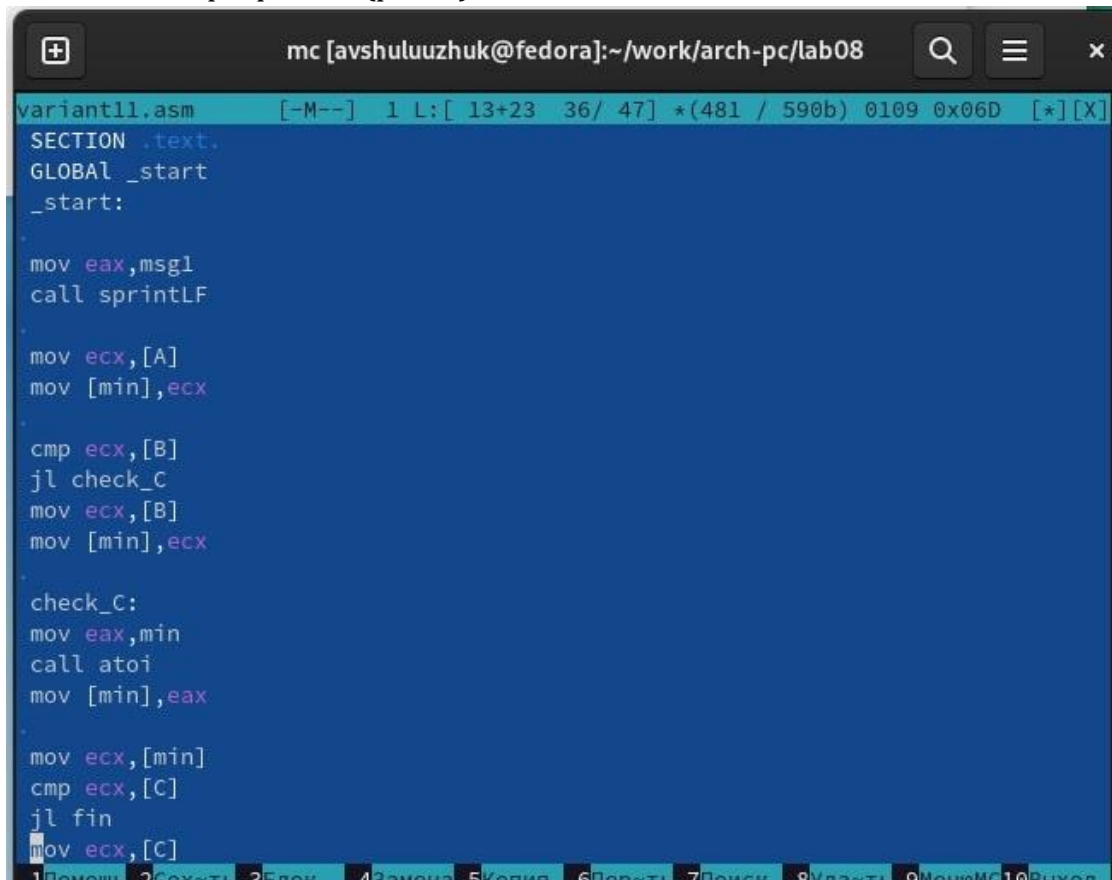
3 Задания для самостоятельной работы

1. Написать программу для нахождения наименьшей из 3 целочисленных переменных A, B, C

2. Написать программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений

3.1 Выполнение самостоятельной работы

1. Создадим файл `variant11.asm` для первой программы (11 вариант). Введем текст программы (рис. 5)



```
variant11.asm  [-M--]  1  L: [ 13+23  36/ 47]  *(481 / 590b)  0109 0x06D  [*] [X]
SECTION .text
GLOBAL _start
_start:

mov eax,msg1
call sprintf

mov ecx,[A]
mov [min],ecx

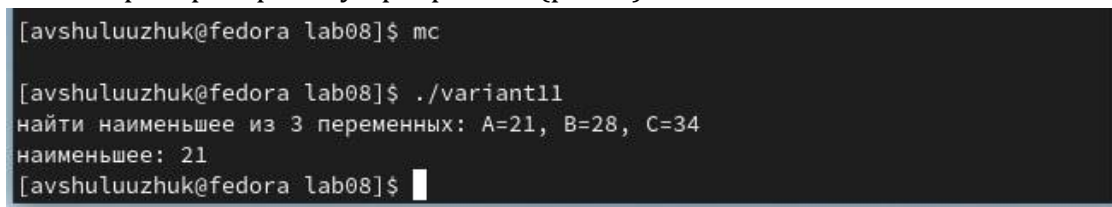
cmp ecx,[B]
jl check_C
mov ecx,[B]
mov [min],ecx

check_C:
mov eax,min
call atoi
mov [min],eax

mov ecx,[min]
cmp ecx,[C]
jl fin
mov ecx,[C]
```

Рис. 5: текст программы для определения наименьшей из 3 переменных

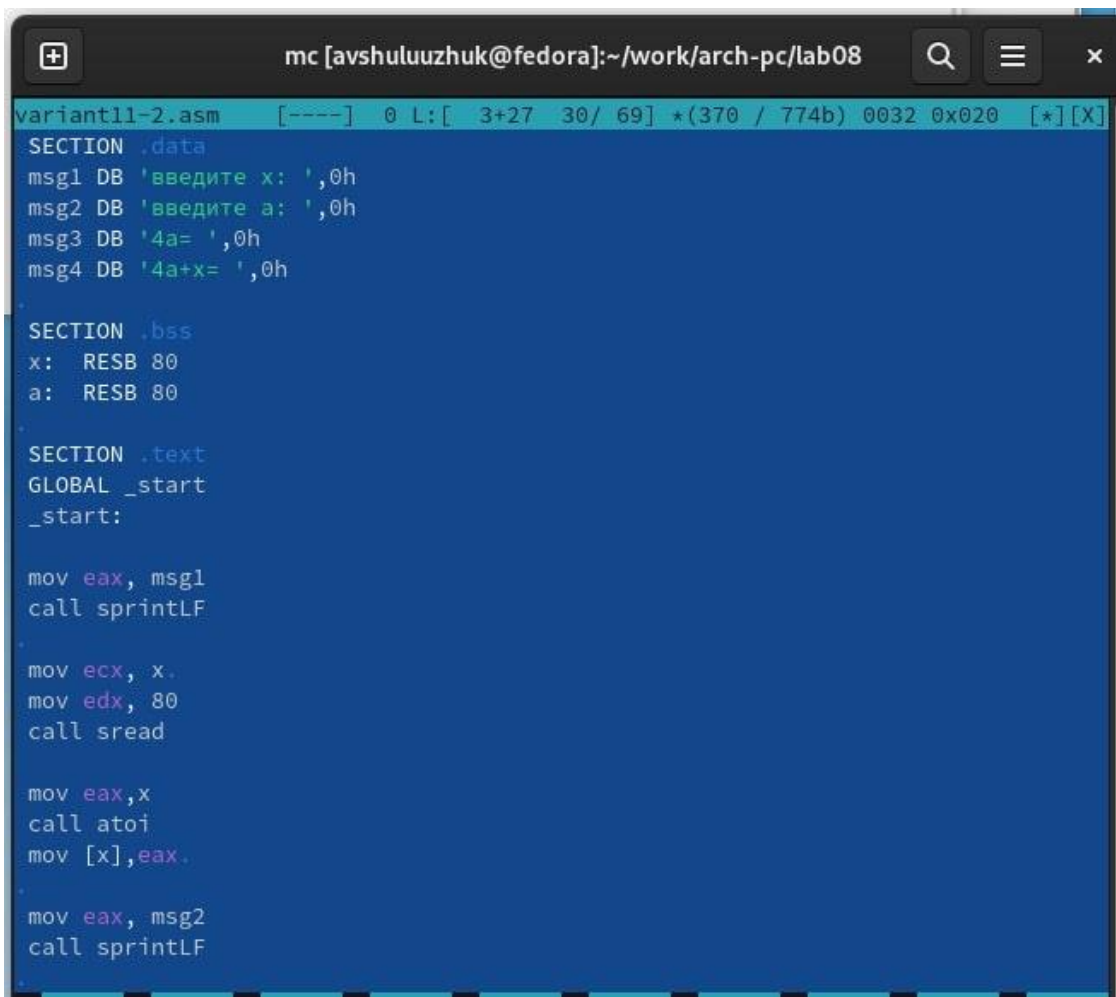
2. Проверим работу программы (рис. 6)



```
[avshuluuzhuk@fedora lab08]$ mc
[avshuluuzhuk@fedora lab08]$ ./variant11
найти наименьшее из 3 переменных: A=21, B=28, C=34
наименьшее: 21
[avshuluuzhuk@fedora lab08]$
```

Рис. 6: результат вывода программы

3. Создадим новый файл `variant11-2` для второй программы, которая для введенных переменных вычисляет значения (11 вариант) (рис. 7)



```
variant11-2.asm [----] 0 L: [ 3+27 30/ 69] *(370 / 774b) 0032 0x020 [*][X]
SECTION .data
msg1 DB 'введите x: ',0h
msg2 DB 'введите a: ',0h
msg3 DB '4a= ',0h
msg4 DB '4a+x= ',0h

SECTION .bss
x: RESB 80
a: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg1
call sprintf

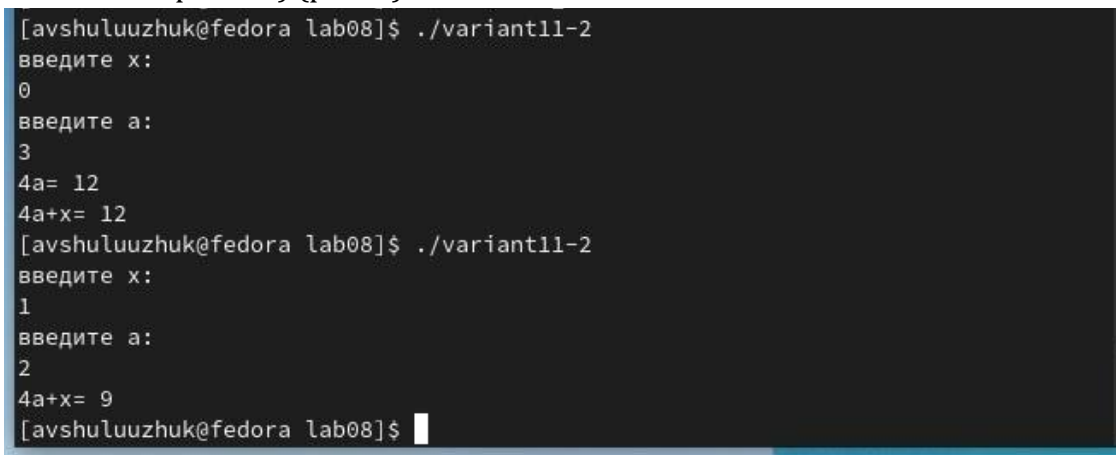
mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi
mov [x], eax

mov eax, msg2
call sprintf
```

Рис. 7: текст программы для вычисления значений

4. Скомпилируем файл и проверим работу для значений x и a (берем значения из 11 варианта) (рис. 8)



```
[avshuluuzhuk@fedora lab08]$ ./variant11-2
введите x:
0
введите a:
3
4a= 12
4a+x= 12
[avshuluuzhuk@fedora lab08]$ ./variant11-2
введите x:
1
введите a:
2
4a+x= 9
[avshuluuzhuk@fedora lab08]$
```

Рис. 8: результат вывода программы

4 Выводы

Были изучены программы для условного и безусловного переходов. Приобретены навыки написания программ с использованием переходов. Ознакомились с назначением и структурой файла листина