

Отчёта по лабораторной работе 8

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

Герра Гарсия Паола Валентина - НКАбд-05-22

-

Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	24
	Список литературы	25

Список иллюстраций

4.1	Файл lab8-1.asm:	9
4.2	Программа lab8-1.asm:	10
4.3	Файл lab8-1.asm:	11
4.4	Программа lab8-1.asm:	12
4.5	Файл lab8-1.asm.....	12
4.6	Программа lab8-1.asm.....	13
4.7	Файл lab8-2.asm.....	14
4.8	Программа lab8-2.asm.....	15
4.9	Файл листинга lab8-2	16
4.10	ошибка трансляции lab8-2.....	18
4.11	файл листинга с ошибкой lab8-2	19
4.12	Файл lab8-3.asm.....	20
4.13	Программа lab8-3.asm.....	21
4.14	Файл lab8-4.asm.....	22
4.15	Программа lab8-4.asm.....	23

Список таблиц

1 Цель работы

Целью работы является изучение команд условного и безусловного переходов. Приобретение навыков написания программ с использованием переходов. Знакомство с назначением и структурой файла листинга.

2 Задание

1. Изучите примеры программ.
2. Изучите файл листинга.
3. Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и c . Значения переменных выбрать из табл. 8.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу
4. Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений. Вид функции $f(x)$ выбрать из таблицы 8.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений X и a из 8.6.

3 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.

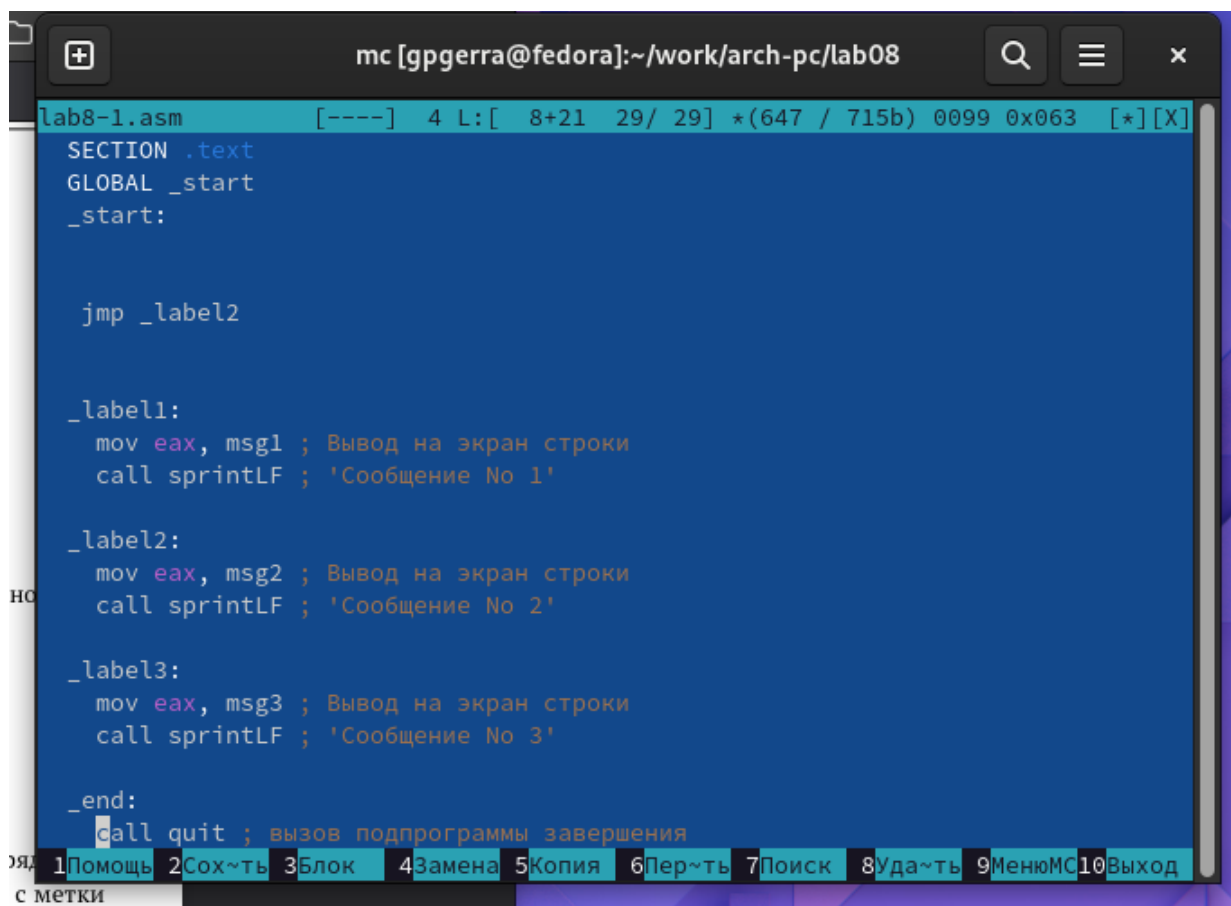
Листинг (в рамках понятийного аппарата NASM) — это один из выходных файлов, создаваемых транслятором. Он имеет текстовый вид и нужен при отладке программы, так как кроме строк самой программы он содержит дополнительную информацию.

4 Выполнение лабораторной работы

1. Создайте каталог для программ лабораторной работы № 8, перейдите в него и создайте файл lab8-1.asm
2. Инструкция jmp в NASM используется для реализации безусловных переходов. Рассмотрим пример программы с использованием инструкции jmp. Введите в файл lab8-1.asm текст программы из листинга 8.1. (рис. 4.1)



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



```
lab8-1.asm  [----]  4  L: [  8+21  29/ 29]  *(647 / 715b)  0099  0x063  [*] [X]
SECTION .text
GLOBAL _start
_start:

    jmp _label2

_label1:
    mov eax, msg1 ; Вывод на экран строки
    call sprintf ; 'Сообщение No 1'

_label2:
    mov eax, msg2 ; Вывод на экран строки
    call sprintf ; 'Сообщение No 2'

_label3:
    mov eax, msg3 ; Вывод на экран строки
    call sprintf ; 'Сообщение No 3'

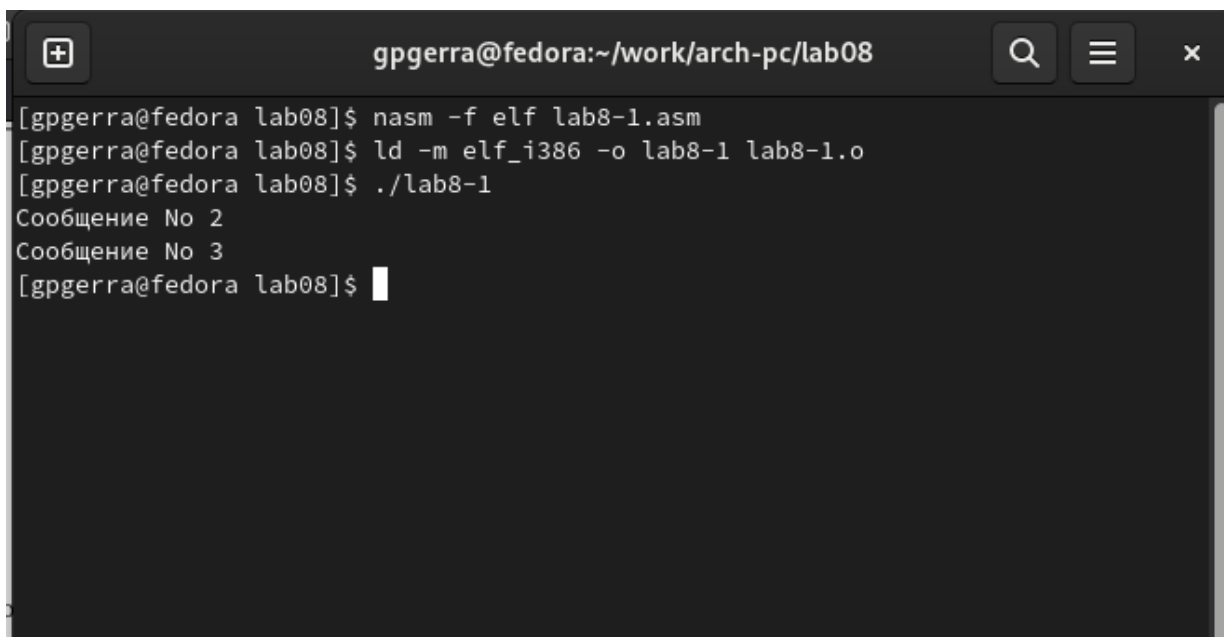
_end:
    call quit ; вызов подпрограммы завершения
```

1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюMC 10Выход

с метки

Рис. 4.1: Файл lab8-1.asm:

Создайте исполняемый файл и запустите его. (рис. 4.2)

A terminal window with a dark background. The title bar shows the user 'gpgerra@fedora' and the directory '~/work/arch-pc/lab08'. The terminal contains the following text:

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

Рис. 4.2: Программа lab8-1.asm:

Инструкция `jmp` позволяет осуществлять переходы не только вперед но и назад. Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого в текст программы после вывода сообщения № 2 добавим инструкцию `jmp` с меткой `_label1` (т.е. переход к инструкциям вывода сообщения № 1) и после вывода сообщения № 1 добавим инструкцию `jmp` с меткой `_end` (т.е. переход к инструкции `call quit`). Измените текст программы в соответствии с листингом 8.2. (рис. 4.3, 4.4)

```
lab8-1.asm [----] 0 L: [ 1+ 1 2/ 29] *(85 / 715b) 0010 0x00A [*] [X]
#include 'in_out.asm' ; подключение внешнего файла

SECTION .data
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0

SECTION .text
GLOBAL _start
_start:

    jmp _label2

_label1:
    mov eax, msg1 ; Вывод на экран строки
    call sprintfLF ; 'Сообщение No 1'

_label2:
    mov eax, msg2 ; Вывод на экран строки
    call sprintfLF ; 'Сообщение No 2'
```

поряд 1Помощь 2Сохранить 3Блок 4Замена 5Копия 6Перейти 7Поиск 8Удалить 9МенюМС10Выход
ая с метки

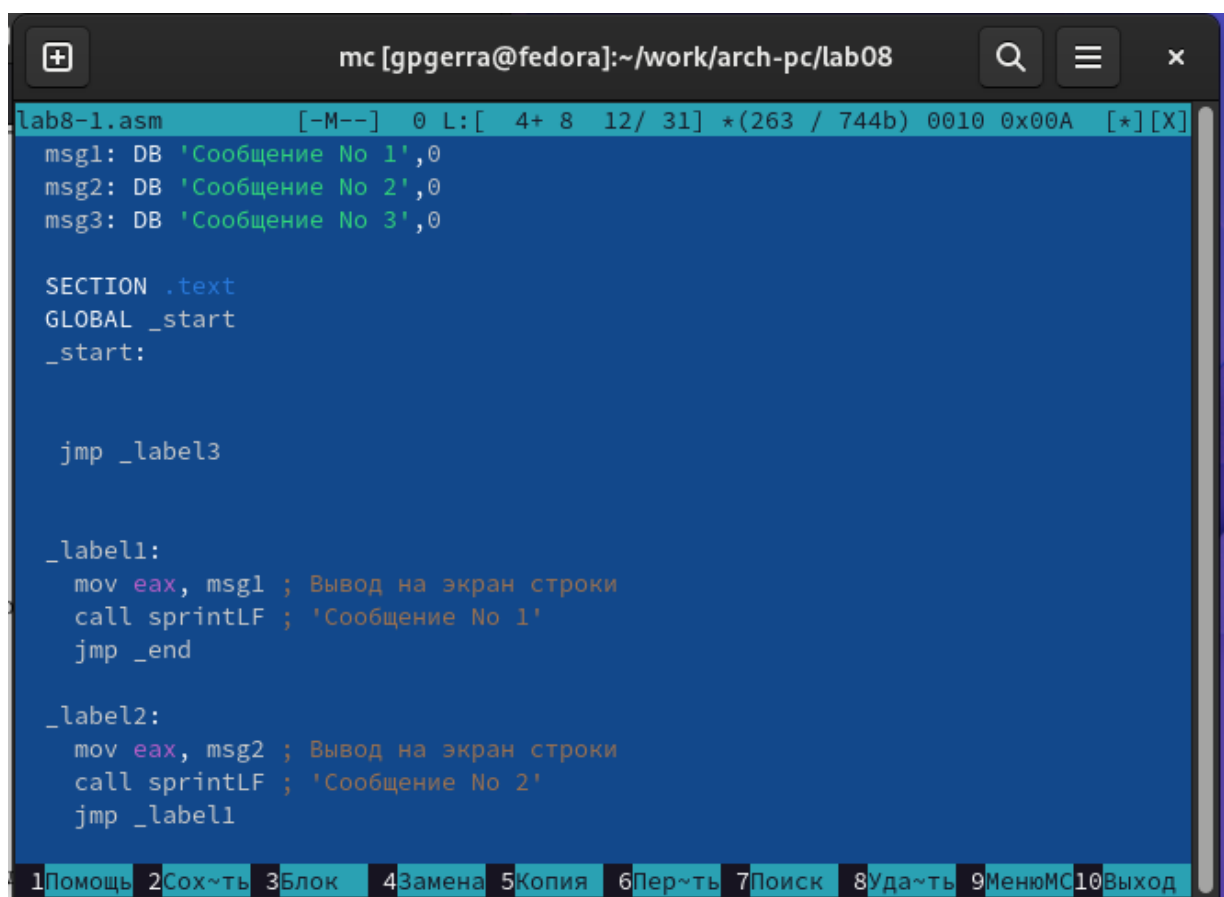
Рис. 4.3: Файл lab8-1.asm:

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

Рис. 4.4: Программа lab8-1.asm:

Изменить текст программы добавив или изменив инструкции `jmp`, чтобы вывод программы был следующим (рис. 4.5, 4.6):

Сообщение № 3
Сообщение № 2
Сообщение № 1



```
lab8-1.asm [-M--] 0 L: [ 4+ 8 12/ 31] *(263 / 744b) 0010 0x00A [*][X]
msg1: DB 'Сообщение No 1',0
msg2: DB 'Сообщение No 2',0
msg3: DB 'Сообщение No 3',0

SECTION .text
GLOBAL _start
_start:

    jmp _label3

_label1:
    mov eax, msg1 ; Вывод на экран строки
    call sprintf ; 'Сообщение No 1'
    jmp _end

_label2:
    mov eax, msg2 ; Вывод на экран строки
    call sprintf ; 'Сообщение No 2'
    jmp _label1

_label3:
    mov eax, msg3 ; Вывод на экран строки
    call sprintf ; 'Сообщение No 3'
    jmp _end

_end:
```

Рис. 4.5: Файл lab8-1.asm

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

Рис. 4.6: Программа lab8-1.asm

3. Использование инструкции `jmp` приводит к переходу в любом случае. Однако, часто при написании программ необходимо использовать условные переходы, т.е. переход должен происходить если выполнено какое-либо условие. В качестве примера рассмотрим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A, B и C. Значения для A и C задаются в программе, значение B вводится с клавиатуры. Создайте исполняемый файл и проверьте его работу для разных значений B. (рис. 4.7, 4.8)

```
lab8-2.asm [----] 20 L:[ 1+ 0 1/ 49] *(20 /1851b) 0039 0x027 [*][X]
#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
```

Рис. 4.7: Файл lab8-2.asm

```
[gpgerra@fedora lab08]$ nasm -f elf lab8-2.asm
[gpgerra@fedora lab08]$ ld -m elf_i386 -o lab8-2 lab8-2.o
[gpgerra@fedora lab08]$ ./lab8-1
Сообщение No 3
Сообщение No 2
Сообщение No 1
[gpgerra@fedora lab08]$
[gpgerra@fedora lab08]$ ./lab8-2
Введите B:
Наибольшее число: 50
[gpgerra@fedora lab08]$
```

Рис. 4.8: Программа lab8-2.asm

4. Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ -l и задав имя файла листинга в командной строке. Создайте файл листинга для программы из файла lab8-2.asm (рис. 4.9)

```
gpgerra@fedora:~/work/arch-pc/lab08 — mcedit lab8-2.lst
lab8-2.lst [-M--] 0 L: [ 1+ 0 1/225] *(0 /14566b) 0032 0x020 [*][X]
1 %include 'in_out.asm'
2 <1> ;----- slen -----
3 <1> ; Функция вычисления длины сообщения
4 <1> slen:.....
5 00000000 53 <1> push ebx.....
6 00000001 89C3 <1> mov ebx, eax.....
7 <1>.....
8 <1> nextchar:.....
9 00000003 803800 <1> cmp byte [eax], 0...
10 00000006 7403 <1> jz finished.....
11 00000008 40 <1> inc eax.....
12 00000009 EBF8 <1> jmp nextchar.....
13 <1>.....
14 <1> finished:
15 0000000B 29D8 <1> sub eax, ebx
16 0000000D 5B <1> pop ebx.....
17 0000000E C3 <1> ret.....
18 <1>.
19 <1>.
20 <1> ;----- sprint -----
21 <1> ; Функция печати сообщения
22 <1> ; входные данные: mov eax, <message>
```

Рис. 4.9: Файл листинга lab8-2

Внимательно ознакомиться с его форматом и содержимым. Подробно объяснить содержимое трёх строк файла листинга по выбору.

строка 144

- 144 - номер строки
- 000000BB - адрес
- 80EB30 - машинный код

- `sub bl, 48` - код программы

строка 145

- 145 - номер строки
- 000000BE - адрес
- 01D8 - машинный код
- `add eax, ebx` - код программы

строка 146

- 146 - номер строки
- 000000C0 - адрес
- BB0A000000 - машинный код
- `mov ebx, 10` - код программы

Откройте файл с программой `lab8-2.asm` и в любой инструкции с двумя операндами удалите один операнд. Выполните трансляцию с получением файла листинга (рис. 4.10, 4.11)

```
type nasm -h for help.  
[gpgerra@fedora lab08]$ nasm -f elf -l lab8-2/lst lab8-2.asm  
lab8-2.asm: error: unable to open listing file `lab8-2/lst'  
[gpgerra@fedora lab08]$ nasm -f elf -l lab8-2.lst lab8-2.asm  
[gpgerra@fedora lab08]$ mcedit lab8-2.lst
```

Рис. 4.10: ошибка трансляции lab8-2

Рис. 4.11: файл листинга с ошибкой lab8-2

5. Напишите программу нахождения наименьшей из 3 целочисленных переменных a, b и c. Значения переменных выбрать из табл. 8.5 в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу (рис. 4.12, 4.13)

для варианта 10 - 41,62,35

Рис. 4.12: Файл lab8-3.asm

Рис. 4.13: Программа lab8-3.asm

6. Напишите программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции $f(x)$ и выводит результат вычислений. Вид функции $f(x)$ выбрать из таблицы 8.6 вариантов заданий в соответствии с вариантом, полученным при выполнении лабораторной работы № 7. Создайте исполняемый файл и проверьте его работу для значений X и a из 8.6. (рис. 4.14,4.15)

для варианта 10

$$\begin{cases} x - 7, & x \geq 7 \\ x^2, & x < 7 \end{cases}$$

Рис. 4.14: Файл lab8-4.asm

Рис. 4.15: Программа lab8-4.asm

5 Выводы

В заключение мы изучили команды условного и безусловного перехода и узнали о файле листинга.

Список литературы

1. Расширенный ассемблер: NASM
2. MASM, TASM, FASM, NASM под Windows и Linux