Отчёт по лабораторной рабоет №8

Компьтерные науки и технология программирования

Сячинова Ксения Ивановна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучить команды условного и безусловного перехода. Приобрести навыки написания программ с использованием переходов, а так же знакомство с назначением и структурой файла листинга.

2 Выполнение лабораторной работы

1. Создаём каталог для программ лабораторной работы №8, переходим в него и создаём файл ‘lab8-1.asm’.(рис. 2.1)

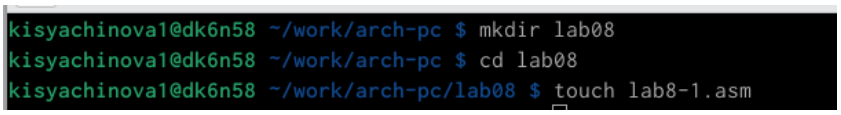

Рис. 2.1: Создание каталога и файла

2. Рассмотрим пример программы, открываем файл и вводим текст программы.(рис. 2.2), (рис. 2.3)

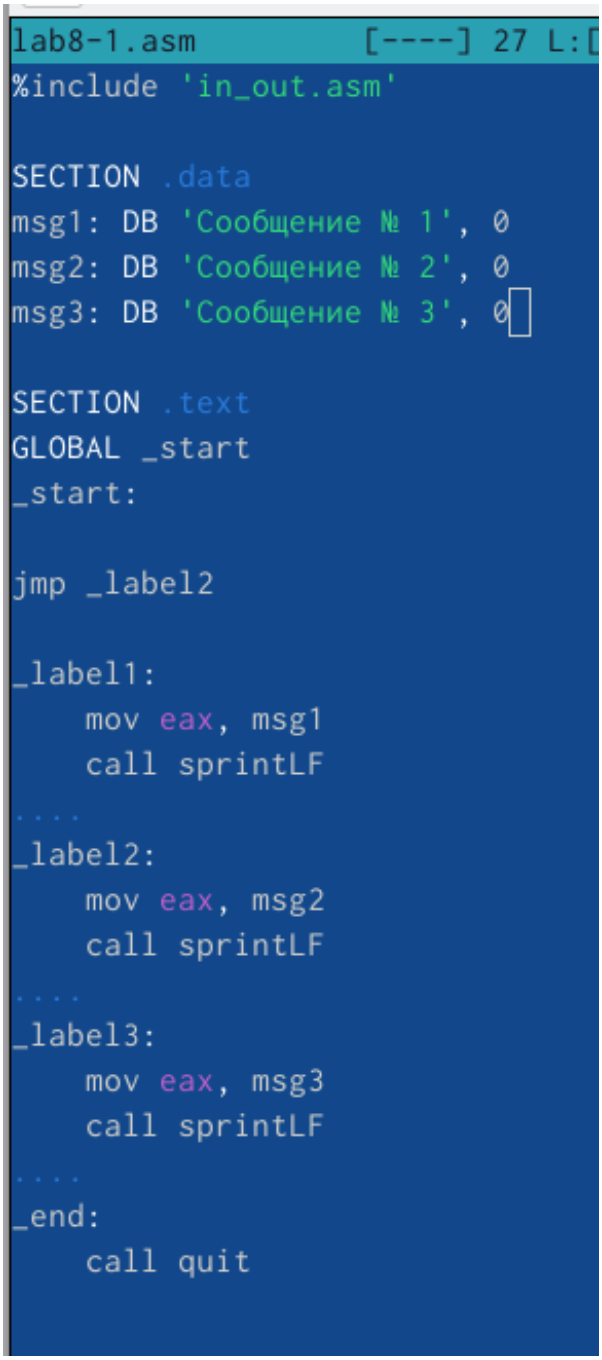

Рис. 2.2: Текст программы

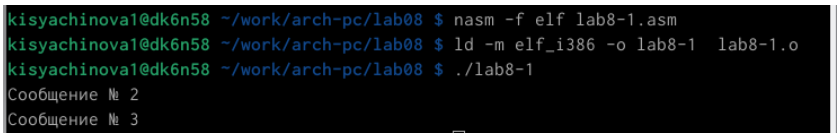

Рис. 2.3: Компиляция и выполнение файла

Команда ‘jmp’ позволяет начать использование инструкции с отмеченной метки, в нашем случае с ’_label2’.

Изменим программу так, чтобы она выводила сначала ‘Сообщение №2’, потом ‘Сообщение №1’ и завершала работу.(рис. 2.4), (рис. 2.5)

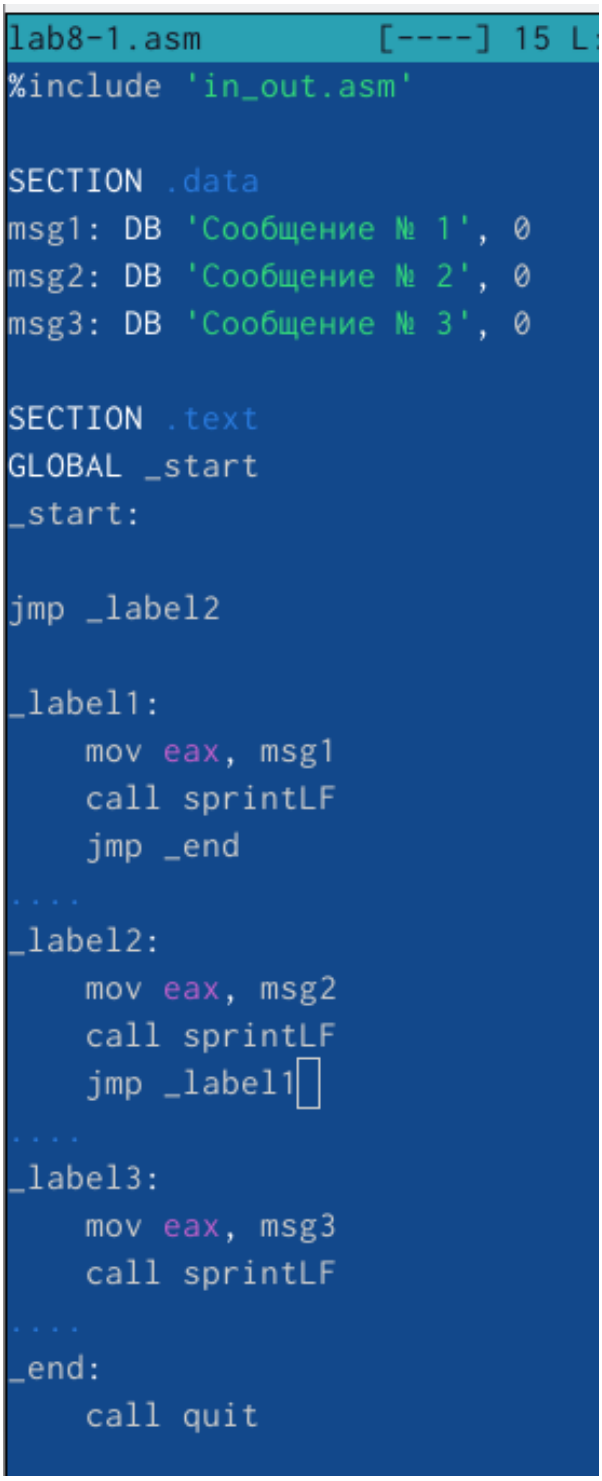

Рис. 2.4: Изменение программы

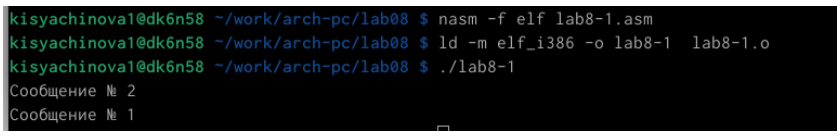

Рис. 2.5: Результат

Затем, изменим программу так, чтобы сообщения выводились в обратном порядке. Программы рабоатет корректно. (рис. 2.6), (рис. 2.7)

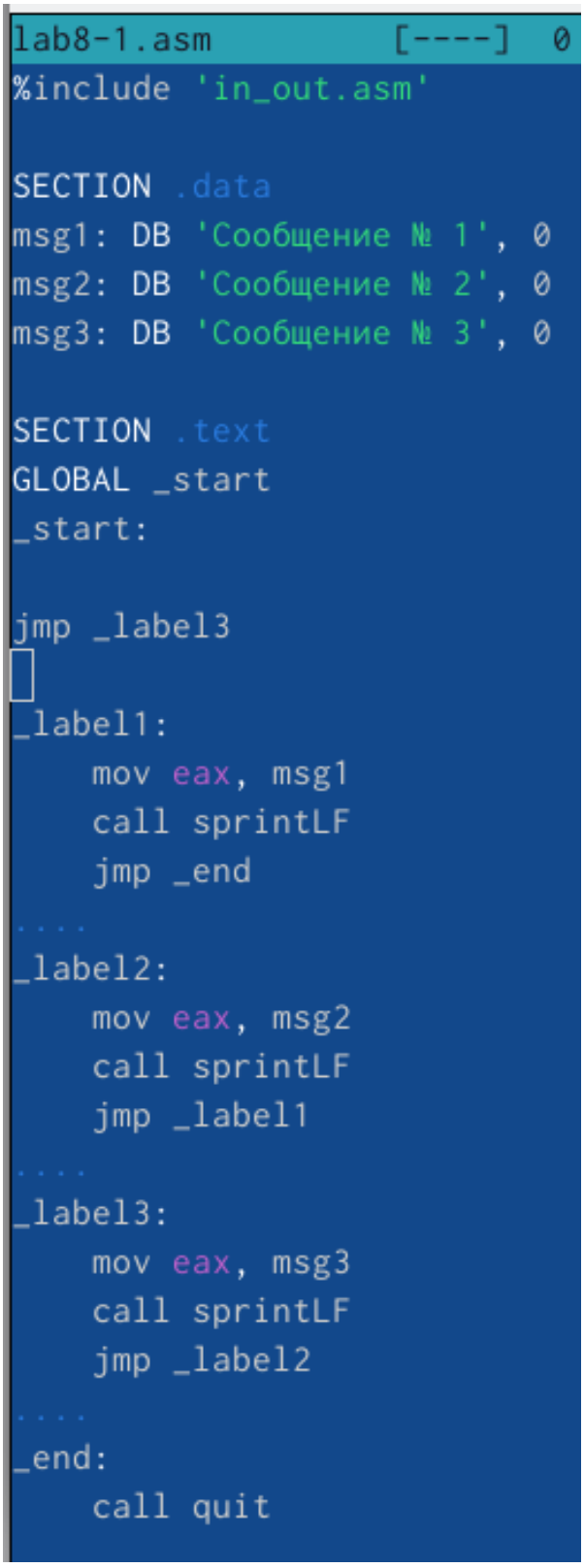

Рис. 2.6: Изменение программы

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1  lab8-1.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ ./lab8-1
Сообщение № 3
Сообщение № 2
Сообщение № 1
```

Рис. 2.7: Результат

3. Создадим файл ‘lab8-2.asm’ и вводим текст программы.(рис. 2.8), (рис. 2.9), (рис. 2.10),(рис. 2.11)

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ touch lab8-2.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ mcedit lab8-2.asm
```

Рис. 2.8: Создание файла

```
lab8-2.asm          [----]  8 L:[  1+ 0   1
%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
```

Рис. 2.9: Текст программы

```
mov ecx,[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
mov ecx,[B]
mov [max],ecx

fin:
mov eax, msg2
call sprint
mov eax, [max]
call iprintLF
call quit
```

Рис. 2.10: Текст программы

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ nasm -f elf lab8-2.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-2  lab8-2.o
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ ./lab8-2
Введите B: 10
Наибольшее число: 50
```

Рис. 2.11: Результат программы

4. Для получения файл листинга указывает ключ ‘-l’. Создадим файл листинга для программы из файла ‘lab8-2.asm’. и откроем его с помощью текстового редактора.(рис. 2.12)

```
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ nasm -f elf -l lab8-2.lst lab8-2.asm
kisyachinova1@dk6n58 ~/work/arch-pc/lab08 $ mcedit lab8-2.lst
```

Рис. 2.12: Создание файла листинга

Объясним содержимое трёх строк файла.

1) 24 - номер строки файла листинга, 00000106 - смещение машинного кода от начала текущего сегмента, E891FFFFFF - машинный код, в который ассеблируется данная инструкция в вие шестнадцатиричной последовательности, call atoi - исходная строка программы.(рис. 2.13)

Рис. 2.13: Строка 1

2) 47 - номер строка файла листинга, 00000159 - смещение машинного кода от начала текущего сегмента, ‘B8[13000000]’ - машинный код, в который ассеблируется данная инструкция в вие шестнадцатиричной последовательности, ‘mov eax,msg1’ - исходная строка программы. (рис. 2.14)

Рис. 2.14: Строка 2

3) 20 - номер строка файла листинга, 000000F7 - смещение машинного кода от начала текущего сегмента, ‘BA0A000000’ - машинный код, в который ассеблируется данная инструкция в вие шестнадцатиричной последовательности, ‘mov edx,10’ - исходная строка программы. (рис. 2.15)

Рис. 2.15: Строка 3

Удалим один операнд в программе, выполним трансляцию с получением файла. В файле листинга нам также выдаёт ошибку. (рис. 2.16)

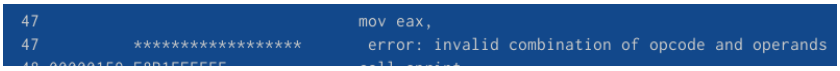

Рис. 2.16: Удаление операнда

3 Задание для самостоятельной работы

1. Напишем программу для нахождения наименьшей из 3 целочисленных переменных a, b и c. В соответсвии с лабораторной вариантом №7 мой вариант 11. Числа A=21, B=28, C=34 (рис. 3.1), (рис. 3.2),(рис. 3.3)„(рис. 3.4),

Рис. 3.1: Создание файла

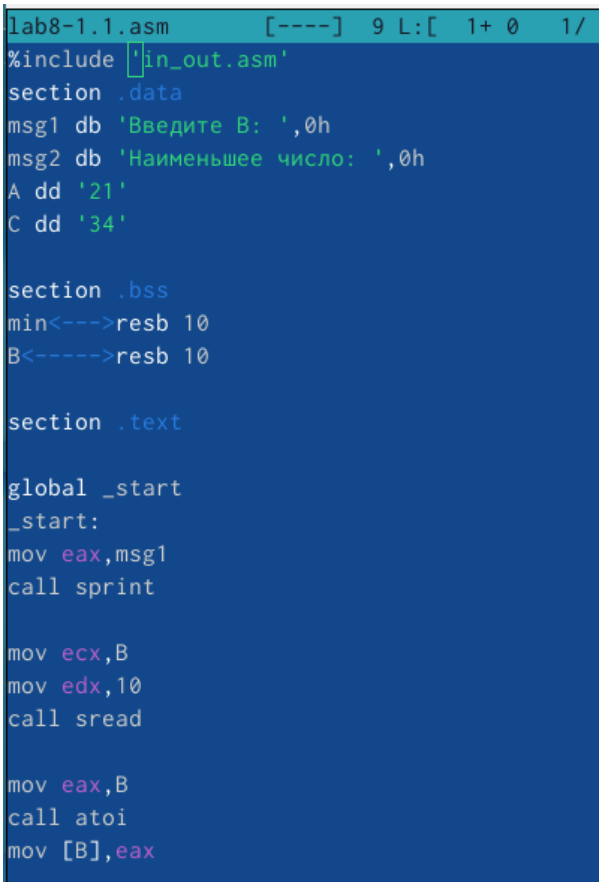

Рис. 3.2: Текст программы

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

cmp ecx,[C]
jb check_B
mov ecx,[C]
mov [min],ecx

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

mov ecx,[min]
cmp ecx,[B]
jb fin
mov ecx,[B]
mov [min],ecx

fin:
mov eax, msg2
call sprint
mov eax,[min]
call iprintLF
call quit
```

Рис. 3.3: Текст прогарммы

```
kisyachinova1@dk6n52 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.1.asm
kisyachinova1@dk6n52 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1.1  lab8-1.1.o
kisyachinova1@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-1.1
Введите B: 28
Наименьшее число: 21
```

Рис. 3.4: Выполнение программы

2. Напишем программу, которая для введённых значений ‘x’ и ‘a’ вычисляет значение заданной функции. Согласно лабораторной №7, мой вариант 11. Тогда имеем систему рис. 3.5. Проверим её с помощью значений (0;3) и (1;2)

11 $\begin{cases} 4a, & x = 0 \\ 4a + x, & x \neq 0 \end{cases}$ (0;3) (1;2)

Рис. 3.5: Система

```
kisyachinova1@dk6n52 ~/work/arch-pc/lab08 $ touch lab8-1.2.asm
kisyachinova1@dk6n52 ~/work/arch-pc/lab08 $ mcedit lab8-1.2.asm
```

Рис. 3.6: Создание файла

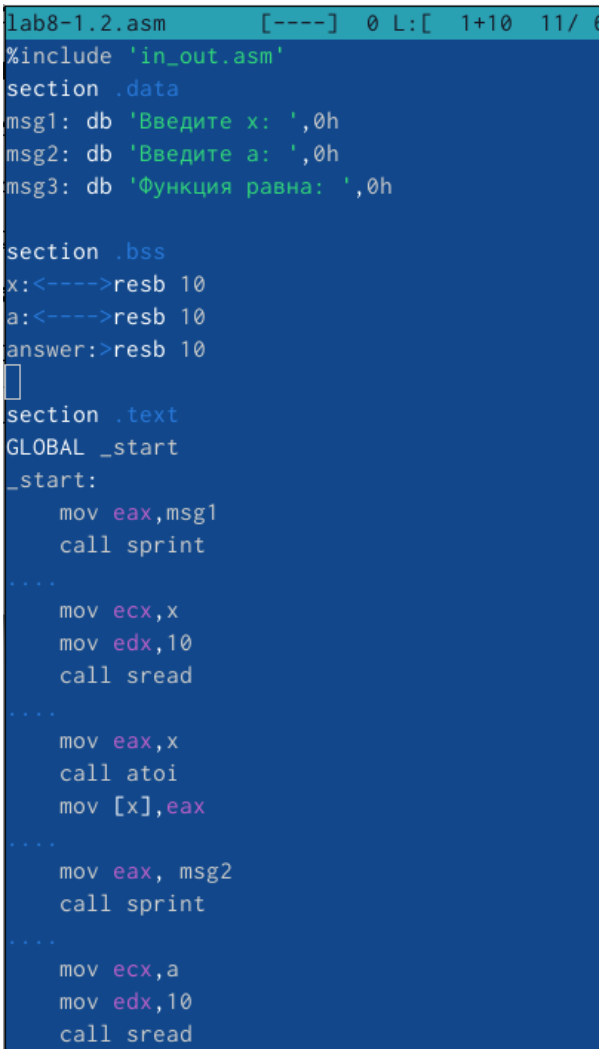

Рис. 3.7: Текст программы

```
    mov eax,a
    call atoi
    mov [a],eax
....
    mov eax, [x]
....
    cmp eax, 0
....
    je Func1
    jne Func2
....
Func1:
    mov eax, 4
    mov ebx, [a]
    mul ebx
    mov [answer], eax
    jmp Final
....
Func2:
    mov eax, 4
    mov ebx, [a]
    mul ebx
    add eax, [x]
    mov [answer], eax
    jmp Final
Final:
    mov eax,msg3
    call sprint
    mov eax, [answer]
    call iprintLF
    call quit
```

Рис. 3.8: Текст программы

Программа выдаёт верный результат. При (0;3) имеем значение выражение равное 12, в при (1;2) равное 9.(рис. 3.9),

```
kisyachinova1@dk6n52 ~/work/arch-pc/lab08 $ nasm -f elf lab8-1.2.asm
kisyachinova1@dk6n52 ~/work/arch-pc/lab08 $ ld -m elf_i386 -o lab8-1.2  lab8-1.2.o
kisyachinova1@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-1.2
Введите x: 0
Введите a: 3
Функция равна: 12
kisyachinova1@dk6n52 ~/work/arch-pc/lab08 $ ./lab8-1.2
Введите x: 1
Введите a: 2
Функция равна: 9
```

Рис. 3.9: Результат программы

#Вывод

В ходе выполнения данной лабораторной работы я изучила команды условного и безусловного перехода. Преобрела навыки по написанию программ с использованием переходов. Так же познакомилась с назначением и структурой файла листинга