**Отчет по лабораторной работе №7**

*дисциплина: Архитектура компьютера*

Сергеев Даниил Олегович

Содержание

# 1 Цель лабораторной работы

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

# 2 Ход выполнения лабораторной работы

1. Создаем каталог для лабораторной работы №7, переходим в него и создаем lab7-1.asm. В файл вводим текст программы из листинга 7.1.

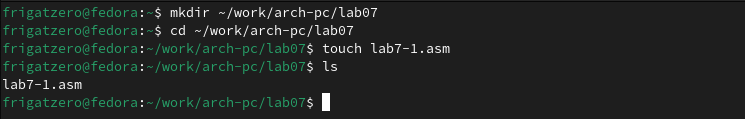


Рис. 1: Создание lab7-1.asm

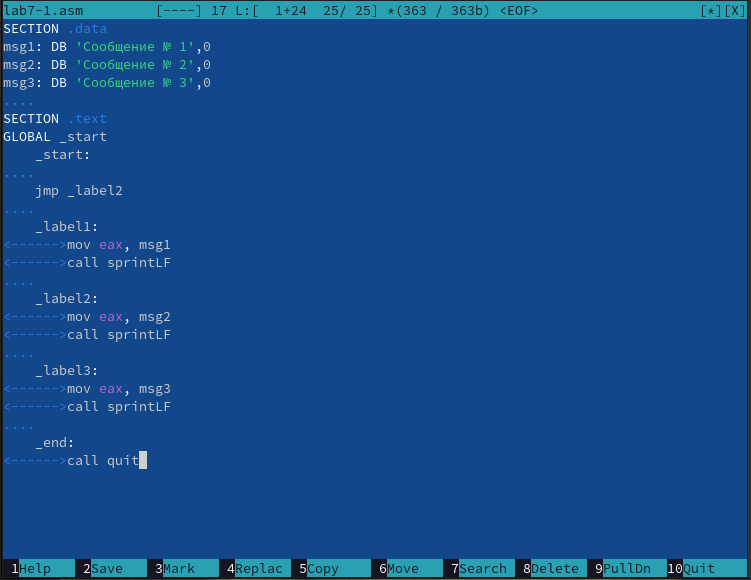


Рис. 2: Код файла lab7-1.asm

1. Создаем исполняемый файл и запускаем его.

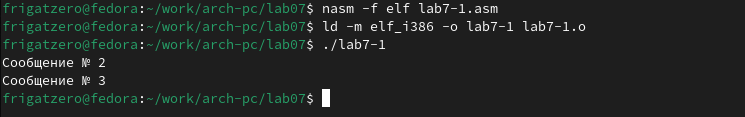


Рис. 3: Результат работы lab7-1

1. Изменим программу так, чтобы она выводила сначала “Сообщение № 2”, потом “Сообщение № 1” и завершал работу, в соответсвии с листингом 7.2. Создадим исполняемый файл и проверим его работу.

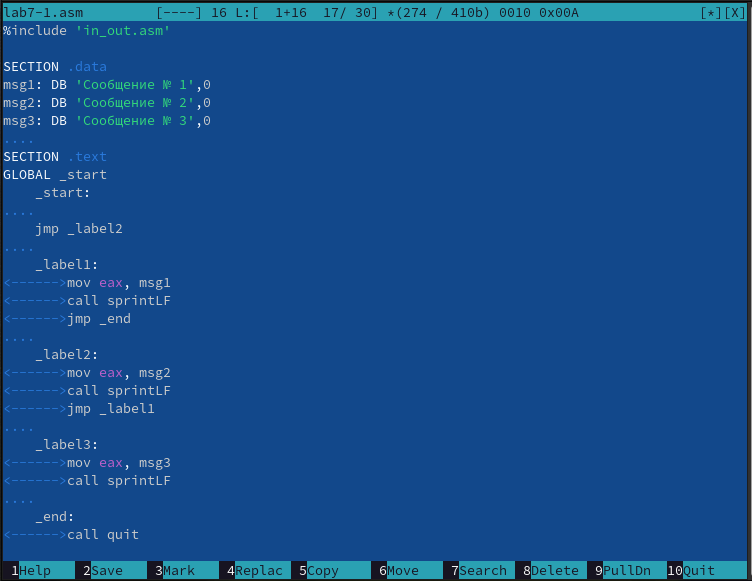


Рис. 4: Код файла из листинга 7.2

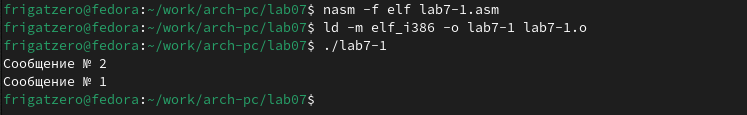


Рис. 5: Результат работы измененного файла lab7-1

1. Изменим программу, чтобы она выводила сообщения в порядке убывания с 3 по 1.

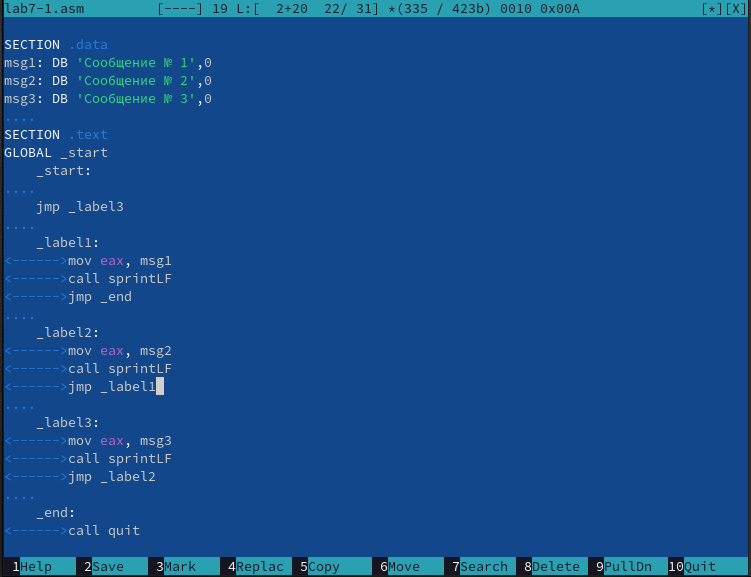


Рис. 6: Измененный код файла lab7-1.asm

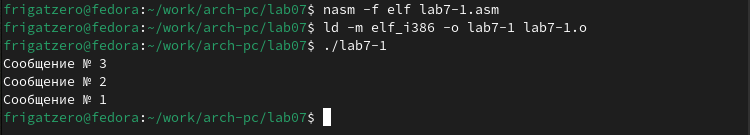


Рис. 7: Результат работы программы

1. Создадим файл lab7-2.asm в том же каталоге, введём код из листинга 7.3. Создадим исполняемый файл и проверим его работу для разных значений B.

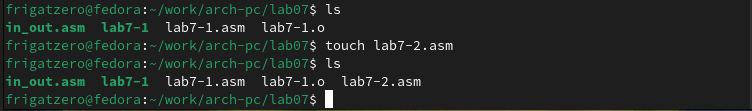


Рис. 8: Создание lab7-2.asm

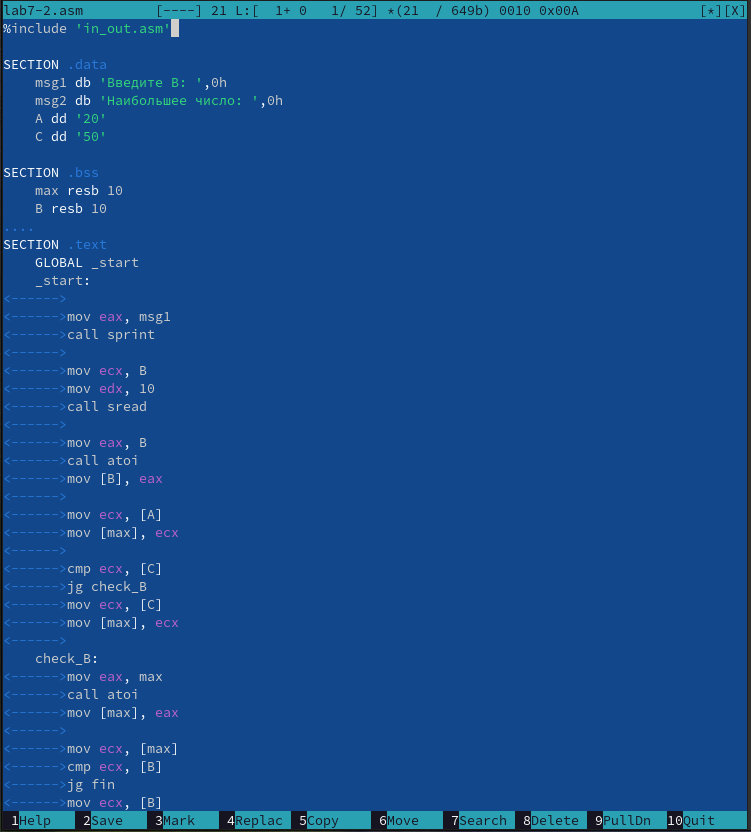


Рис. 9: Код файла из листинга 7.3

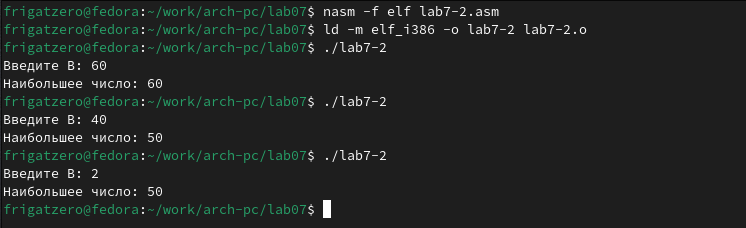


Рис. 10: Результат работы файла lab7-2

1. Создадим файл листинга для программы из файла lab7-2.asm и откроем его с помощью mcedit.

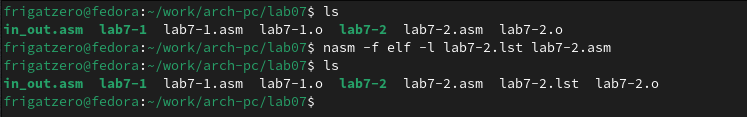


Рис. 11: Создание файла листинга для lab7-2.asm

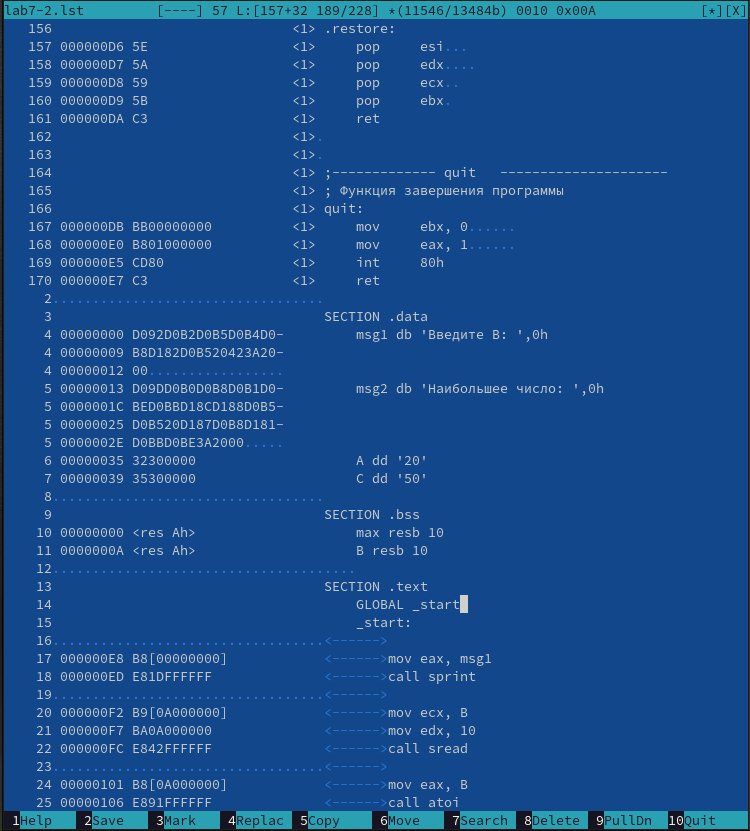


Рис. 12: Вид листинга из текстового редактора

1. Для объяснения выберем три строки из файла листинга.

Рис. 13: Код из листинга 6.1

Рис. 13: Код из листинга 6.1

* 1.Номер строки (20, 21, 22) - это номер строки файла листинга.
* 2.Адрес (000000F2/F7/FC) - это смещение машинного кода от начала текущего сегмента.
* 3.Машинный код (B9[0A000000], BA0A000000, E842FFFFFF) - это исходная строка представленная в виде 16-тиричной последовательности.
* 4.Исходный текст программы (mov ecx, B; mob edx, 10; call sread) - это строка исходной программы.

1. В инструкции mov edx, 10 удалим правый операнд и выполним трансляцию с получением файла листинга. После трансляции, выходит ошибка в файле lab7-2.asm. В выходных файлах мы получили только листинг, объектный файл не создался. В самом же листинге продублировалась строка 21, в которую записалась ошибка.

Рис. 14: Результат работы lab7-2 после удаления операнда

Рис. 14: Результат работы lab7-2 после удаления операнда

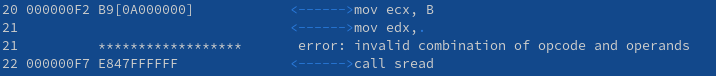


Рис. 15: Вывод листинга для файла lab7-2

# 3 Ход выполнения заданий для самостоятельной работы

1. Напишем программу для нахождения наименьшей из 3 целочисленных переменных a, b и c. Выберем значения переменных из №18, табл. 7.5. Создадим исполняемый файл и проверим его работу

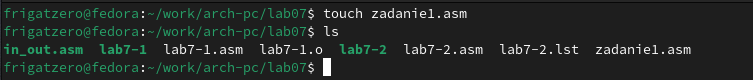


Рис. 16: Создаем файл zadanie1.asm

**Листинг 3.1** Программа для нахождения наименьшей переменной из a, b и c.

%include 'in\_out.asm'  
  
SECTION .data  
fnum db 'Первое число (a): ',0  
snum db 'Второе число (b): ',0  
tnum db 'Третье число (c): ',0  
minn db 'Наименьшее число: ',0  
  
a dd 83  
b dd 73  
c dd 30  
  
SECTION .bss  
  
min resb 10  
  
SECTION .text  
GLOBAL \_start  
 \_start:  
 ; Выводим значение a  
 mov eax, fnum  
 call sprint  
 mov eax, [a]  
 call iprintLF  
 ; Выводим значение b  
 mov eax, snum  
 call sprint  
 mov eax, [b]  
 call iprintLF  
 ; Выводим значение c  
 mov eax, tnum  
 call sprint  
 mov eax, [c]  
 call iprintLF  
 ; ----------------  
 ; Сравниваем a и b  
 ; ----------------  
 mov ecx, [a] ; ecx = a  
 mov [min], ecx ; min = ecx(a)  
 cmp ecx, [b] ; a < b  
 jl check  
 mov ecx, [b] ; ecx = b  
 mov [min], ecx ; min = ecx(b)  
 check:  
 ; ------------------  
 ; Сравниваем ecx и c  
 ; ------------------  
 cmp ecx, [c] ; ecx < c  
 jl fin  
 mov ecx, [c] ; ecx = c  
 mov [min], ecx ; min = ecx(c)  
 fin:  
 mov eax, minn  
 call sprint  
 mov eax, [min]  
 call iprintLF  
 call quit



Рис. 17: Результат работы программы zadanie1

1. Напишем программу, которая для введенных с клавиатуры значений x и a вычислит значение заданной функции f(x) и выведет результат вычислений. Вид функции возьмем под №18 из табл. 7.6. Проверим её работу с значениями x=1, a=2 и x=2, a=1.

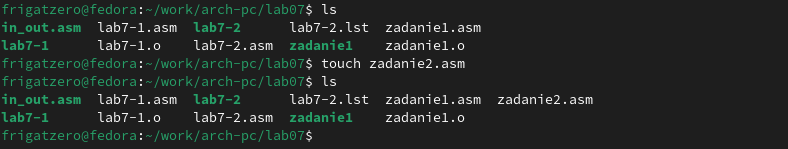


Рис. 18: Создаем файл zadanie2.asm

**Листинг 3.2** Программа для вычисления выражения f(x).

%include 'in\_out.asm'  
  
SECTION .data  
  
func db 'Задана функция №18',0  
fu\_1 db 'f(x) = a^2, a != 1',0  
fu\_2 db 'f(x) = 10 + x, a == 1',0  
inp\_x db 'Введите значение x: ',0  
inp\_a db 'Введите значение a: ',0  
result db 'Результат: ',0  
  
SECTION .bss  
  
x resb 4  
a resb 4  
f resb 4  
  
SECTION .text  
GLOBAL \_start  
 \_start:  
 ;-[Вывод сообщений func, fu\_1, fu\_2]  
 mov eax, func   
 call sprintLF  
 mov eax, fu\_1  
 call sprintLF  
 mov eax, fu\_2  
 call sprintLF  
 ;-[Ввод значения x]  
 mov eax, inp\_x  
 call sprint   
   
 mov ecx, x  
 mov edx, 10  
 call sread  
 ;-[Преобразование символа x в число]  
 mov eax, x  
 call atoi  
 mov [x], eax  
 ;-[Ввод значения a]  
 mov eax, inp\_a  
 call sprint  
   
 mov ecx, a  
 mov edx, 10  
 call sread  
 ;-[Преобразование символа a в число]  
 mov eax, a  
 call atoi  
 mov [a], eax  
 ; ==============  
 ; Вычисление f(x)  
 ; =============  
 mov ecx, [a] ; ecx = a  
 cmp ecx, 1 ; Сравниваем ecx и 1  
 je if\_equal ; если ecx = 1, то переходим к if\_equal  
 mov eax, [a] ; иначе записываем f = a^2  
 mov ebx, [a]  
 mul ebx ; eax = a^2  
   
 mov [f], eax ; f = eax  
 jmp fin ; переходим к fin  
 if\_equal:  
 mov ecx, [x] ; ecx = x  
 add ecx, 10 ; ecx = x + 10  
 mov [f], ecx ; f = ecx  
 ;-[Вывод результата]  
 fin:  
 mov eax, result  
 call sprint  
 mov eax, [f]  
 call iprintLF  
 call quit

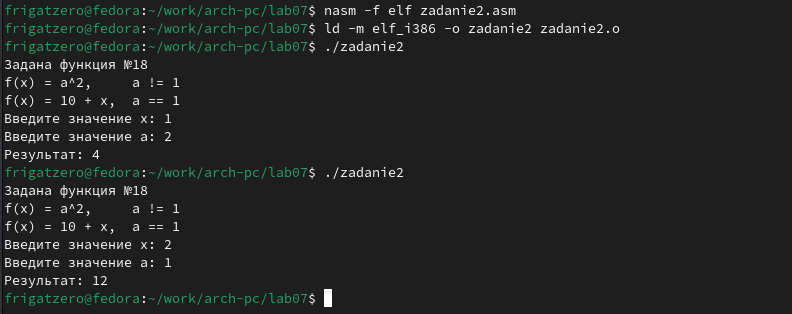


Рис. 19: Результат работы программы zadanie2

# 4 Вывод

После выполнения заданий лабораторной работы и заданий для самостоятельной работы я приобрел навыки написания программ с использованием условных и беусловных переходов, ознакомился с назначением и структурой файла листинга.