Отчет по лабораторной работе №8

*дисциплина: Архитектура компьютера*

Галацан Николай, НПИбд-01-22

Содержание

# 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. Ввожу команды для создания каталога лабораторной работы, перехожу в него, создаю файл lab8-1.asm

mkdir ~/work/arch-pc/lab08  
cd ~/work/arch-pc/lab08  
touch lab8-1.asm

1. Ввожу в файл lab8-1.asm текст программы из листинга 8.1, сохраняю файл. (рис. 1).

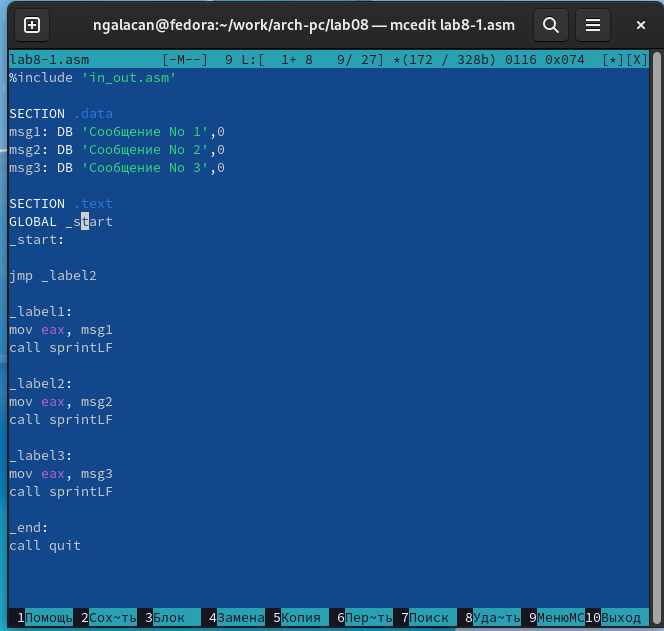


Рис. 1: Редактирование файла lab8-1.asm

Создаю исполняемый файл и запускаю его, предварительно скопировав файл in\_out.asm в соответствующий каталог (рис. 2).

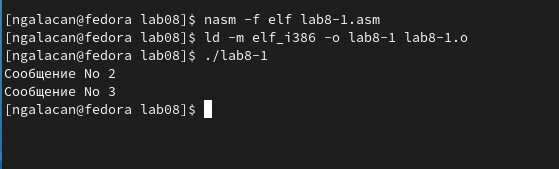


Рис. 2: Трансляция, компоновка, запуск исполняемого файла lab 8-1

Результат исполнения программы соответствует результату из инструкции.

Далее изменяю текст программы в соответствии с листингом 8.2 (рис. 3).

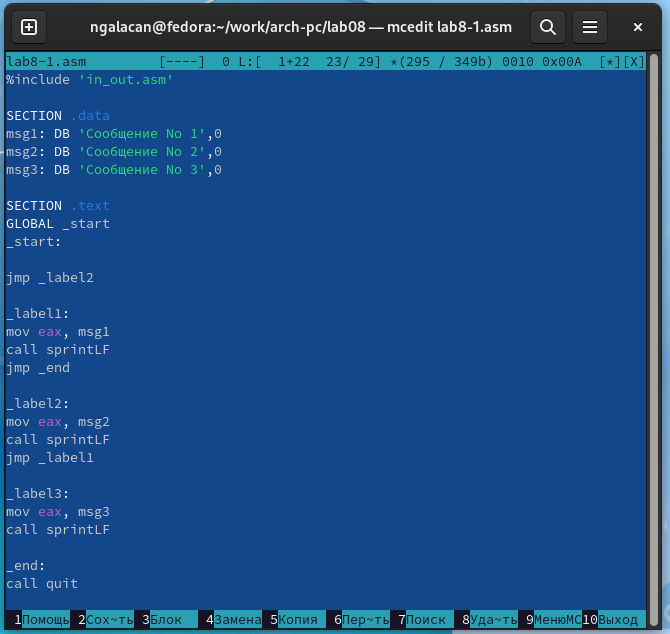


Рис. 3: Редактирование файла lab8-1.asm по листингу 8.2

Создаю исполняемый файл и запускаю его (рис. 4).

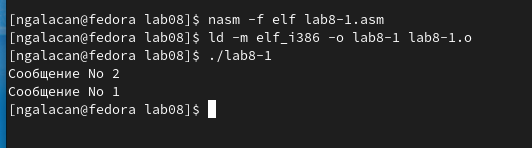


Рис. 4: Трансляция, компоновка, запуск измененного исполняемого файла lab 8-1

В результате исполнения программы выводится сначала “Сообщение №2”, а после “Сообщение №1”, так как был применен безусловный переход.

Изменяю текст программы, чтобы сообщения выводились в обратном порядке (от третьего к первому) рис. 5).

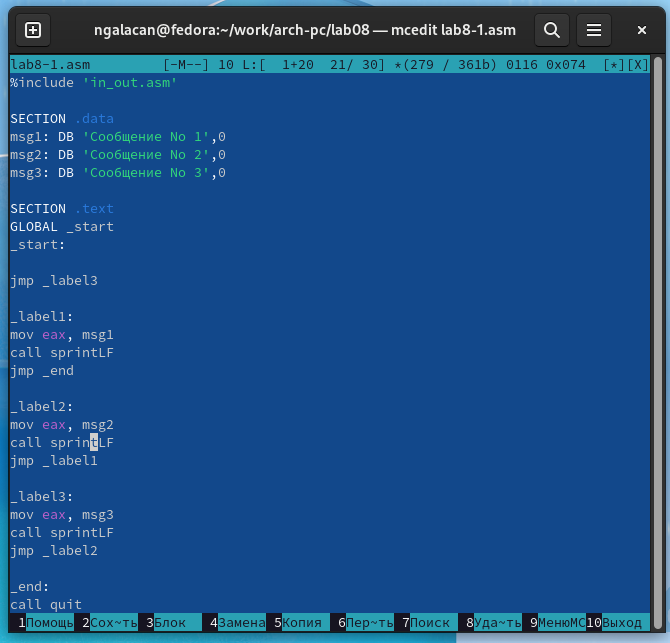


Рис. 5: Редактирование файла lab8-1.asm по инструкции

Создаю исполняемый файл и запускаю его (рис. 6).

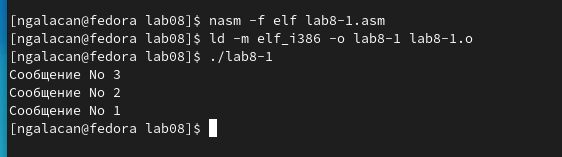


Рис. 6: Трансляция, компоновка, запуск исполняемого файла lab 8-1 после 3-го изменения

В результате работы программа выводит сообщения в нужном порядке.

1. Создаю новый файл: touch lab8-2.asm. Ввожу в него текст программы из листинга 8.3, внимательно изучив (рис. 7).

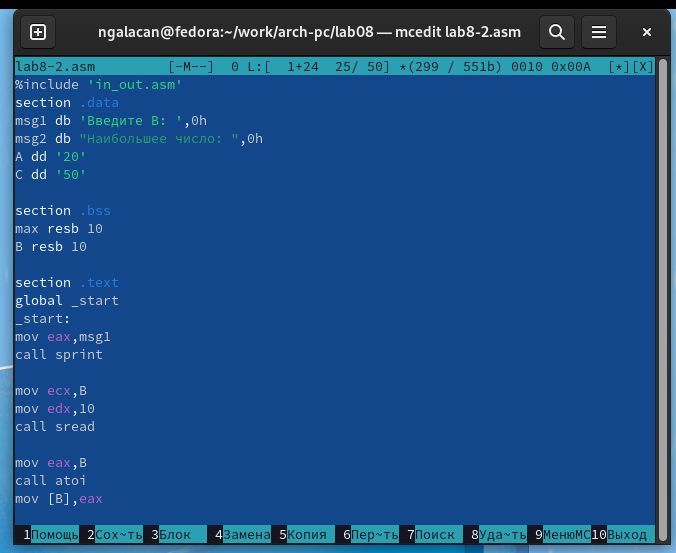


Рис. 7: Редактирование файла lab 8-2.asm

Создаю исполняемый файл и запускаю его. Проверяю работу программы для разных значений *В*. (рис. 8).

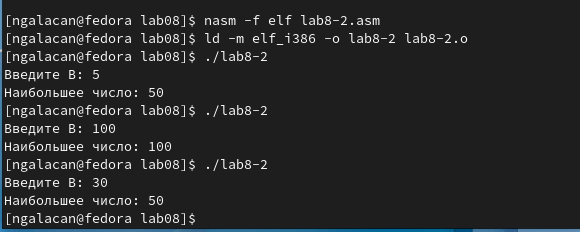


Рис. 8: Трансляция, компоновка, запуск исполняемого файла lab 8-2

1. Ввожу команду для получения файла листинга и открываю его в mcedit(рис. 9, рис. 10).

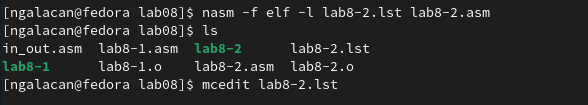


Рис. 9: Создание файла листинга lab8-2.lst

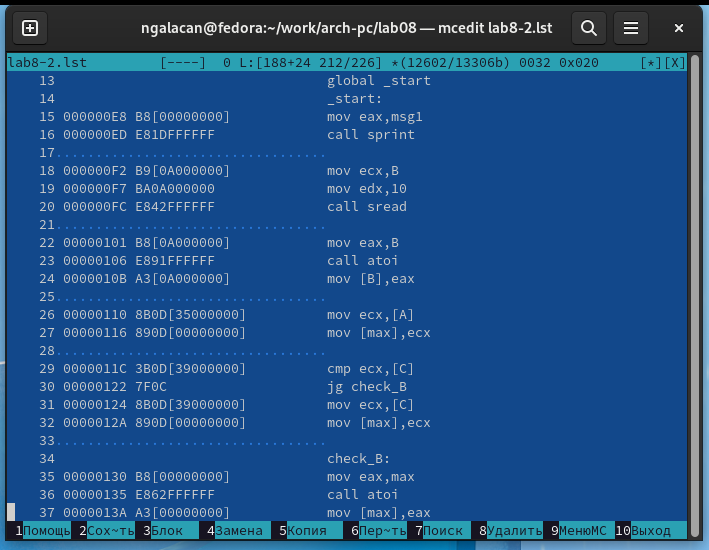


Рис. 10: Открытый файл листинга lab8-2.lst

Строки

mov ecx,B  
mov edx,10  
call sread

отвечают за считывание переменной *В*. Файл листинга, кроме исходного текста программы, содержит дополнительную информацию:

* в первом столбце указаны номера строк
* во втором столбце указан адрес - смещение машинного кода от начала текущего сегмента
* в третьем столбце - машинный код в шестнадцатеричном представлении, который ассемблируется из исходных строк

Например, инструкция mov ecx,B ассемблируется в машинный код B9[0A000000], а адресом является 000000F2, и все это находится на строке 18. Инструкция mov edx,10 ассемблируется в машинный код BA0A000000, а адресом является 000000F7, находится на строке 19. Инструкция call sread ассемблируется в машинный код E842FFFFFF, а адресом является 000000FC, находится на строке 20.

Открываю файл с программой lab8-2.asm и в инструкции mov eax, msg1 удаляю один операнд (рис. 11).

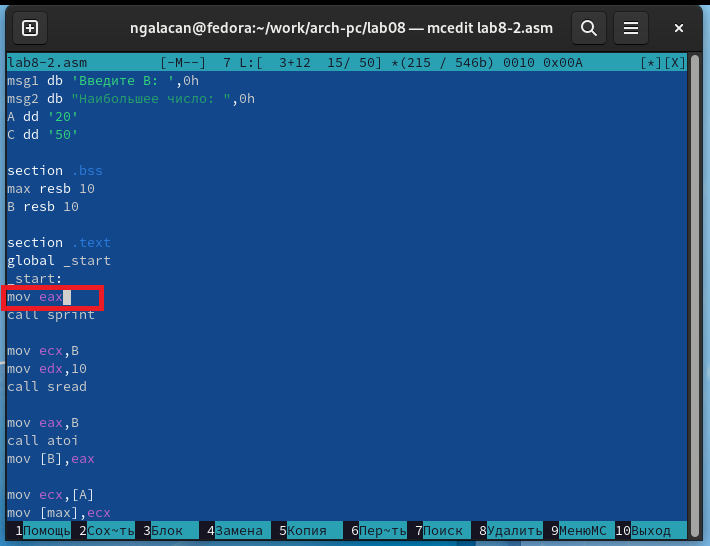


Рис. 11: Удаление одного операнда в lab8-2.asm

Выполняю трансляцию с получением файла листинга (рис. 12).

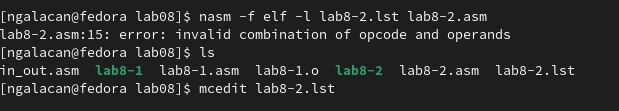


Рис. 12: Получение файла листинга измененной программы lab8-2.asm

Транслятор указывает на ошибку. Не создается объектный файл. Проверяю наличие файла листинга и открываю его (рис. 13).

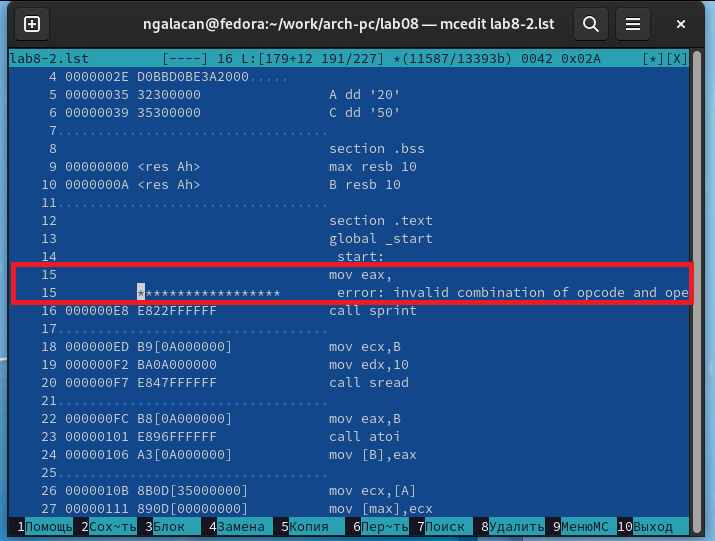


Рис. 13: Открытый файл листинга lab8-2.lst измененной программы

В листинге добавляется строка с указанием ошибки под строкой, в которой был удален операнд. Раздел с машинным кодом заполнен “\*“, адрес пуст.

# 3 Выполнение заданий для самостоятельной работы

В ЛР №7 был получен 4-ый вариант заданий.

1. Необходимо написать программу для нахождения наименьшей из 3 целочисленных переменных. Значения переменных выбраны из табл. 8.5 в соответствии с вариантом, и равны 8, 88, 68 соответственно.

Создаю файл lab8-sam-1.asm и набираю текст программы. Написанная программа имеет следующий вид:

%include 'in\_out.asm'  
section .data  
res db 'Наибольшее число: ',0h  
A dd '8'  
B dd '88'  
C dd '68'  
  
section .bss  
max resb 10  
  
section .text  
global \_start  
\_start:  
  
mov ecx,[A] ;запись А в max  
mov [max],ecx  
  
cmp ecx,[B] ;сравнение А и В  
jg check\_C ;если А>B  
mov ecx,[B] ;иначе  
mov [max],ecx  
  
check\_C:  
cmp ecx,[C] ;сравнение большего из А и В с С  
jg fin ;если max>С  
mov ecx,[C]  
mov [max],ecx  
  
fin:  
mov eax, res  
call sprint  
  
mov eax,max  
call atoi ;преобразование символа в число  
mov [max],eax  
  
mov eax,[max]  
call iprintLF  
call quit

Создаю исполняемый файл и запускаю, программа выводит верное наибольшее число (рис. 14).

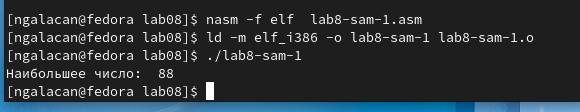


Рис. 14: Трансляция, компоновка, запуск исполняемого файла lab8-sam-1

1. Необходимо написать программу, которая вычисляет значение функции в соответствии с введенными *х* и *а* и выводит результат. Вид функции выбран из табл. 8.6 согласно варианту.

Создаю файл lab8-sam-2.asm и набираю текст программы. Для вычисления значения функции в зависимости от *а* использую условный переход je, переход совершается при выполнении равенства операндов инструкции cmp. Также использую команду безусловного перехода jmp. Написанная программа имеет следующий вид:

%include 'in\_out.asm'  
section .data  
var db 'Вариант 4.',0h  
msgx db 'Введите x: ',0h  
msga db 'Введите a: ',0h  
res db 'Результат: ',0h   
  
section .bss  
x resb 10  
a resb 10  
r resb 10  
  
section .text  
global \_start  
\_start:  
mov eax,var  
call sprintLF  
  
mov eax,msgx  
call sprint  
  
mov ecx,x  
mov edx,10  
call sread  
  
mov eax, x  
call atoi  
mov [x],eax  
  
mov eax,msga  
call sprint  
  
mov ecx,a  
mov edx,10  
call sread  
  
mov eax, a  
call atoi  
mov [a],eax  
  
mov edx,0  
cmp edx,[a]  
je label1  
mov eax,[x]  
mov ebx,2  
mul ebx  
add eax, [a]  
mov [r],eax  
jmp fin  
  
label1:  
mov eax,[x]  
mov ebx,2  
mul ebx ; 2\*x  
inc eax ; 2\*x+1  
mov [r],eax  
  
fin:  
mov eax, res  
call sprint  
mov eax,[r]  
call iprintLF  
call quit

Создаю исполняемый файл и запускаю. Проверяю результат для разных *х* и *а*, взятых из табл. 8.6 (рис. 15).

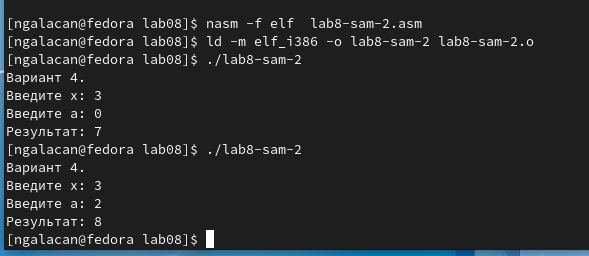


Рис. 15: Трансляция, компоновка, запуск исполняемого файла lab8-sam-2

Программа вычисляет верные значения функции.

# 4 Выводы

В ходе лабораторной работы изучены команды условного и безусловного переходов. Приобретены навыки написания программ с использованием переходов. Изучено назначение и структура файла листинга.