Лабораторная работа №8.

Команды безусловного и условного переходов в Nasm. Программирование ветвлений.

Хрусталев Влад Николаевич

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

Создадим каталог для программам лабораторной работы № 8, перейдем в него и создадим файл lab8-1.asm(рис. 1)

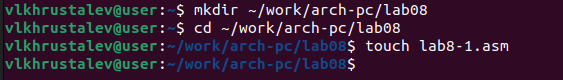


Рис. 1: Создание файла lab8-1.asm в соответствующем каталоге

Введем в файл lab8-1.asm текст программы из листинга 8.1.(рис. 2)

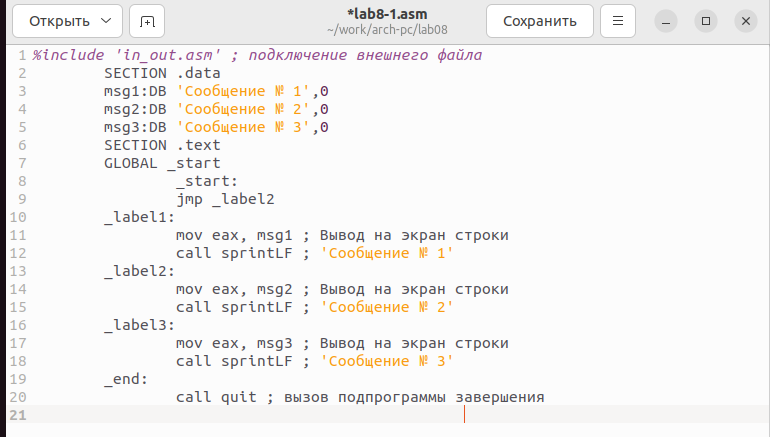


Рис. 2: Текст программы из листинга 8.1.

Создадим исполняемый файл и запустим его.(рис. 3)

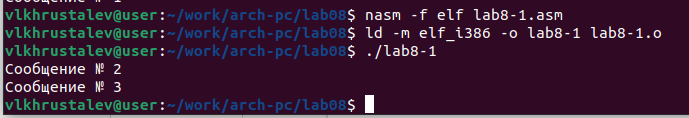


Рис. 3: Запуск исполняемого файла lab8-1.asm

Далее изменим текст программы в соответствии с листингом 8.2(рис. 4)

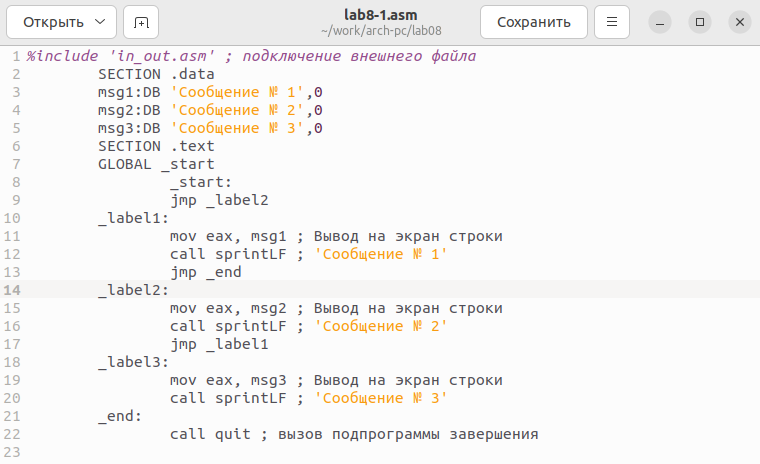


Рис. 4: Текст измененной программы

Создадим исполняемый файл исправленного текста программы lab8-1.asm и запустим его.(рис. 5)

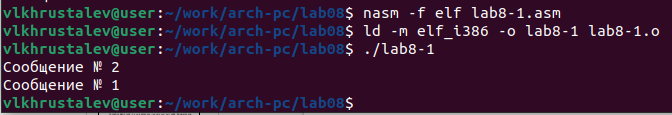


Рис. 5: Запуск исправленного исполняемого файла lab8-1.asm

Требуемый вывод:

user@dk4n31:~$ ./lab8-1

Сообщение № 3

Сообщение № 2

Сообщение № 1

user@dk4n31:~$(рис. 6)

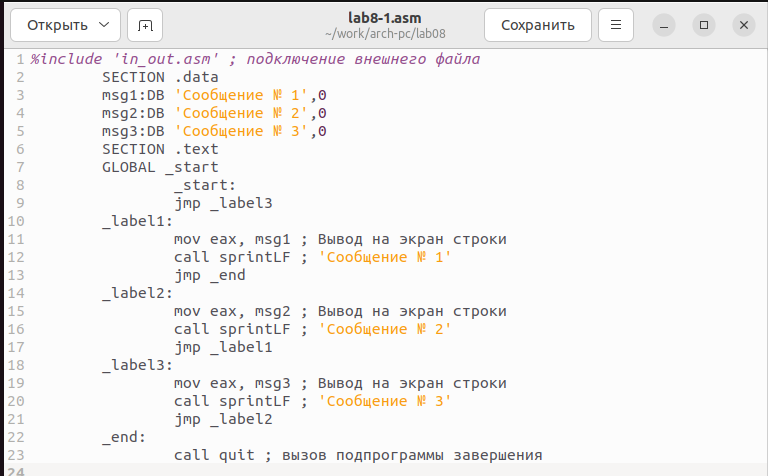


Рис. 6: Листинг программы с требуемым выводом

Создадим исполняемый файл и запустим его(рис. 7)

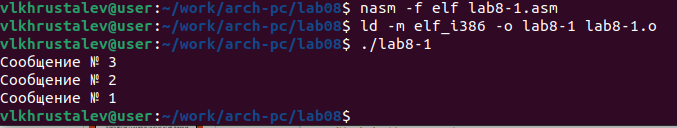


Рис. 7: Запуск измененного исполняемого файла lab8-1.asm

Создадим файл lab8-2.asm в каталоге ~/work/arch-pc/lab08. Внимательно изучим текст программы из листинга 8.3 и введем в lab8-2.asm.(рис. 8)

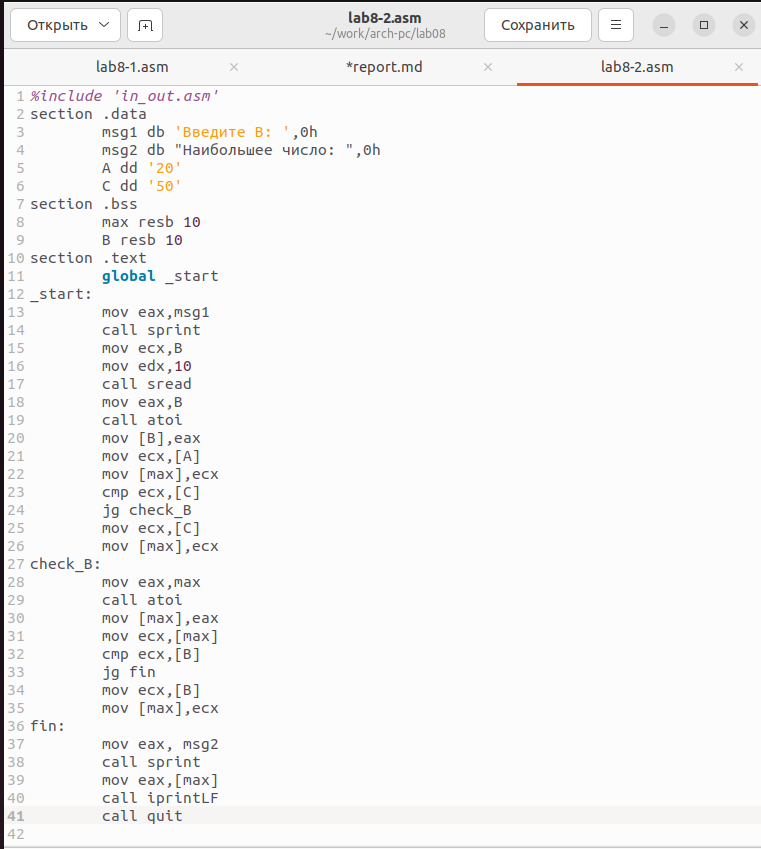


Рис. 8: Листинга 8.3

Создадим измененный исполняемый файл и запустим его(рис. 9)

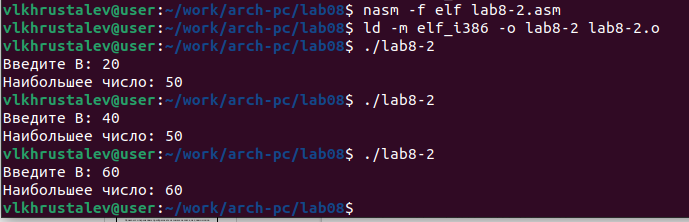


Рис. 9: Запуск исполняемого файла lab8-2.asm

Создадим файл листинга для программы из файла lab8-2.asm и откроем файл при помощи текстового редактора mcedit(рис. 10)

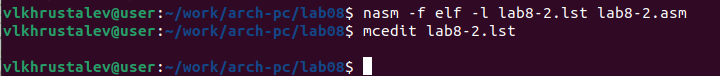


Рис. 10: Создание файла листинга

Разберем три строки листинга программы

15 000000F2 B9[0A000000] mov ecx,B

Значение строки:

15-номер строки в коде листинга от начала сегмента

000000F2 - адрес

B9[0A000000] - машинный код(B9[0A000000] - инструкция mov ecx,B; B9 - обозначет что действие производится с регистром ecx, а конкретно mov ecx(в данной программе) ; [0A000000] - ссылка на переменную B)

mov ecx,B - исходный текст программы

16 000000F7 BA0A000000 mov edx,10

Значение строки:

16-номер строки в коде листинга от начала сегмента

000000F7 - адрес

BA0A000000 - машинный код(BA0A000000 - инструкция mov edx,10 ; BA - обозначает что работает с с регисторм edx)

mov edx,10 - исходный текст программы

17 000000FC E842FFFFFF call sread

Значение строки:

17-номер строки в коде листинга от начала сегмента

000000FC - адрес

E842FFFFFF - машинный код(E842FFFFFF - инструкция call sread, E8 - значит что рабоает через переменную eax)

call sread - исходный текст программы(рис. 11)

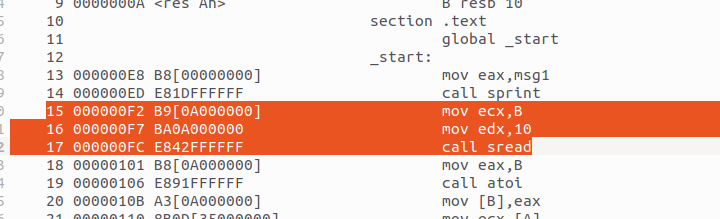


Рис. 11: Листинг программы

Откроем файл с программой lab8-2.asm и в любой инструкции с двумя операндами удалим один операнд. Выполните трансляцию с получением файла листинга:(рис. 12)

Рис. 12: Ошибка трансляции в терминале

Рис. 12: Ошибка трансляции в терминале

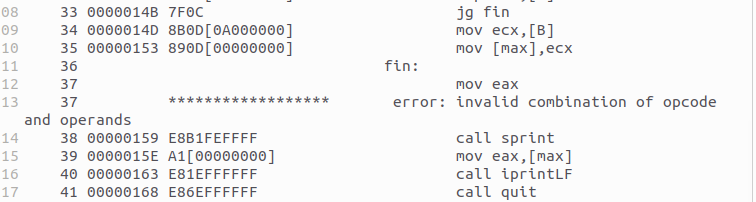


Рис. 13: Вывод ошибки в листинге

На выходе получаем листинг lab8-2.lst с ошибкой 46 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* error: invalid combination of opcode and operands

#Самостоятельная работа

Выполняем вариант номер 12

Напишем программу нахождения наименьшей из 3 целочисленных переменных A, B, C. Значения переменных выберем из табл. 8.5 в соответствиис вариантом, полученным при выполнении лабораторной работы № 7. Создадим исполняемый файл и проверьте его работу.(рис. 14)

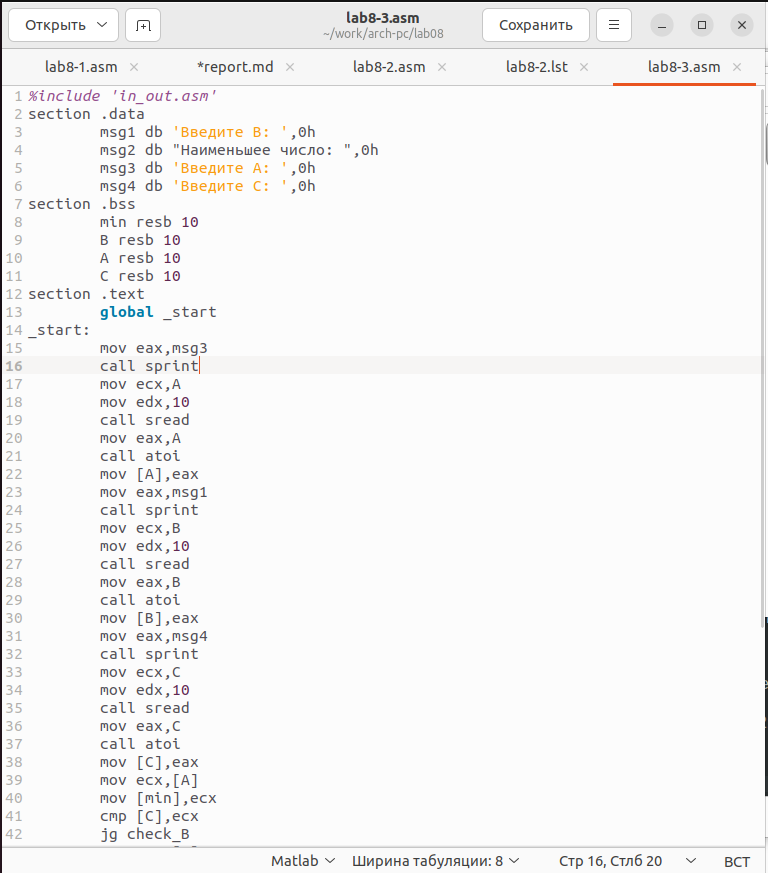


Рис. 14: Текст программы

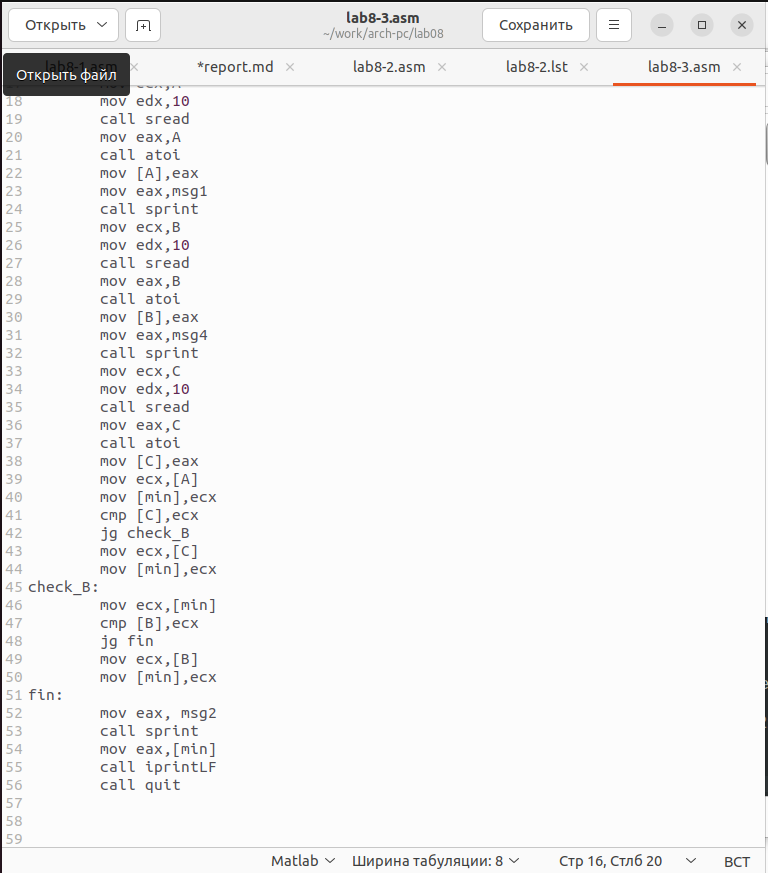


Рис. 15: Текст программы

Проверим работу исполняемого файла(рис. 16)

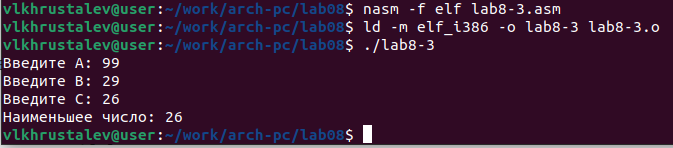


Рис. 16: Проверка работы исполняемого файла

Напишем 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. 17)

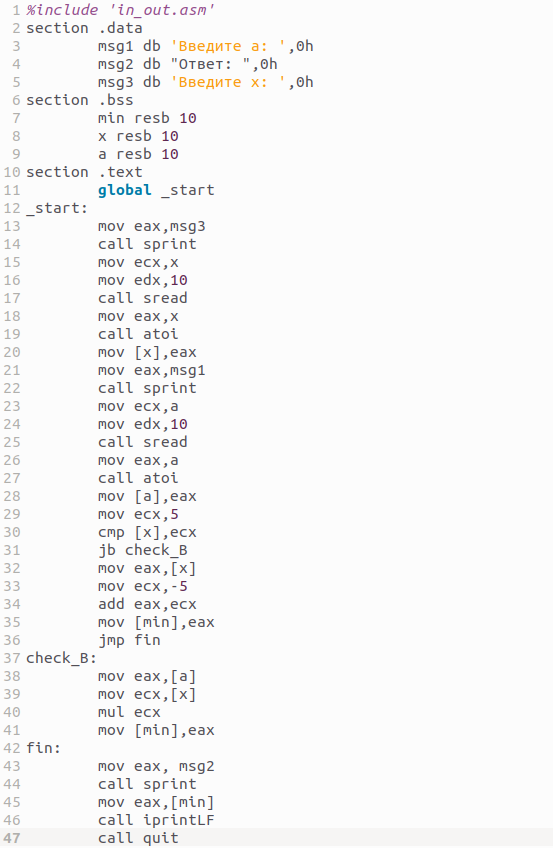


Рис. 17: Листинг программы lab8-4.asm

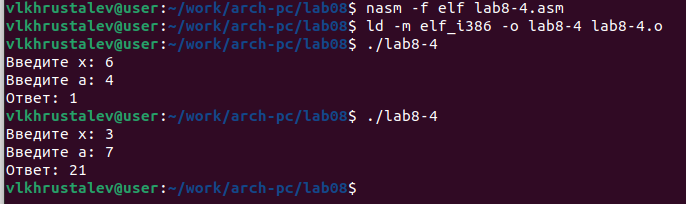


Рис. 18: Проверка работы программы

# 3 Выводы

В ходе лабораторной работы мы изучили команды условного и безусловного переходов, приобрели навыки написания программ с использованием переходов, познакомились с назначением и структурой файла листинга.

Ссылка на github: https://github.com/bezura/study\_2022-2023\_arch-pc