Отчёт по лабораторной работе №7

дисциплина: архитектура компьютеров

Ведьмина Александра Сергеевна

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 Задание

1. Ознакомиться с командами условного и безусловного перехода
2. Изучить структуру файла листинга
3. Рассмотреть программы, использующие разные переходы в nasm
4. Выполнить задания для самостоятельной работы

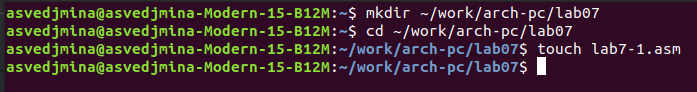
# 3 Теоретическое введение

Безусловный переход выполняется инструкцией jmp. Адрес перехода может быть либо меткой, либо адресом области памяти, в которую предварительно помещен указатель перехода. Для условных переходов задаются условия: например, мнемокод JE обозначает a=b, значение флага ZF=1 и осуществляет переход, если операнды равны.

Листинг - это один из выходных файлов, создаваемых транслятором. Он имеет текстовый вид и нужен при отладке программы. В его структуру входят номер строки (это номер строки файла листинга), адрес (смещение машинного кода от начала текущего сегмента), машинный код (ассемблированная исходная строка в виде шестнадцатеричной последовательности), исходный текст программы (строка исходной программы вместе с комментариями).

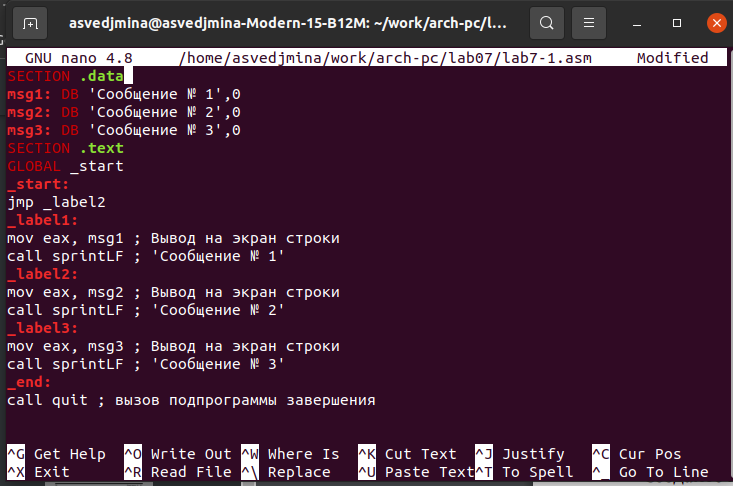
# 4 Выполнение лабораторной работы

Создаю каталог lab07 и файл lab7-1.asm в нём.



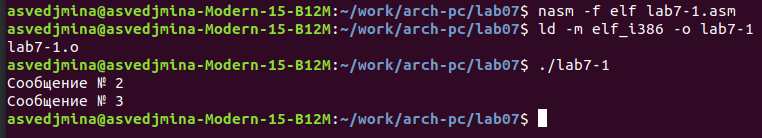
Создание каталога lab07 и файла lab7-1.asm

Ввожу в файл lab7-1.asm программу, использующую инструкцию jmp.



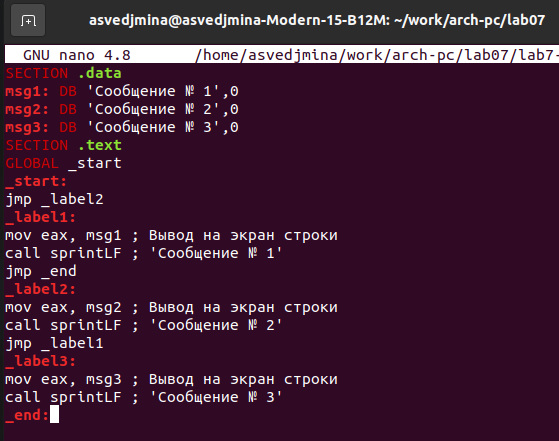
Ввод программы в файл lab7-1.asm

Создаю исполняемый файл и запускаю его.



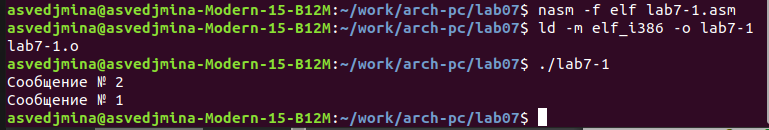
Запуск lab7-1

Изменяю программу так, чтобы выводилось сообщение 2, а потом сообщение 1.



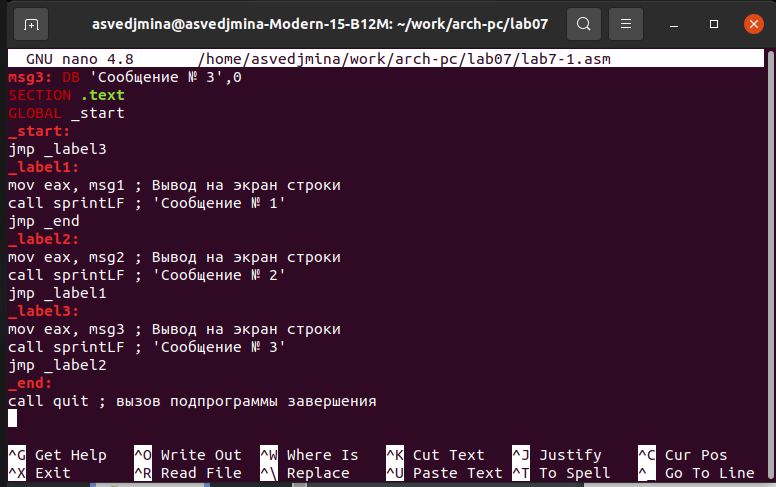
Изменение программы lab7-1.asm

Создаю исполняемый файл и запускаю его.



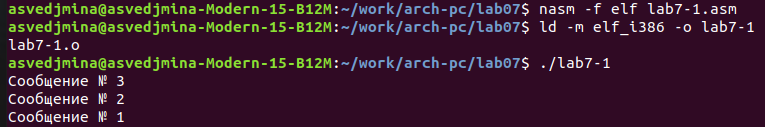
Запуск lab7-1

Снова изменяю текст программы таким образом, что выводилось сначала сообщение 3, затем сообщение 2 и сообщение 1.



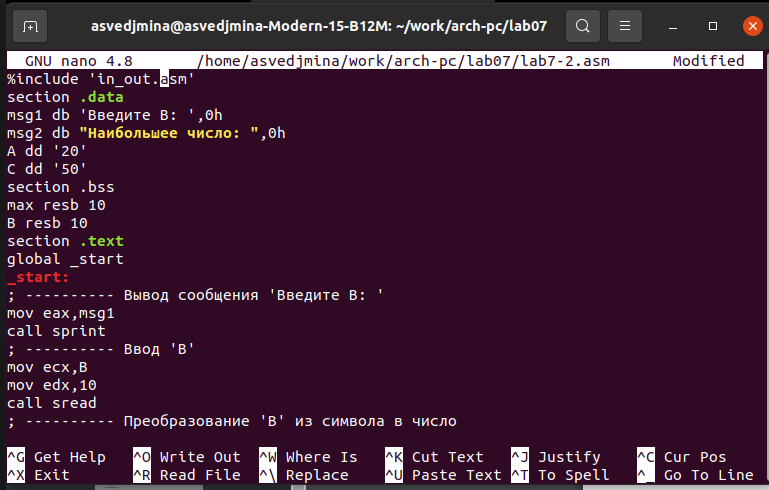
Изменение программы lab7-1.asm

Создаю исполняемый файл и проверяю его работу.



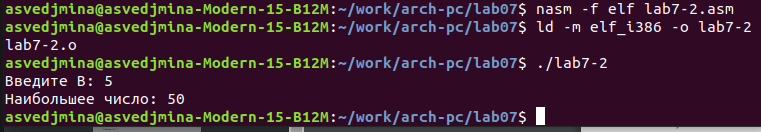
Запуск изменённого lab7-1

Создаю файл lab7-2.asm и ввожу в него текст программы, которая определяет и выводит на экран наибольшую из переменных A, B, C.

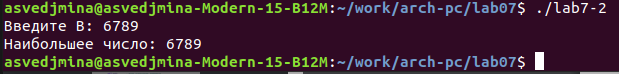


Ввод программы в lab7-2.asm

Создаю исполняемый файл и запускаю его, проверяю его работу при разных значениях В.



Запуск lab7-2 при В=5



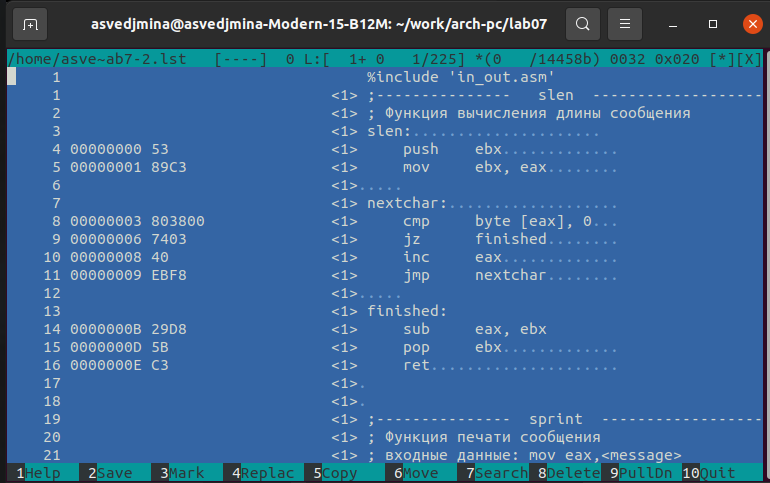
Запуск lab7-2 при В=6789

Создаю файл листинга для программы из файла lab7-2.asm.

Создание lab7-2.lst

Создание lab7-2.lst

Открываю файл lab7-2.lst с помощью mcedit и изучаю содержимое.



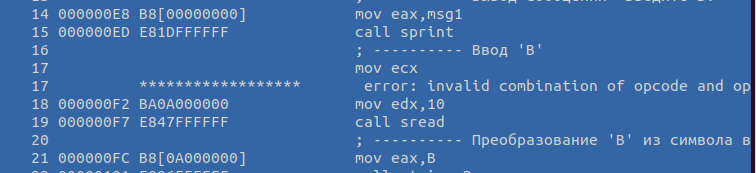
Просмотр lab7-2.lst

Cтрока lab7-2.lst 224: 1. 47 - номер строки в программе 2. 0000015Е - адрес 3. А1[0000000] - машинный код 4. mov eax, [max] - перемещение максимального значения среди переменных в eax

Cтрока lab7-2.lst 225: 1. 48 - номер строки в программе 2. 00000163 - адрес 3. Е81ЕFFFFFF - машинный код 4. call iprintLF - вывод переменной с наибольшим значением

Cтрока lab7-2.lst 226: 1. 49 - номер строки в программе 2. 00000168 - адрес 3. Е86ЕFFFFFF - машинный код 4. call quit - выход из программы

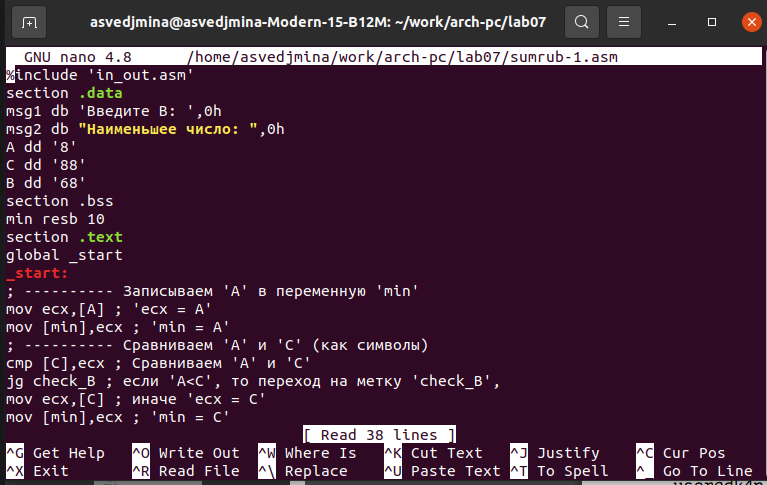
В файле с программой lab7-2.asm в инструкции с двумя операндами удаляю один операнд, после чего выполняю трансляцию с получением файла листинга. Открываю полученный файл и вижу, что звёздочками там отображается место, где в коде программы допущена ошибка и приводится её пояснение.



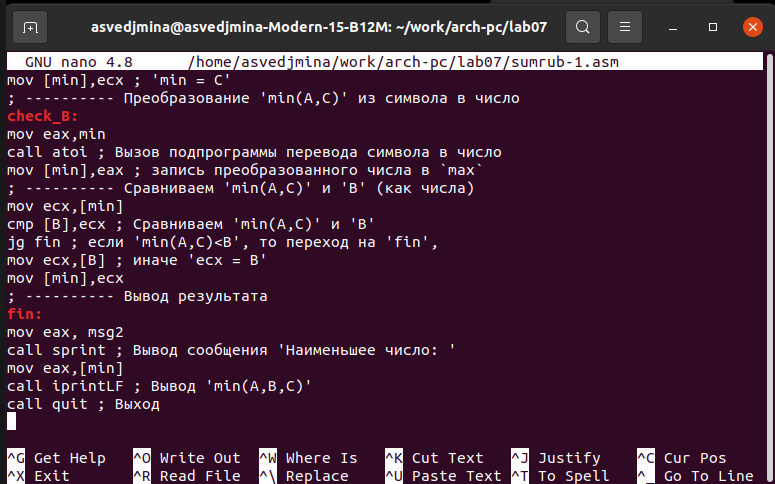
Просмотр lab7-2.lst с ошибкой

# 5 Выполнение заданий для самостоятельной работы

1. Создаю файл sunrub-1.asm и ввожу в него текст программы нахождения наименьшей из 3 целочисленных переменных. Мой вариант, полученный в предыдущей лабораторной - 4, поэтому я буду использовать следующие значения переменных: a = 8, b = 88, c = 68.

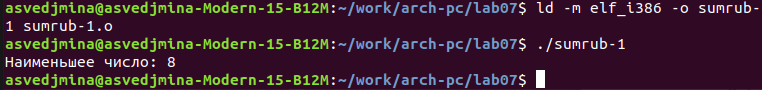


Ввод программы в sumrub-1.asm, 1 часть



Ввод программы в sumrub-1.asm, 2 часть

Создаю исполняемый файл и проверяю его работу.

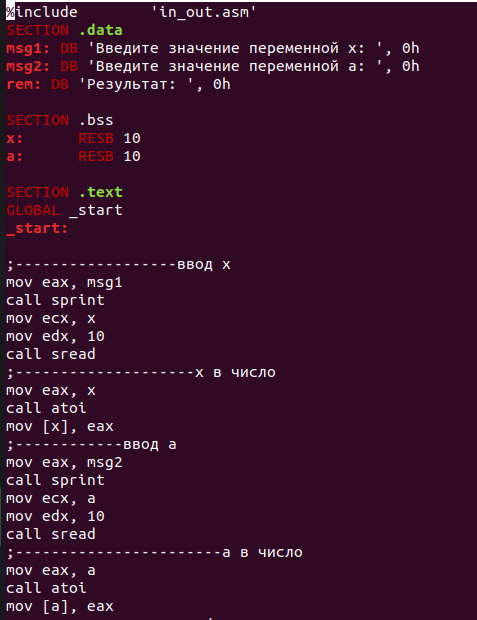


Запуск sumrub-1

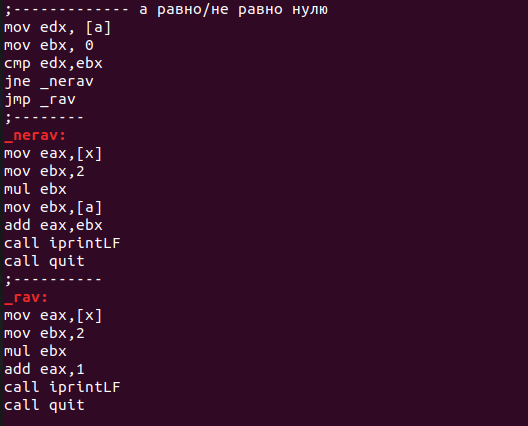
1. Создаю файл sumrub-2.asm и ввожу в него программу, которая для введенных с клавиатуры значений х и а вычисляет значение заданной функции. Я буду реализовывать функцию вида 2x+a при а, не равном нулю, и 2х+1 при а, равном нулю.

Создание файла sumrub-2.asm

Создание файла sumrub-2.asm

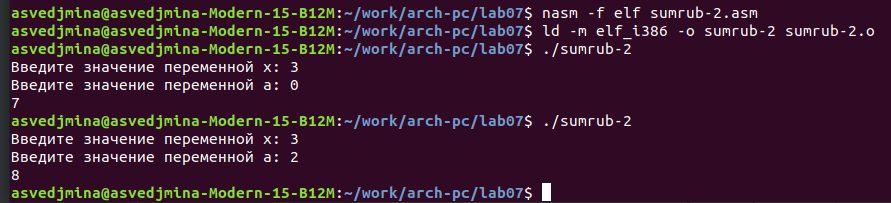


Ввод программы в sumrub-2.asm, 1 часть



Ввод программы в sumrub-2.asm, 2 часть

Создаю исполняемый файл и проверяю его работу для значений х и а, указанных в задании.



Запуск sumrub-2

# 6 Выводы

В ходе лабораторной работы я изучила команды условного и безусловного перехода, освоила написание программ с использованием переходов, а также познакомилась со структурой файла листинга.