Лабораторная работа №7

Архитектура компьютера

Кучмар София Игоревна

Содержание

# 1 Цель работы

Эта работа направлена на изучение команд условного и безусловного переходов, приобретение навыков написания программ с использованием инструкции jmp, знакомство с назначением и структурой файла листинга, команд условного перехода, инструкции cmp.

# 2 Задание

Данная работа посвящена практическому освоению ассемблера NASM. Будут изучены основы работы адресацией в NASM, освоены арифметические операции в NASM, целочисленное сложение add, целочисленное вычитание sub, команды инкремента и декремента, команда изменения знака операнда neg, основными директивами ассемблера, команды умножения mul и imul и будет написана программу для вычисления выражений с использованием инструкции jmp и программа, которая определяет и выводит на экран наибольшую/ наименьшую из 3 целочисленных переменных: A,B и C. Будет подключен внешний файл in\_out.asm с функциями ввода и вывода данных.

# 3 Выполнение лабораторной работы

Создадим каталог для программам лабораторной работы № 7, перейдём в него и создадим файл lab6-1.asm(рис. 1).

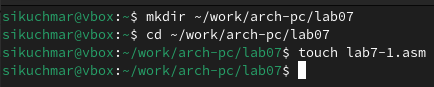


Рис. 1: Создание каталога и файла в нём

Рассмотрим пример программы с использованием инструкции jmp (рис. 2).

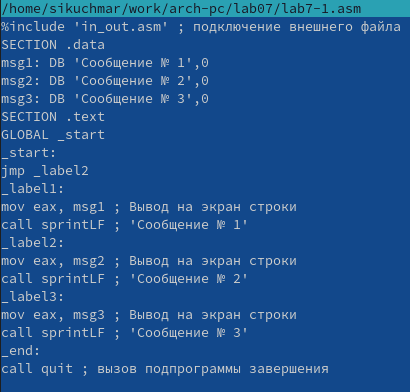


Рис. 2: Вводим программу

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

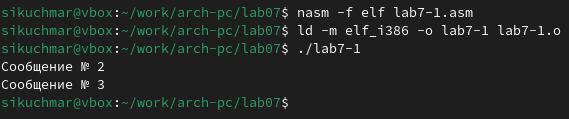


Рис. 3: Запуск файла

Изменим программу таким образом, чтобы она выводила сначала ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. Для этого в текст программы после вывода сообщения № 2 добавим инструкцию jmp с меткой \_label1 и после вывода сообщения № 1 добавим инструкцию jmp с меткой \_end(рис. 4).

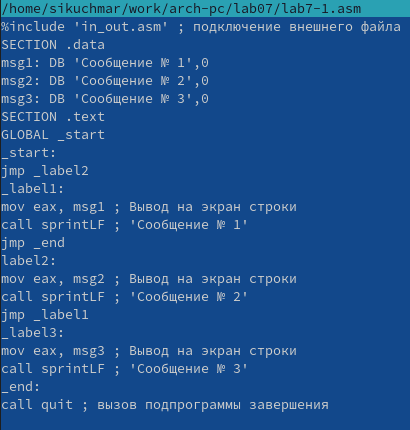


Рис. 4: Изменение текст программы

Создадим исполняемый файл и запустим его (рис. 5)

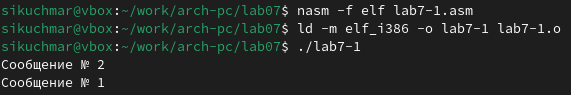


Рис. 5: Запуск файла

Изменим текст программы добавив или изменив инструкции jmp, чтобы вывод программы был следующим: сначала ‘Сообщение № 3’, ‘Сообщение № 2’, потом ‘Сообщение № 1’ и завершала работу. (рис. 6).

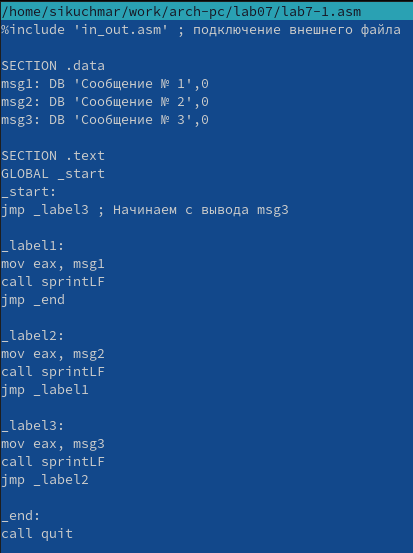


Рис. 6: Изменение текст программы

Создадим исполняемый файл и запустим его (рис. 7).

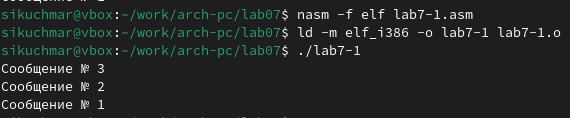


Рис. 7: Запуск файла

Создадим файл lab7-2.asm в каталоге ~/work/arch-pc/lab07 и создадим программу, которая определяет и выводит на экран наибольшую из 3 целочисленных переменных: A,B и C. Значения для A и C задаются в программе, значение B вводиться с клавиатуры. (рис. 8).

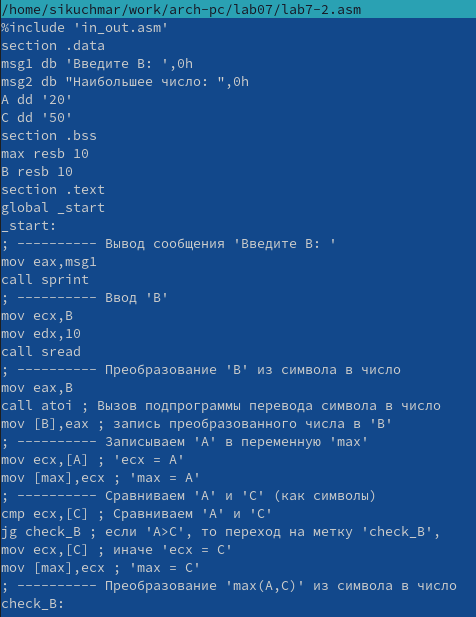


Рис. 8: Создание программы

Создадим исполняемый файл и проверим его работу для разных значений B (рис. 9).

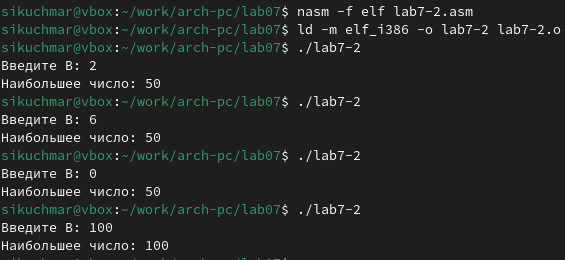


Рис. 9: Запуск файла

Обычно nasm создаёт в результате ассемблирования только объектный файл. Получить файл листинга можно, указав ключ -l и задав имя файла листинга в командной строке. Создадим файл листинга для программы из файла lab7-2.asm(рис. 10). Откроем файл листинга lab7-2.lst с помощью любого текстового редактора. Внимательно ознакомиться с его форматом и содержимым. Например: Строка 8: cmp byte [eax], 0 сравнивает байт по адресу eax с нулем, проверяя конец строки (нуль-терминатор).

Строка 14: sub eax, ebx вычисляет длину строки, вычитая начальный адрес (ebx) из конечного (eax).

Строка 27: call slen вызывает функцию slen для определения длины строки.

Давайте удалим один операнд из инструкции в строке 27. Например, удалим slen. В результате будут созданы два файла lab7-2.o : объектный файл, содержащий машинный код и lab7-2.lst: листинговый файл.В листинговом файле в строке 27 мы увидим сообщение об ошибке от ассемблера. Ассемблер не сможет корректно собрать программу, так как инструкция call требует операнда.

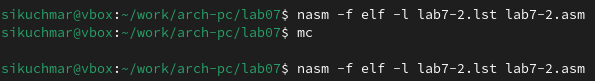


Рис. 10: Создание lst файлов

Напишем программу нахождения наименьшей из 3 целочисленных переменных a,b и с. (рис. 11).

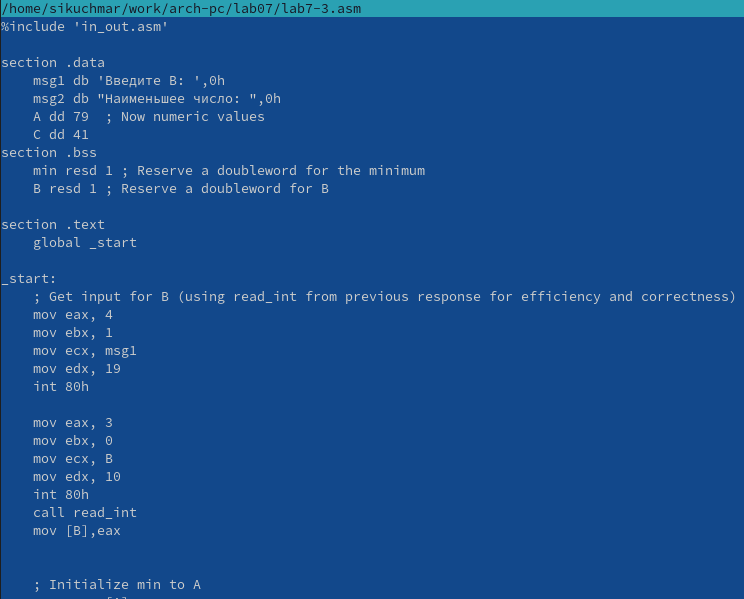


Рис. 11: Создание программы

Создадим исполняемый файл и запустим его для значений Варианта 6: a=79, b=83, c=41 (рис. 12).

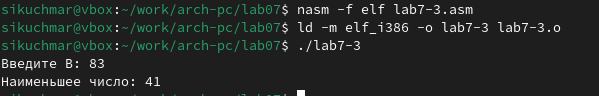


Рис. 12: Запуск файла

Напишем программу, которая для введенных с клавиатуры значений x и a вычисляет значение заданной функции x+a, при x не = а, 5x, при x=a и выводит результат вычислений. (рис. 13).

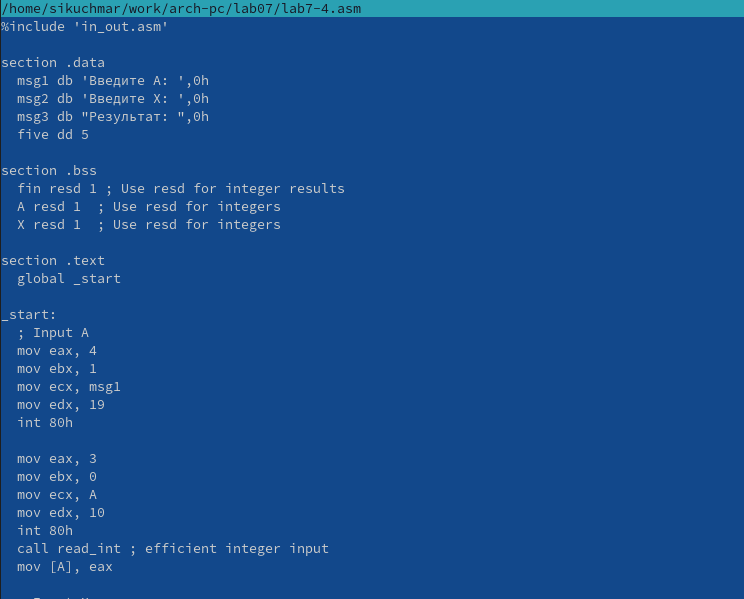


Рис. 13: Создание программы

Создадим исполняемый файл и запустим его для значений Варианта 6: x1=2, a1=1, x2=2, a2=1 (рис. 14).

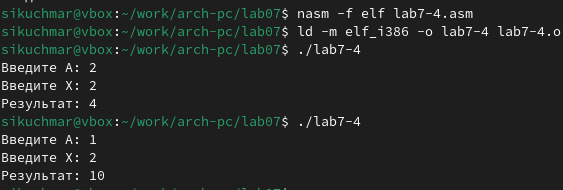


Рис. 14: Запуск файла

# 4 Выводы

В рамках данной работы были успешно освоены основы работы с ассемблером NASM. Были освоены разные операции с переменными, такие как сравнение и вычисление наибольшего и наименьшего в NASM и создание программ для вычисления выражений.