Отчёт по лабораторной работе 6

Архитектура компьютера

Саяпин Артём Максимович

Содержание

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

Я создал папку для хранения файлов шестой лабораторной работы, перешел в нее и сформировал файл с исходным кодом lab6-1.asm.

В ходе этой лабораторной мы изучим примеры программ, демонстрирующих вывод символов и цифровых данных на экран. Эти программы будут оперировать данными, помещенными в регистр eax.

В одной из программ мы помещаем символ ‘6’ в регистр eax с помощью инструкции (mov eax, ‘6’) и символ ‘4’ в регистр ebx (mov ebx, ‘4’). После этого выполняем сложение значений, находящихся в регистрах eax и ebx (add eax, ebx), и отображаем полученный итог.

Чтобы воспользоваться функцией sprintLF, которая ожидает адрес в регистре eax, мы применяем вспомогательную переменную. Сначала мы копируем содержимое регистра eax в переменную buf1 (mov [buf1], eax), затем загружаем адрес buf1 обратно в регистр eax (mov eax, buf1) и выполняем вызов функции sprintLF.

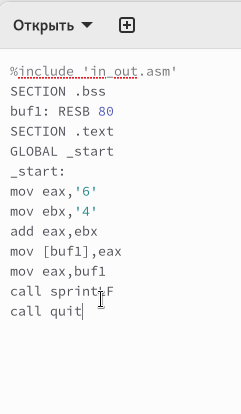


Figure 1: Код программы lab6-1.asm

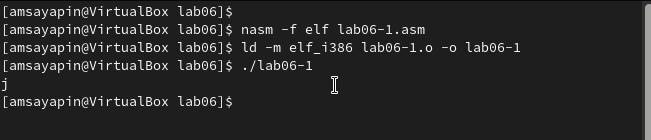


Figure 2: Компиляция и запуск программы lab6-1.asm

В этом примере мы ожидаем на выходе число 10 после сложения значений регистров eax и ebx. Тем не менее, на экране появится символ ‘j’, так как двоичный код символа ‘6’ равен 00110110 (что соответствует 54 в десятичной системе), а символа ‘4’ – 00110100 (или 52 в десятичной системе). Инструкция add eax, ebx приводит к записи в регистр eax суммы этих кодов – 01101010 (или 106 в десятичной системе), что соответствует коду символа ‘j’.

После этого я внес изменения в код программы, заменив символы на числовые значения в регистрах.

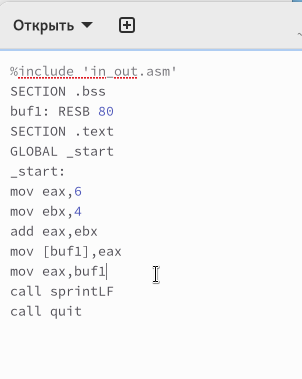


Figure 3: Код программы lab6-1.asm

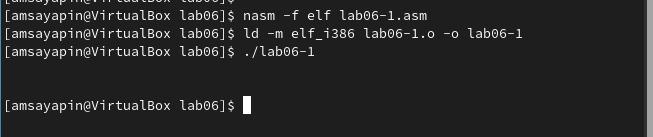


Figure 4: Компиляция и запуск программы lab6-1.asm

Так же, как и в прошлый раз, при запуске программы мы не увидим число 10. Вместо этого пройдет вывод символа, который соответствует коду 10. Этот символ представляет собой перевод строки (или возврат каретки). На экране консоли он невидим, но создает пустую строку.

В файле in\_out.asm внедрены вспомогательные процедуры для преобразования ASCII-символов в числовые значения и наоборот. Я внес изменения в код программы, применив эти процедуры.

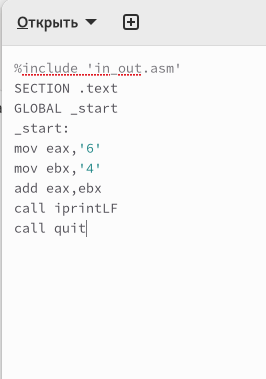


Figure 5: Код программы lab6-2.asm

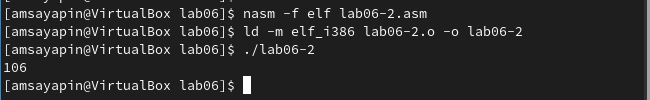


Figure 6: Компиляция и запуск программы lab6-2.asm

После запуска программы на экран будет выведено число 106. В этом случае, подобно первому примеру, команда add суммирует численные значения символов ‘6’ и ‘4’ (54+52=106). Но в отличие от прошлой версии программы, функция iprintLF позволяет отобразить именно число, а не символ с соответствующим числовым кодом.

Таким же образом, как и в предыдущем случае, мы провели замену символов на их числовые эквиваленты.

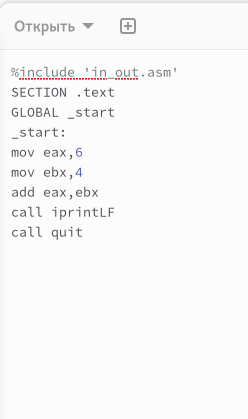


Figure 7: Код программы lab6-2.asm

Функция iprintLF предназначена для вывода числовых значений, и в данном контексте она работает с числами, а не с их символьными кодами. В результате мы видим на экране число 10.

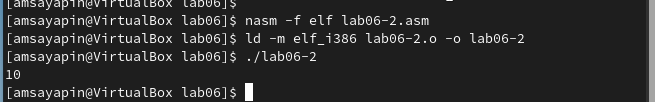


Figure 8: Компиляция и запуск программы lab6-2.asm

Функция iprintLF используется для отображения чисел, и в этой ситуации в качестве операндов выступают числа (не их символьные коды), что приводит к выводу числа 10.

Мы произвели замену функции iprintLF на iprint. Скомпилировали программу, создали исполняемый файл и запустили его. Разница в выводе заключается в отсутствии перевода строки.

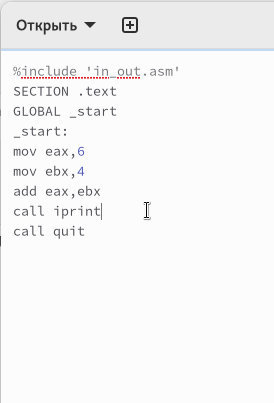


Figure 9: Код программы lab6-2.asm

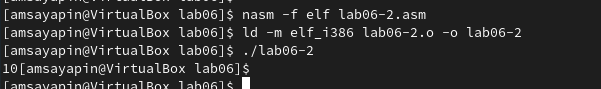


Figure 10: Компиляция и запуск программы lab6-2.asm

В рамках изучения выполнения арифметических действий в NASM была рассмотрена программа для расчета арифметической функции

.



Figure 11: Код программы lab6-3.asm

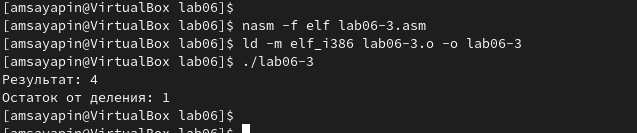


Figure 12: Компиляция и запуск программы lab6-3.asm

Затем модифицировал код программы для того, чтобы она могла вычислять функцию. После создания исполняемого файла он провел его тестирование.

.

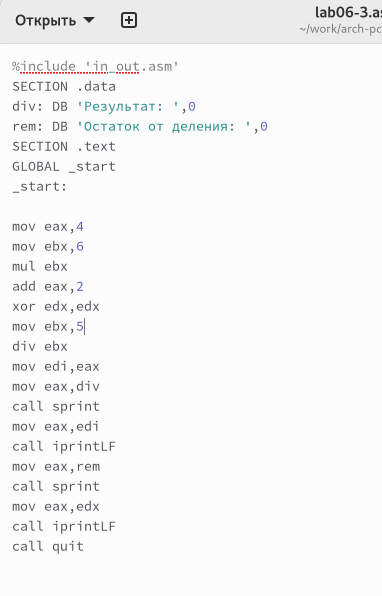


Figure 13: Код программы lab6-3.asm

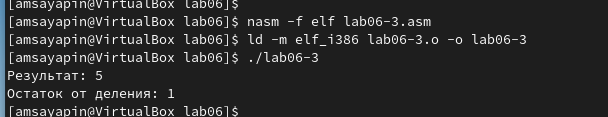


Figure 14: Компиляция и запуск программы lab6-3.asm

В качестве другого примера мы рассмотрели программу вычисления варианта задания по номеру студенческого билета.

В этом примере число для выполнения арифметических действий пользователь вводит с клавиатуры. Как было отмечено ранее, ввод осуществляется в виде символов, и для того чтобы арифметические операции в NASM были выполнены правильно, эти символы необходимо преобразовать в числовой формат. Для конвертации можно применить функцию atoi, которая содержится в файле in\_out.asm.

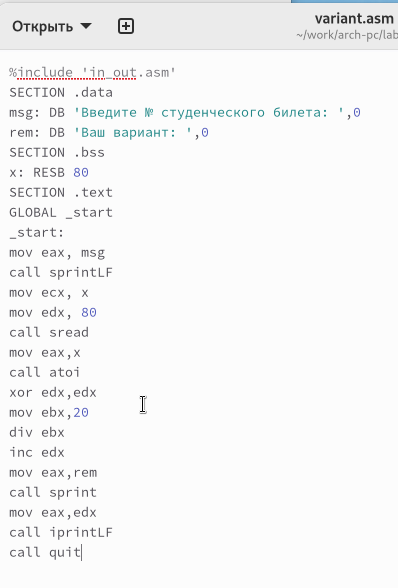


Figure 15: Код программы variant.asm

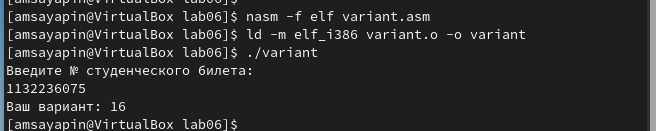


Figure 16: Компиляция и запуск программы variant.asm

ответы на вопросы

1. Какие строки листинга отвечают за вывод на экран сообщения ‘Ваш вариант:’?

*Команда mov eax, rem загружает в регистр eax строку с текстом “Ваш вариант:”. После этого команда call sprint инициирует процедуру, которая выводит эту строку на экран.*

1. Для чего используется следующие инструкции?

*Команда mov ecx, x копирует значение из регистра ecx в переменную x. Команда mov edx, 80 присваивает регистру edx число 80. Команда call sread запускает процедуру чтения данных из стандартного ввода.*

1. Для чего используется инструкция “call atoi”?

*Команда call atoi конвертирует строку символов в целое число.*

1. Какие строки листинга отвечают за вычисления варианта?

*Команды, выполняющие вычисление варианта, включают: xor edx, edx для обнуления регистра edx, mov ebx, 20 для присвоения регистру ebx числа 20, div ebx для деления значения в аккумуляторе на значение в ebx, и inc edx для увеличения результата в регистре edx на единицу.*

1. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

*Остаток от деления после выполнения команды div ebx записывается в регистр edx.*

1. Для чего используется инструкция “inc edx”?

*Команда inc edx служит для инкремента, то есть увеличения значения в регистре edx на один. Это необходимо для корректного расчета варианта по заданной формуле, где к остатку от деления добавляется единица.*

1. Какие строки листинга отвечают за вывод на экран результата вычислений?

*Команда mov eax, edx помещает результат вычислений в регистр eax. Затем команда call iprintLF активирует процедуру, которая выводит результат на экран с переводом строки.*

## 2.1 Задание для самостоятельной работы

Написать программу вычисления выражения y = f(x). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x, вычислять заданное выражение в зависимости от введенного x, выводить результат вычислений. Вид функции f(x) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x1 и x2 из 6.3.

Вариант 16 - для

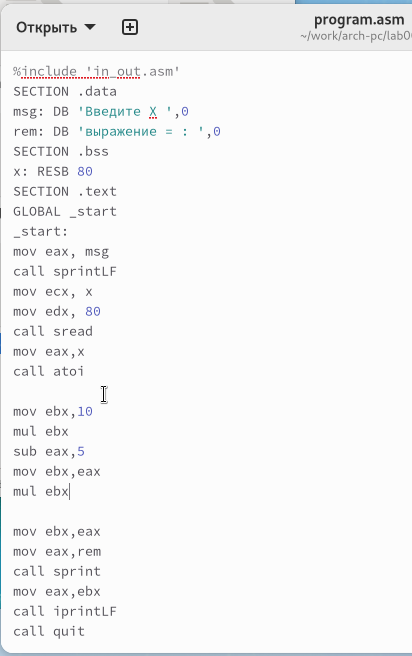


Figure 17: Код программы program.asm

при

при

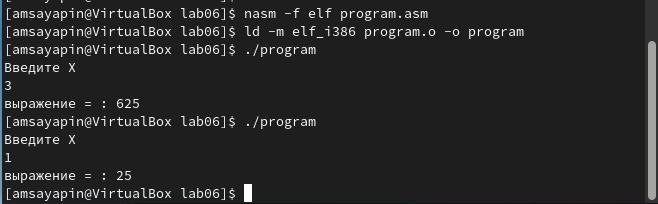


Figure 18: Компиляция и запуск программы program.asm

Программа считает верно.

# 3 Выводы

Изучили работу с арифметическими операциями.