Отчёт по лабораторной работе 6

Архитектура компьютера

Рыбалко Николай НБИбд-02-23

Содержание

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

Я создал каталог для программ лабораторной работы № 6, перешел в него и создал файл lab6-1.asm.

Давайте рассмотрим примеры программ, которые выводят символьные и числовые значения. В этих программах значения будут записываться в регистр eax.

В данной программе мы записываем символ ‘6’ в регистр eax (mov eax, ‘6’), а символ ‘4’ в регистр ebx (mov ebx, ‘4’). Затем мы добавляем значение регистра ebx к значению в регистре eax (add eax, ebx, результат сложения записывается в регистр eax). После этого мы выводим результат. Однако, для работы функции sprintLF, необходимо, чтобы в регистре eax был записан адрес, поэтому мы используем дополнительную переменную. Мы записываем значение регистра eax в переменную buf1 (mov [buf1], eax), а затем записываем адрес переменной buf1 в регистр eax (mov eax, buf1) и вызываем функцию sprintLF. Указано на рис. [[1](#fig:001)] [[2](#fig:002)]



Figure 1: Код программы lab6-1.asm

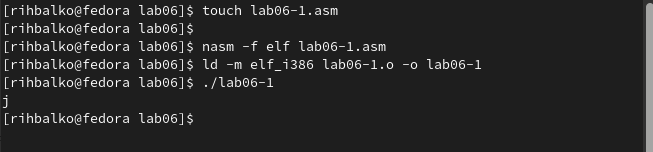


Figure 2: Сборка и проверка программы lab6-1.asm

В данном случае, когда мы ожидаем увидеть число 10 при выводе значения регистра eax, фактическим результатом будет символ ‘j’. Это происходит из-за того, что код символа ‘6’ равен 00110110 в двоичном представлении (или 54 в десятичном представлении), а код символа ‘4’ равен 00110100 (или 52 в десятичном представлении). Когда мы выполняем команду add eax, ebx, результатом будет сумма кодов - 01101010 (или 106 в десятичном представлении), который соответствует символу ‘j’.

Далее изменяю текст программы и вместо символов, запишем в регистры числа. Указано на рис. [[3](#fig:003)] [[4](#fig:004)]

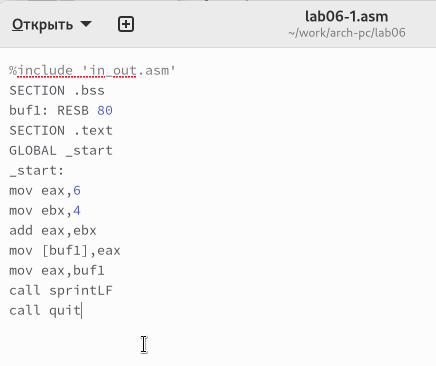


Figure 3: Код программы lab6-1.asm

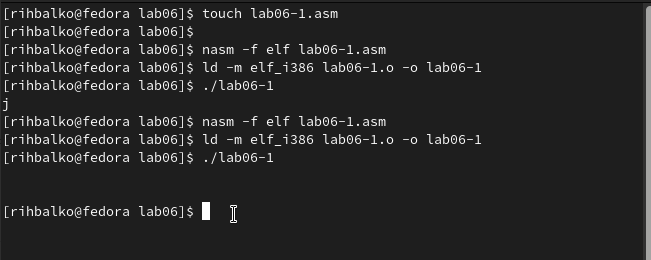


Figure 4: Сборка и проверка программы lab6-1.asm

При изменении текста программы и записи чисел в регистры, мы также не получим число 10 при выполнении программы. Вместо этого будет выведен символ с кодом 10, который представляет собой символ конца строки (возврат каретки). В консоли этот символ не отображается, но он добавляет пустую строку.

Как уже было отмечено ранее, в файле in\_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Я преобразовал текст программы, используя эти функции. Указано на рис. [[5](#fig:005)] [[6](#fig:006)]

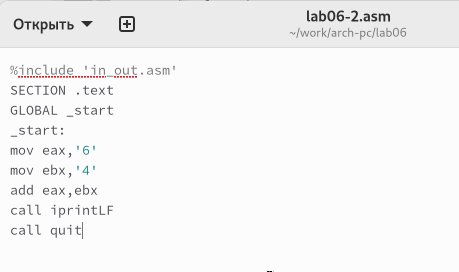


Figure 5: Код программы lab6-2.asm

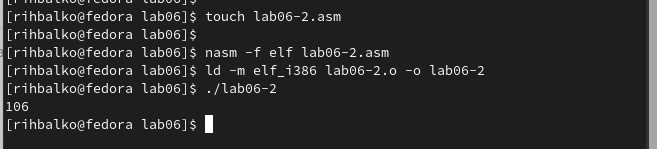


Figure 6: Сборка и проверка программы lab6-2.asm

В результате выполнения программы мы получим число 106. В данном случае, как и в первом случае, команда add складывает коды символов ‘6’ и ‘4’ (54+52=106). Однако, в отличие от предыдущей программы, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число.

Аналогично предыдущему примеру изменим символы на числа. Указано на рис. [[7](#fig:007)]

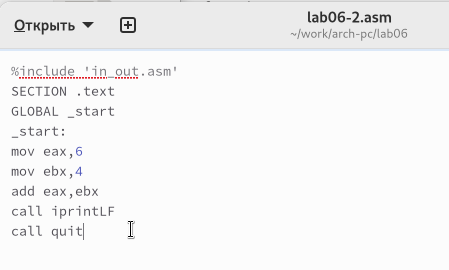


Figure 7: Код программы lab6-2.asm

Функция iprintLF позволяет вывести число и операндами были числа (а не коды символов). Поэтому получаем число 10. Указано на рис. [[8](#fig:008)]

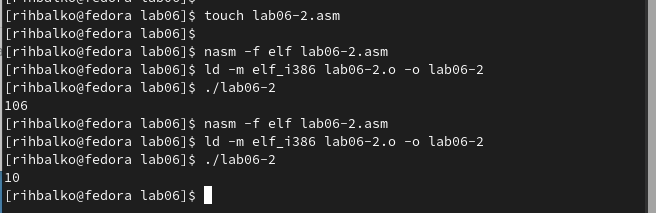


Figure 8: Сборка и проверка программы lab6-2.asm

Заменил функцию iprintLF на iprint. Создал исполняемый файл и запустил его. Вывод отличается тем, что нет переноса строки. Указано на рис. [[9](#fig:009)]

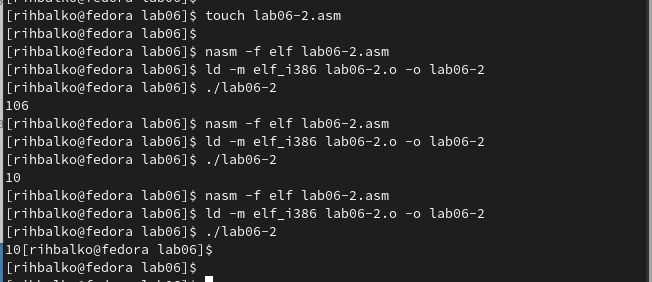


Figure 9: Сборка и проверка программы lab6-2.asm

В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения . Указано на рис. [[10](#fig:010)] [[11](#fig:011)]

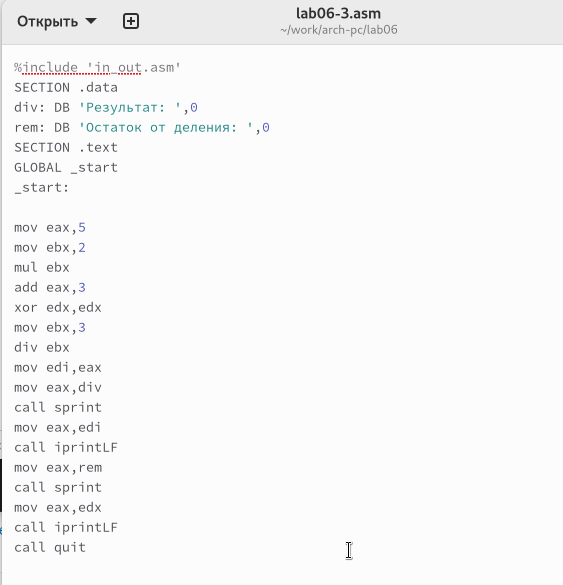


Figure 10: Код программы lab6-3.asm

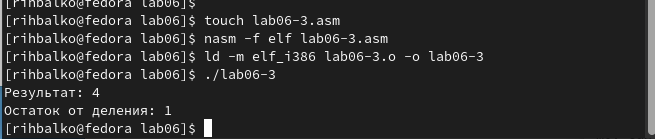


Figure 11: Сборка и проверка программы lab6-3.asm

Изменил текст программы для вычисления выражения . Создал исполняемый файл и проверил его работу. Указано на рис. [[12](#fig:012)] [[13](#fig:013)]

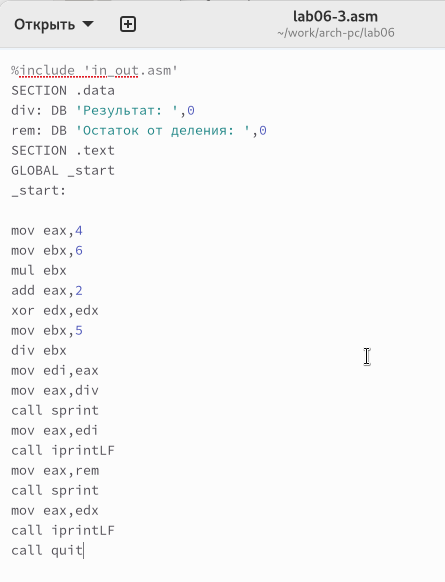


Figure 12: Код программы lab6-3.asm

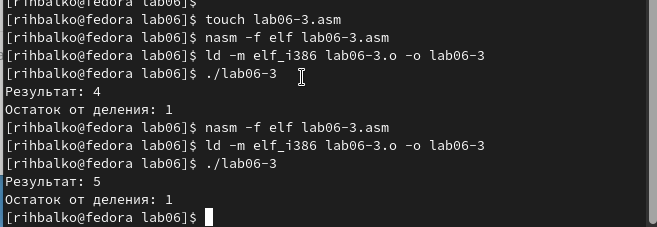


Figure 13: Сборка и проверка программы lab6-3.asm

В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета. Указано на рис. [[14](#fig:014)] [[15](#fig:015)]

В данном случае число, над которым нужно выполнить арифметические операции, вводится с клавиатуры. Как было отмечено ранее, ввод с клавиатуры осуществляется в символьном виде, и для правильной работы арифметических операций в NASM символы должны быть преобразованы в числа. Для этой цели можно использовать функцию atoi из файла in\_out.asm.

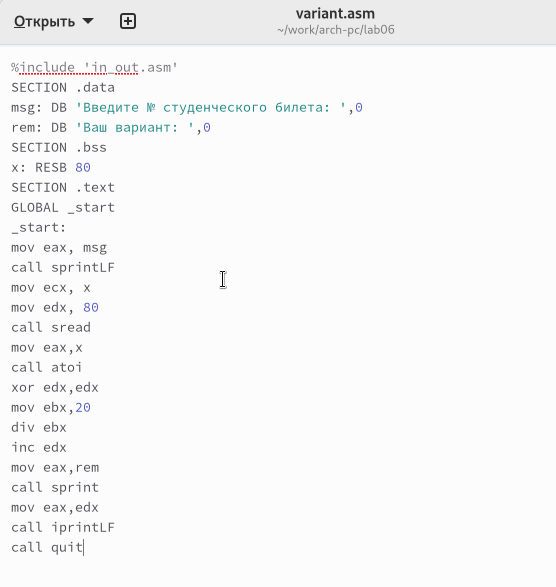


Figure 14: Код программы variant.asm

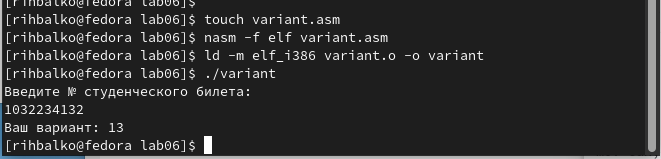


Figure 15: Сборка и проверка программы variant.asm

ответы на вопросы

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

Строка “mov eax, rem” перекладывает значение переменной с фразой ‘Ваш вариант:’ в регистр eax.

Строка “call sprint” вызывает подпрограмму для вывода строки.

1. Для чего используется следующие инструкции?

Инструкция “mov ecx, x” используется для перемещения значения переменной x в регистр ecx.

Инструкция “mov edx, 80” используется для перемещения значения 80 в регистр edx.

Инструкция “call sread” вызывает подпрограмму для считывания значения студенческого билета из консоли.sread

1. Для чего используется инструкция “call atoi”?

Инструкция “call atoi” используется для преобразования введенных символов в числовой формат.

1. Какие строки листинга отвечают за вычисления варианта?

Строка “xor edx, edx” обнуляет регистр edx.

Строка “mov ebx, 20” записывает значение 20 в регистр ebx.

Строка “div ebx” выполняет деление номера студенческого билета на 20.

Строка “inc edx” увеличивает значение регистра edx на 1.

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

Остаток от деления записывается в регистр edx.

1. Для чего используется инструкция “inc edx”?

Инструкция “inc edx” используется для увеличения значения в регистре edx на 1, согласно формуле вычисления варианта.

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

Строка “mov eax, edx” перекладывает результат вычислений в регистр eax.

Строка “call iprintLF” вызывает подпрограмму для вывода значения на экран.

## 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. Указано на рис. [[16](#fig:016)] [[17](#fig:017)]

Получили вариант 13 - для

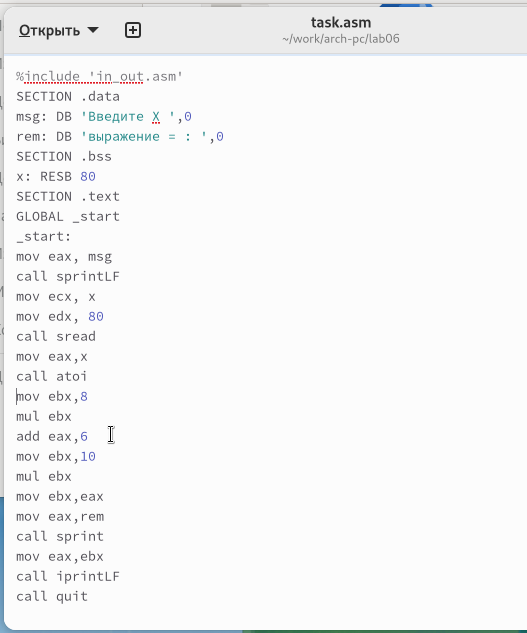


Figure 16: Код программы task.asm

Также размещаю код программы в отчете.

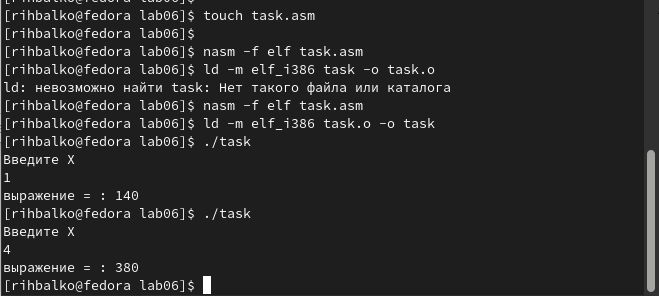


Figure 17: Сборка и проверка программы task.asm

# 3 Выводы

Изучили работу с арифметическими операциями.