Отчёт по лабораторной работе 6

Архитектура компьютера

Диденко Дмитрий Владимирович НПИбд-03-23

Содержание

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

## 2.1 Символьные и численные данные в NASM

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

Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения, записанные в регистр eax.

В данной программе (рис. [[1](#fig:001)]) в регистр eax записывается символ 6 (mov eax,‘6’), в регистр ebx символ 4 (mov ebx,‘4’). Далее к значению в регистре eax прибавляем значение регистра ebx (add eax,ebx, результат сложения запишется в регистр eax). Далее выводим результат. (рис. [[2](#fig:002)])

Так как для работы функции sprintLF в регистр eax должен быть записан адрес, необходимо использовать дополнительную переменную. Для этого запишем значение регистра eax в переменную buf1 (mov [buf1],eax), а затем запишем адрес переменной buf1 в регистр eax (mov eax,buf1) и вызовем функцию sprintLF.

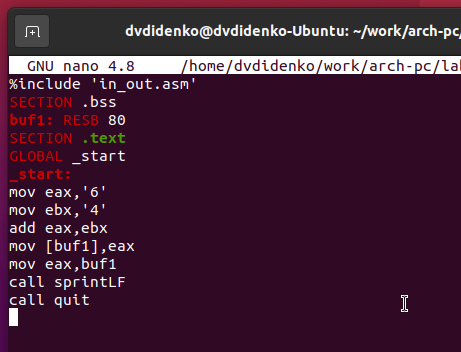


Figure 1: Код программы lab6-1.asm

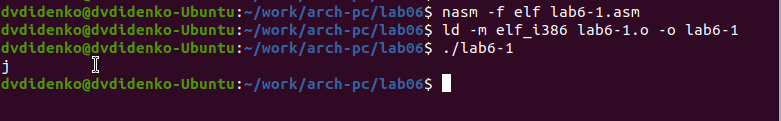


Figure 2: Компиляция и запуск программы lab6-1.asm

В данном случае при выводе значения регистра eax мы ожидаем увидеть число 10. Однако результатом будет символ j. Это происходит потому, что код символа 6 равен 00110110 в двоичном представлении (или 54 в десятичном представлении), а код символа 4 – 00110100 (52). Команда add eax,ebx запишет в регистр eax сумму кодов – 01101010 (106), что в свою очередь является кодом символа j.

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

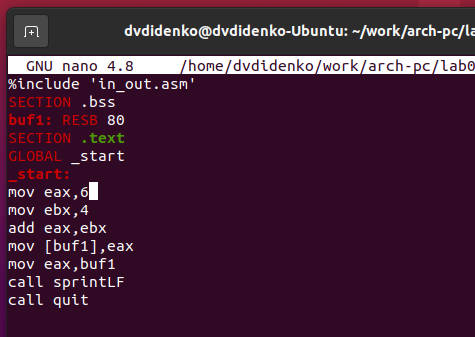


Figure 3: Код программы lab6-1.asm

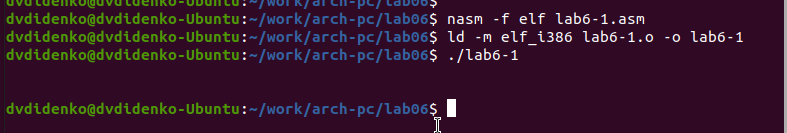


Figure 4: Компиляция и запуск программы lab6-1.asm

Как и в предыдущем случае при исполнении программы мы не получим число 10. В данном случае выводится символ с кодом 10 (рис. [[4](#fig:004)]). Это символ конца строки (возврат каретки). В консоле он не отображается, но добавляет пустую строку.

Как отмечалось выше, для работы с числами в файле in\_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразовал текст программы с использованием этих функций. (рис. [[5](#fig:005)])

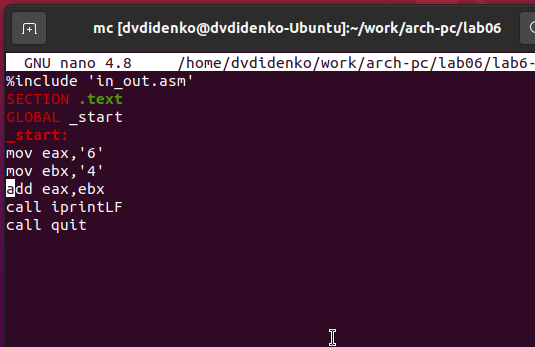


Figure 5: Код программы lab6-2.asm

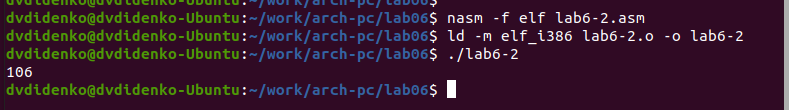


Figure 6: Компиляция и запуск программы lab6-2.asm

В результате работы программы мы получим число 106.(рис. [[6](#fig:006)]) В данном случае, как и в первом, команда add складывает коды символов ‘6’ и ‘4’ (54+52=106). Однако, в отличии от прошлой программы, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число.

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

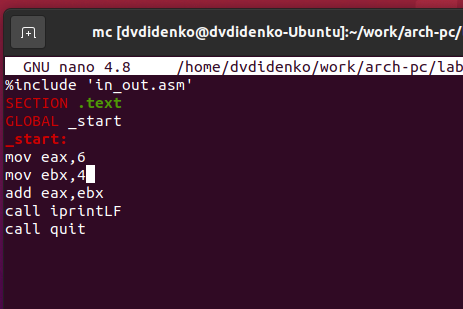


Figure 7: Код программы lab6-2.asm

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

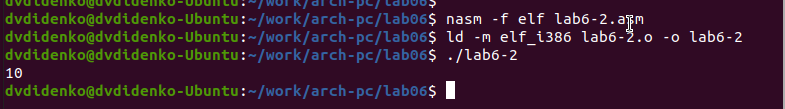


Figure 8: Компиляция и запуск программы lab6-2.asm

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

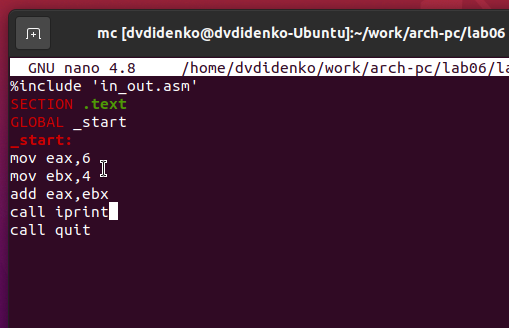


Figure 9: Код программы lab6-2.asm

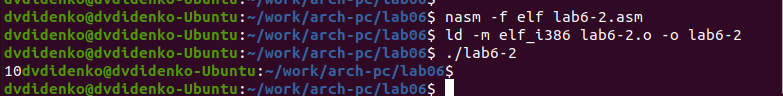


Figure 10: Компиляция и запуск программы lab6-2.asm

## 2.2 Выполнение арифметических операций в NASM

В качестве примера выполнения арифметических операций в NASM приведем программу вычисления арифметического выражения

. (рис. [[11](#fig:011)]) (рис. [[12](#fig:012)])

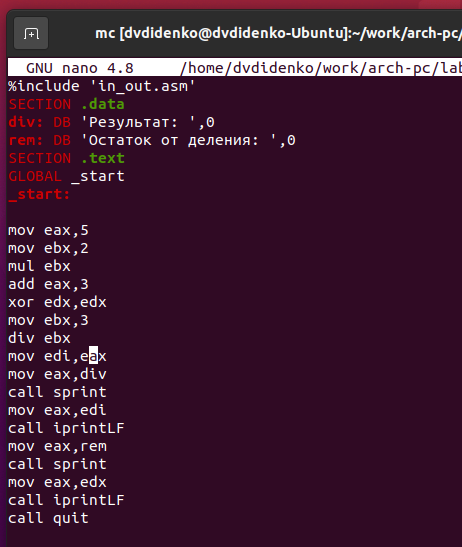


Figure 11: Код программы lab6-3.asm

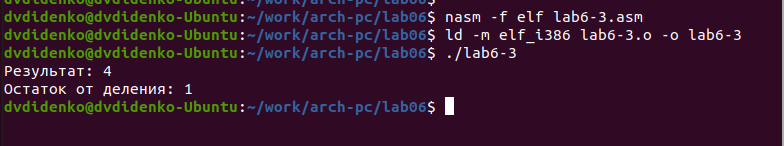


Figure 12: Компиляция и запуск программы lab6-3.asm

Изменил текст программы для вычисления выражения

. Создал исполняемый файл и проверил его работу. (рис. [[13](#fig:013)]) (рис. [[14](#fig:014)])

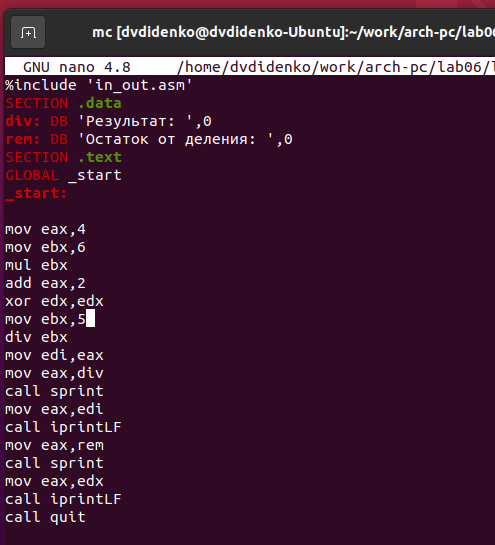


Figure 13: Код программы lab6-3.asm

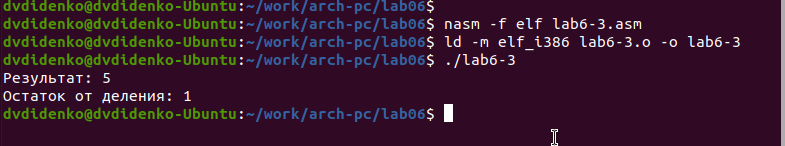


Figure 14: Компиляция и запуск программы lab6-3.asm

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

В данном случае число, над которым необходимо проводить арифметические операции, вводится с клавиатуры. Как отмечалось выше ввод с клавиатуры осуществляется в символьном виде и для корректной работы арифметических операций в NASM символы необходимо преобразовать в числа. Для этого может быть использована функция atoi из файла in\_out.asm. (рис. [[15](#fig:015)]) (рис. [[16](#fig:016)])

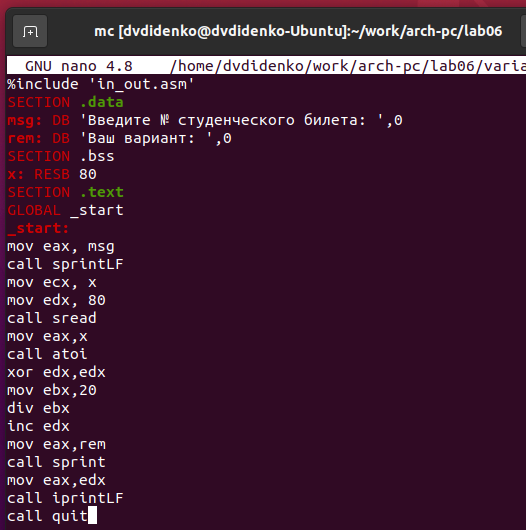


Figure 15: Код программы variant.asm

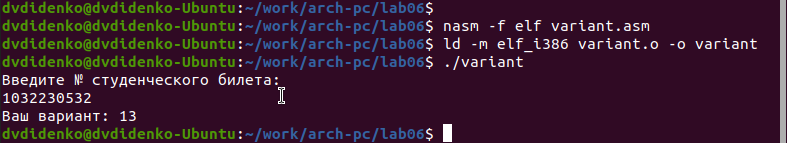


Figure 16: Компиляция и запуск программы variant.asm

ответы на вопросы

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

Переменная с фразой “Ваш вариант:” перекладывается в регистр eax с помощью строки mov eax, rem.

Для вызова подпрограммы вывода строки используется строка call sprint.

1. Для чего используется следующие инструкции?

* mov ecx, x - перекладывает регистр ecx в переменную
* mov edx, 80 - устанавливает значение 80 в регистр edx.
* call sread - вызывает подпрограмму для считывания значения из консоли.

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?

* xor edx, edx - обнуляет регистр edx.
* mov ebx, 20 - устанавливает значение 20 в регистр ebx.
* div ebx - выполняет деление номера студенческого билета на 20.
* 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 регистра edx на 1. В данном случае она используется для выполнения формулы вычисления варианта, где требуется добавить 1 к остатку от деления.

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

* Результат вычислений перекладывается в регистр eax с помощью строки mov eax, edx.
* Для вызова подпрограммы вывода используется строка call iprintLF.

## 2.3 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. (рис. [[17](#fig:017)]) (рис. [[18](#fig:018)]) Создайте исполняемый файл и проверьте его работу для значений x1 и x2 из 6.3.

Вариант 13 - (8x + 6)\*10 для x=1, x=4

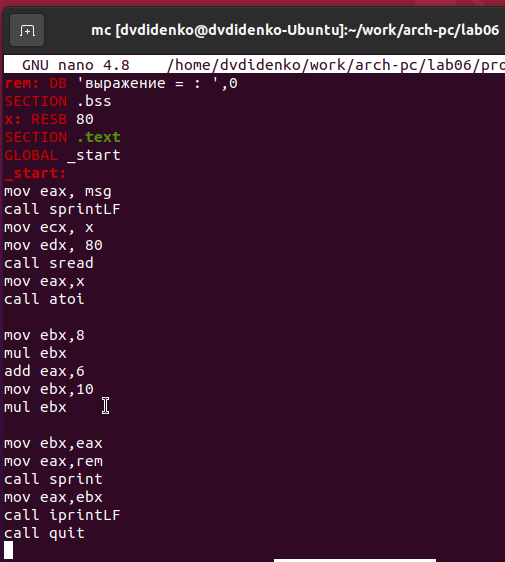


Figure 17: Код программы program.asm

при x=1 f(x) = 140

при x=4 f(x) = 380

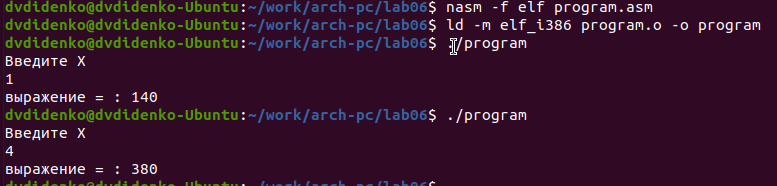


Figure 18: Компиляция и запуск программы program.asm

Программа считает верно.

# 3 Выводы

Изучили работу с арифметическими операциями.