Отчёт по лабораторной работе 6

Архитектура компьютера

Чермашенцев Павел Андреевич НБИбд-03-24

Содержание

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

Создан каталог для программ лабораторной работы № 6, в который был добавлен файл lab6-1.asm.

Рассмотрим примеры программ, выводящих символьные и числовые значения. Эти программы будут выводить данные, записанные в регистр eax.

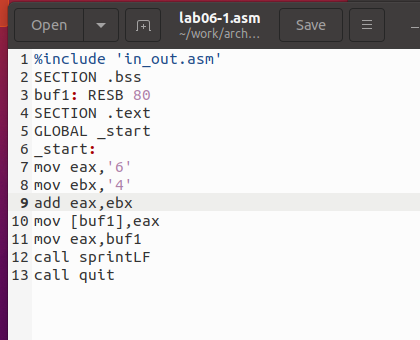


Рис. 1: Программа в файле lab6-1.asm

В данной программе (рис. 1) в регистр eax записывается символ ‘6’ (команда mov eax, '6'), а в регистр ebx — символ ‘4’ (команда mov ebx, '4'). Затем происходит сложение значений регистров eax и ebx (команда add eax, ebx), результат операции записывается в регистр eax. После этого выводится результат. Для использования функции sprintLF необходимо, чтобы в регистре eax находился адрес, поэтому используется дополнительная переменная. Значение регистра eax записывается в переменную buf1 (команда mov [buf1], eax), а затем в регистр eax записывается адрес переменной buf1 (команда mov eax, buf1), после чего вызывается функция sprintLF.

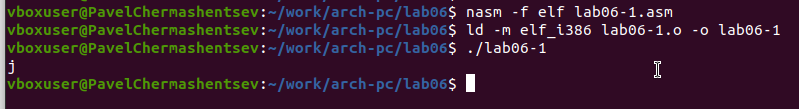


Рис. 2: Запуск программы lab6-1.asm

В случае, когда ожидаем получить число 10 при выводе содержимого регистра eax, фактический результат будет символ ‘j’. Это объясняется тем, что код символа ‘6’ в двоичном представлении равен 00110110 (54 в десятичном), а код символа ‘4’ — 00110100 (52 в десятичном). При выполнении команды add eax, ebx результатом будет сумма этих кодов — 01101010 (106 в десятичном), что соответствует символу ‘j’ (рис. 2).

Далее изменяем программу, заменяя символы на числа (рис. 3).

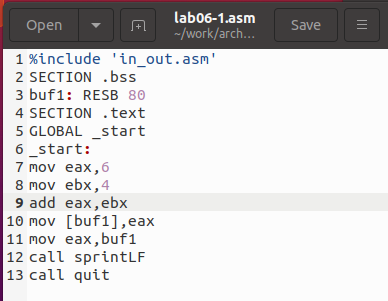


Рис. 3: Программа в файле lab6-1.asm

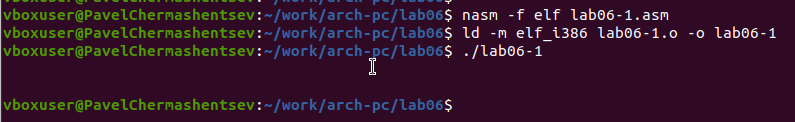


Рис. 4: Запуск программы lab6-1.asm

Как и в предыдущем примере, при выполнении программы мы не получаем число 10. Вместо этого выводится символ с кодом 10, который представляет собой символ конца строки (возврат каретки). Этот символ не отображается в консоли, но добавляет пустую строку.

Как упоминалось ранее, для работы с числами в файле in\_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразовал программу с использованием этих функций (рис. 5).

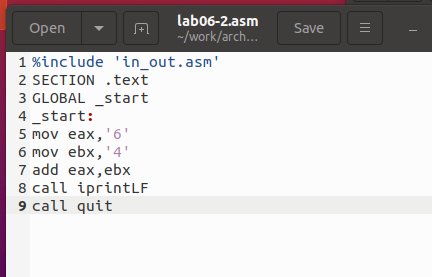


Рис. 5: Программа в файле lab6-2.asm

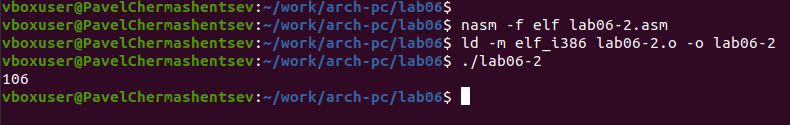


Рис. 6: Запуск программы lab6-2.asm

В результате выполнения программы выводится число 106 (рис. 6). В этом случае команда add суммирует коды символов ‘6’ и ‘4’ (54 + 52 = 106). Однако, в отличие от предыдущей программы, функция iprintLF позволяет вывести число, а не символ, который соответствует этому числу.

Аналогично предыдущему примеру изменяем символы на числа (рис. 7).

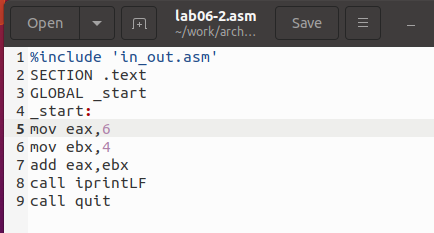


Рис. 7: Программа в файле lab6-2.asm

Функция iprintLF позволяет вывести число, и операндами являются числа, а не коды символов. Поэтому на экране будет выведено число 10 (рис. 8).

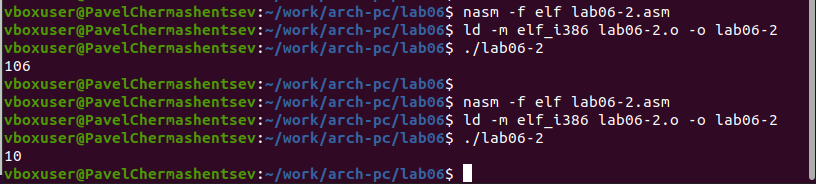


Рис. 8: Запуск программы lab6-2.asm

Заменил функцию iprintLF на iprint, создал исполняемый файл и запустил его. Вывод отличается тем, что теперь нет переноса строки (рис. 9).

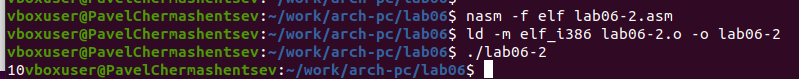


Рис. 9: Запуск программы lab6-2.asm

Для примера арифметических операций в NASM привожу программу для вычисления выражения (рис. 10) (рис. 11):

.

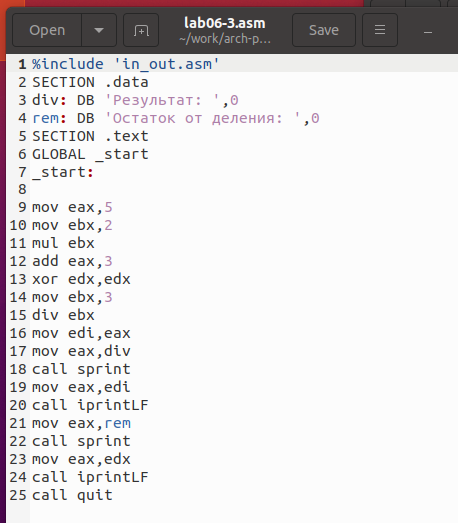


Рис. 10: Программа в файле lab6-3.asm

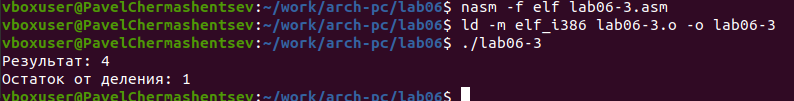


Рис. 11: Запуск программы lab6-3.asm

Изменил программу для вычисления выражения:

.

Создал исполняемый файл и проверил его работу (рис. 12) (рис. 13).

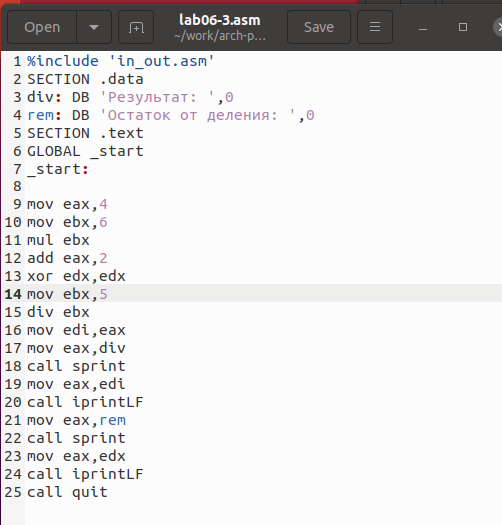


Рис. 12: Программа в файле lab6-3.asm

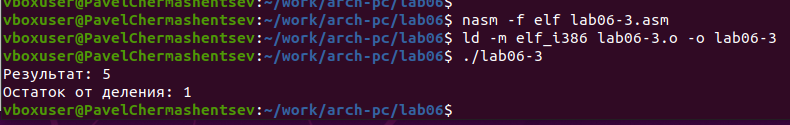


Рис. 13: Запуск программы lab6-3.asm

Рассмотрим еще один пример программы для вычисления варианта задания по номеру студенческого билета (рис. 14) (рис. 15).

В этом примере число, с которым нужно проводить арифметические операции, вводится с клавиатуры. Так как ввод осуществляется в виде символов, для корректной работы арифметических операций символы необходимо преобразовать в числа. Для этого используется функция atoi из файла in\_out.asm.

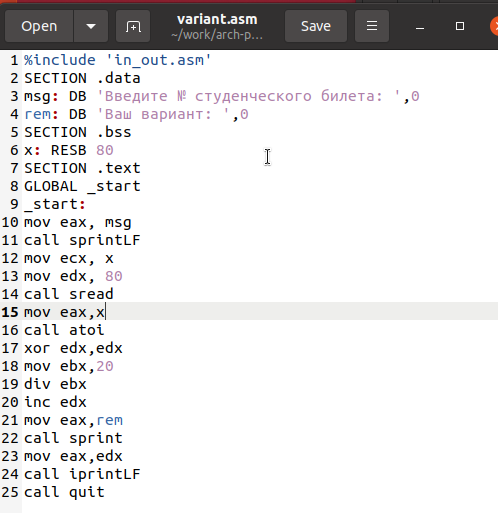


Рис. 14: Программа в файле variant.asm

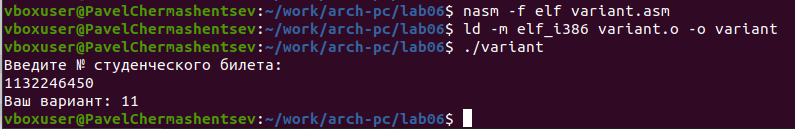


Рис. 15: Запуск программы variant.asm

## 2.1 Ответы на вопросы по программе variant.asm

1. **Какие строки листинга отвечают за вывод на экран сообщения «Ваш вариант:»?**

Строка mov eax, rem записывает в регистр значение переменной с фразой «Ваш вариант:», а строка call sprint вызывает подпрограмму для вывода этой строки на экран.

1. **Для чего используются следующие инструкции?**

* nasm: используется для компиляции кода на языке ассемблера NASM.
* mov ecx, x: перемещает значение переменной x в регистр ecx.
* mov edx, 80: перемещает значение 80 в регистр edx.
* call sread: вызывает подпрограмму для считывания значения студенческого билета с 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: записывает 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 увеличивает значение в регистре edx на 1, что соответствует формуле вычисления варианта.

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

Строка mov eax, edx перекладывает результат вычислений в регистр eax, а строка call iprintLF вызывает подпрограмму для вывода этого результата на экран.

## 2.2 Самостоятельное задание

Написана программа для вычисления выражения $ y = f(x) $. Программа выводит формулу для вычисления, запрашивает ввод значения $ x $, вычисляет выражение в зависимости от введенного $ x $ и выводит результат. В зависимости от лабораторного задания, был выбран вариант 11 — $ 10(x + 1) - 10 $ для $ x\_1 = 1 $, $ x\_2 = 7 $ (рис. 16) (рис. 17).

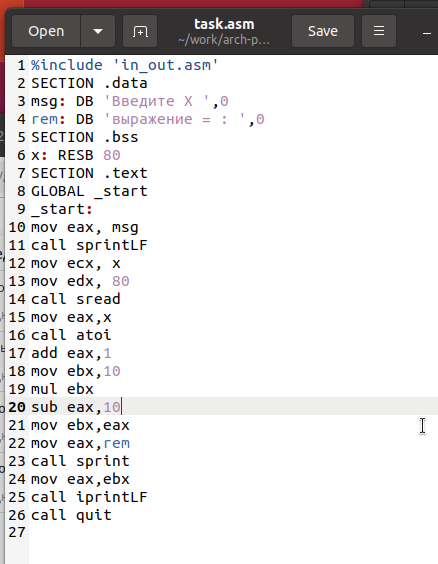


Рис. 16: Программа в файле task.asm

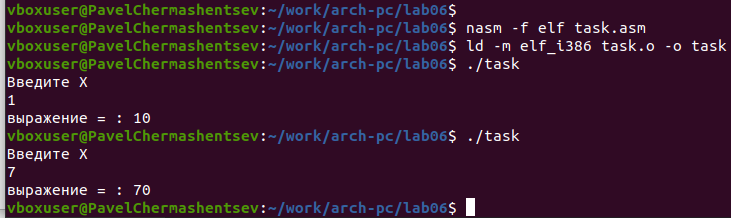


Рис. 17: Запуск программы task.asm

# 3 Выводы

Изучили работу с арифметическими операциями.