Отчёт по лабораторной работе 6

Дисциплина архитектура компьютера

Неустроева Ирина

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

*1* Создала каталог для программ лабораторной работы №6, перешла в него и создала файл с названием “lab6-1.asm”.

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

В данной программе, в регистр eax записан символ ‘6’, а в регистр ebx символ ‘4’. Затем мы прибавляем значение регистра ebx к значению в регистре eax (результат сложения будет записан в регистр eax). После этого мы выводим результат.

Так как для работы функции sprintLF в регистр eax должен быть записан адрес, мы используем дополнительную переменную. Мы записали значение регистра eax в переменную с именем “buf1”, а затем записали адрес переменной buf1 в регистр eax и вызвали функцию sprintLF. (рис. [[1](#fig:001)]) (рис. [[2](#fig:002)])

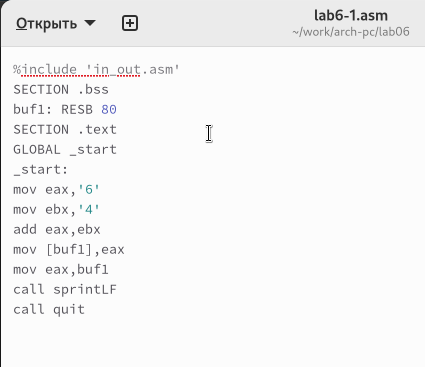


Figure 1: Редактирование файла lab6-1.asm

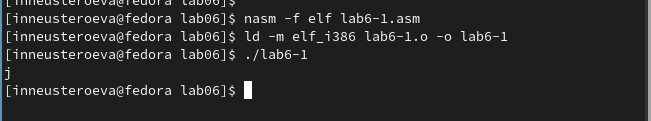


Figure 2: Проверка кода lab6-1.asm

В данном случае, при выводе значения регистра eax, ожидалось увидеть число 10. Однако, результатом был символ ‘j’. Это произошло 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 записала в регистр eax сумму кодов – 01101010 (106), что в свою очередь является кодом символа ‘j’.

*3* Далее был изменен текст программы и вместо символов записаны числа. (рис. [[3](#fig:003)]) (рис. [[4](#fig:004)])

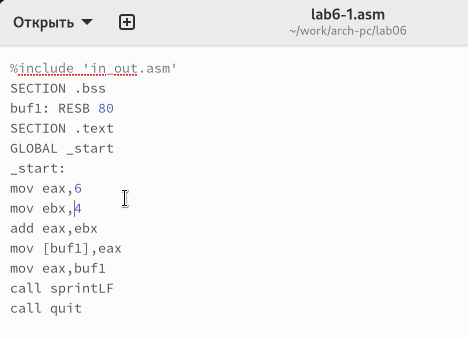


Figure 3: Редактирование файла lab6-1.asm

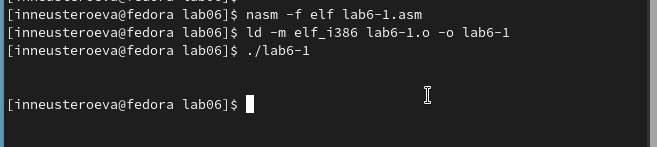


Figure 4: Проверка кода lab6-1.asm

В процессе выполнения программы не получили ожидаемое число 10. Вместо этого был выведен символ с кодом 10. Это символ конца строки (возврат каретки), который в консоли не отображается, но добавляет пустую строку.

*4* В файле “in\_out.asm” реализованы подпрограммы для работы с числами и преобразования символов ASCII. Был модифицирован текст программы с использованием этих функций. (рис. [[5](#fig:005)]) (рис. [[6](#fig:006)])

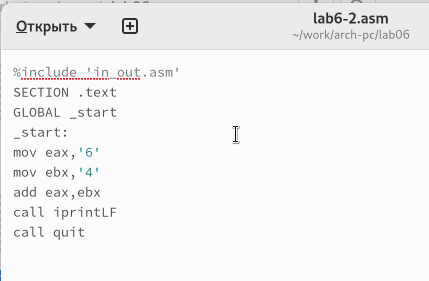


Figure 5: Редактирование файла lab6-2.asm

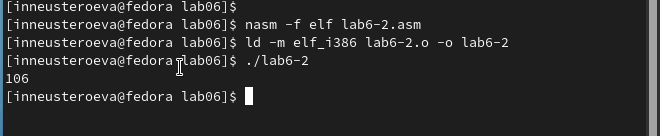


Figure 6: Проверка кода lab6-2.asm

В результате выполнения обновленной программы было выведено число 106. Здесь, как и в первом случае, команда add складывает коды символов ‘6’ и ‘4’ (54 + 52 = 106). Но в отличие от предыдущей версии, функция iprintLF позволяет напечатать само число, а не символ с соответствующим кодом.

*5* По аналогии с предыдущим примером, были заменены символы на числа. (рис. [[7](#fig:007)]) (рис. [[8](#fig:008)])

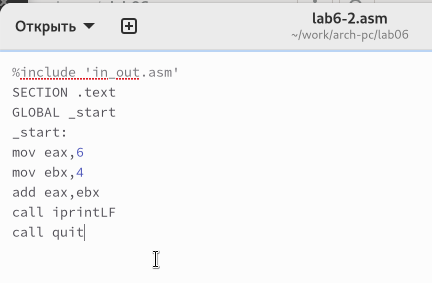


Figure 7: Редактирование файла lab6-2.asm

Функция iprintLF позволяет выводить числа, и на этот раз в качестве операндов использовались именно числа, а не коды символов. В результате мы получили число 10.

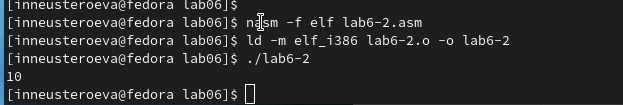


Figure 8: Проверка кода lab6-2.asm

Далее была заменена функция iprintLF на iprint, создан исполняемый файл и запущен. Вывод теперь отличается отсутствием перехода на новую строку. (рис. [[9](#fig:009)]) (рис. [[10](#fig:010)])

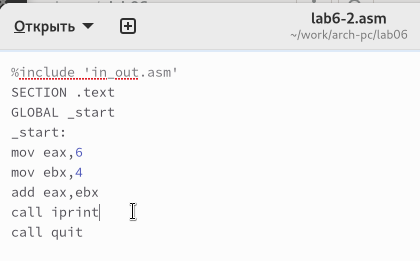


Figure 9: Редактирование файла lab6-2.asm

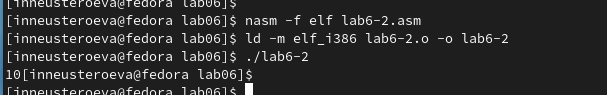


Figure 10: Проверка кода lab6-2.asm

## 2.2 Выполнение арифметических операций в NASM

*6* В качестве примера выполнения арифметических операций в NASM рассмотрим программу для вычисления арифметического выражения

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

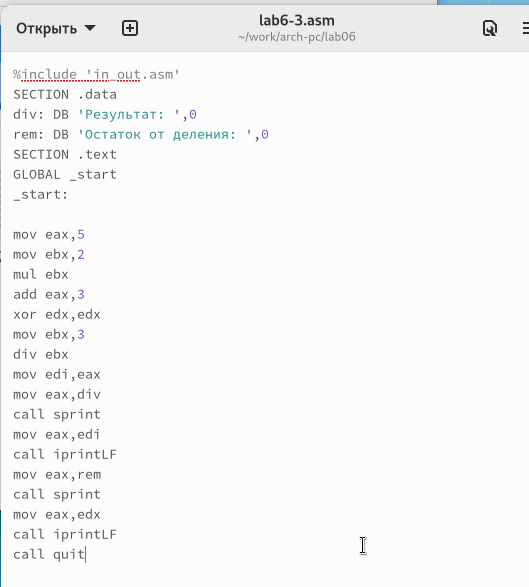


Figure 11: Редактирование файла lab6-3.asm

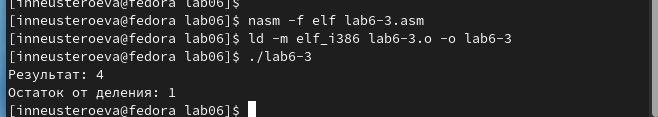


Figure 12: Проверка кода lab6-3.asm

Изменила текст программы для вычисления выражения

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

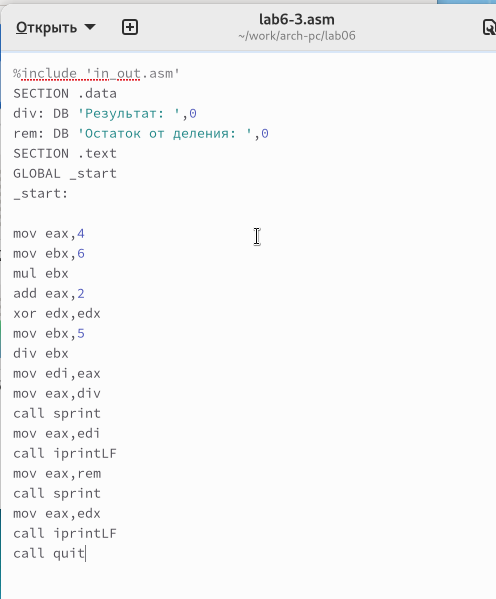


Figure 13: Редактирование файла lab6-3.asm

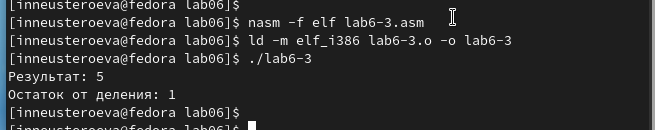


Figure 14: Проверка кода lab6-3.asm

*7* В качестве еще одного примера рассмотрим программу для вычисления варианта задания на основе номера студенческого билета.

В этом случае число, над которым нужно выполнять арифметические операции, вводится с клавиатуры. Как уже отмечалось ранее, ввод с клавиатуры осуществляется в символьном виде. Для корректной работы арифметических операций в NASM эти символы необходимо преобразовать в числовой формат. С этой целью можно использовать функцию atoi из файла in\_out.asm. Она конвертирует строку символов в эквивалентное десятичное число. (рис. [[15](#fig:015)]) (рис. [[16](#fig:016)])

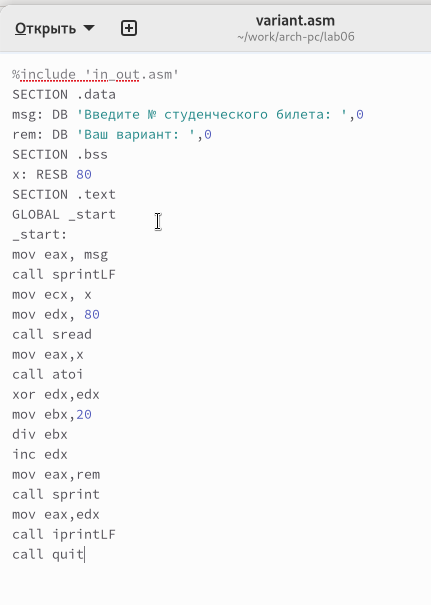


Figure 15: Редактирование файла variant.asm

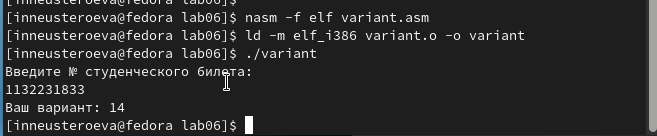


Figure 16: Проверка кода variant.asm

### 2.2.1 Ответы на вопросы по программе variant.asm

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

За отображение фразы “Ваш вариант:” отвечают те строки кода, где содержится команда mov eax, rem для передачи строки в регистр eax и последующий вызов процедуры sprint через call sprint.

1. Для чего используется следующие инструкции?

* mov ecx, x: Здесь значение регистра ecx копируется в переменную x.
* mov edx, 80: Эта команда присваивает регистру edx число 80.
* call sread: С помощью этой команды происходит вызов функции чтения данных из стандартного ввода.

1. Для чего используется инструкция “call atoi”?

Использование “call atoi” предназначено для конвертации строковых данных в целочисленное значение.

1. Какие строки листинга отвечают за вычисления варианта?

Расчет варианта осуществляется в строках с инструкциями:

* xor edx, edx: Обнуляет регистр edx.
* mov ebx, 20: Задает регистру ebx значение 20.
* div ebx: Выполняет деление с сохранением остатка.
* inc edx: Увеличивает значение регистра edx, что требуется для получения конечного варианта.

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

Остаток от деления при выполнении “div ebx” записывается в регистр edx.

1. Для чего используется инструкция “inc edx”?

Команда “inc edx” применяется для увеличения содержимого регистра edx, что используется в расчетах варианта.

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

Вывод результатов вычислений на экран осуществляется строками с командами:

* mov eax, edx: Помещает результат вычисления в регистр eax.
* call iprintLF: Вызывает функцию для отображения результата на экране.

## 2.3 Выполнение заданий для самостоятельной работы

*1* Написала программу для вычисления выражения y = f(x). Программа выводит выражение для вычисления, запрашивает ввод значения x, вычисляет заданное выражение в зависимости от введенного x и выводит результат вычислений. Для выбора вида функции f(x) использовала таблицу 6.3 вариантов заданий, в соответствии с номером, полученным при выполнении лабораторной работы.

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

Вариант 14 - для

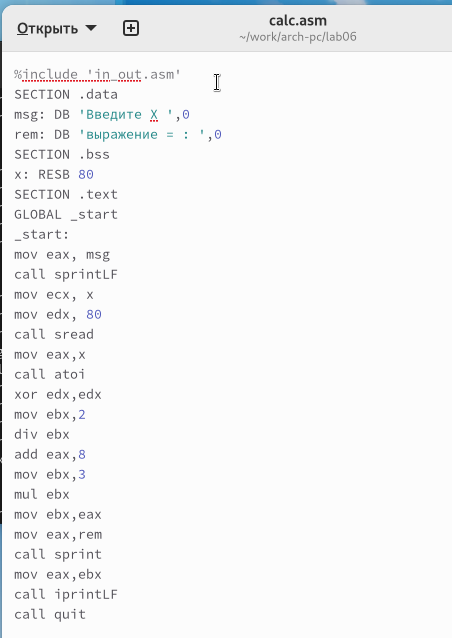


Figure 17: Редактирование файла calc.asm

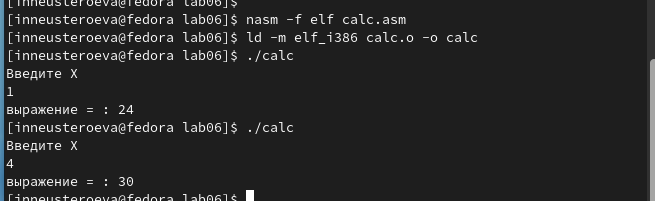


Figure 18: Проверка кода calc.asm

Программа считает верно.

# 3 Выводы

Изучили работу с арифметическими операциями.