Отчёт по лабораторной работе 6

Архитектура компьютеров и операционные системы

Старикова Владислава Александровна НММбд-03-24

Содержание

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

Я создала каталог для программ лабораторной работы № 6, перешла в него и создала файл lab6-1.asm.  
(рис. 1)

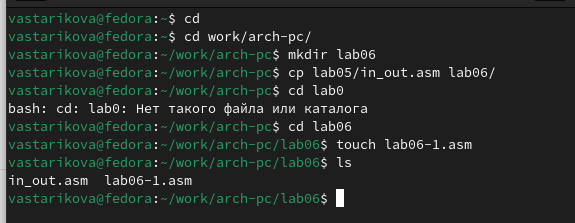


Рис. 1: Подготовила каталог

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

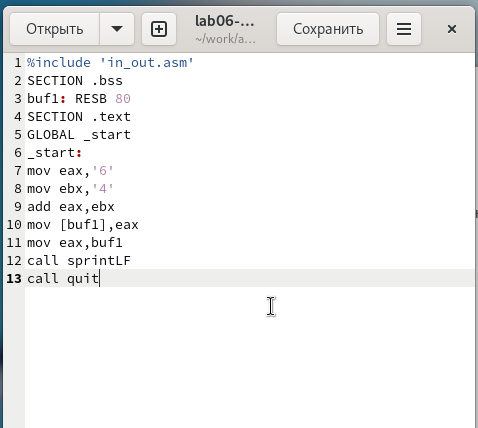


Рис. 2: Программа в файле lab6-1.asm

В данной программе (рис. 2) я записываю символ ‘6’ в регистр eax (mov eax, ‘6’), а символ ‘4’ в регистр ebx (mov ebx, ‘4’).  
Затем я добавляю значение регистра ebx к значению в регистре eax (add eax, ebx, результат сложения записывается в регистр eax).  
После этого я вывожу результат.  
Однако, для использования функции sprintLF, необходимо, чтобы в регистре eax был записан адрес, поэтому я использую дополнительную переменную.  
Я записываю значение регистра eax в переменную buf1 (mov [buf1], eax),  
а затем записываю адрес переменной buf1 в регистр eax (mov eax, buf1) и вызываю функцию sprintLF.

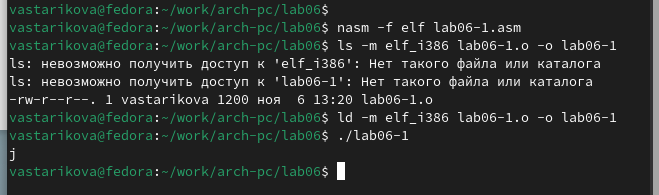


Рис. 3: Запуск программы lab6-1.asm

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 я выполняю команду add eax, ebx, результатом будет сумма кодов - 01101010 (или 106 в десятичном представлении),  
который соответствует символу ‘j’. (рис. 3)

Далее я изменила текст программы и вместо символов записала в регистры числа. (рис. 4)

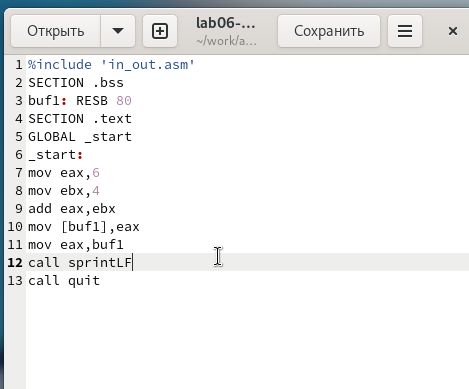


Рис. 4: Программа в файле lab6-1.asm

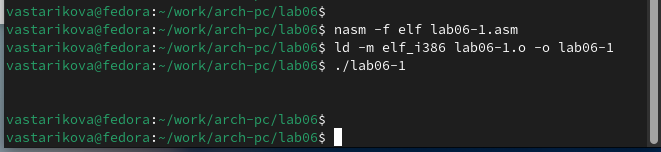


Рис. 5: Запуск программы lab6-1.asm

Как и в предыдущем случае, при выполнении программы мы не получим число 10.  
Вместо этого выводится символ с кодом 10, который представляет собой символ конца строки (возврат каретки).  
Этот символ не отображается в консоли, но он 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.  
Я преобразовала текст программы с использованием этих функций. (рис. 6)

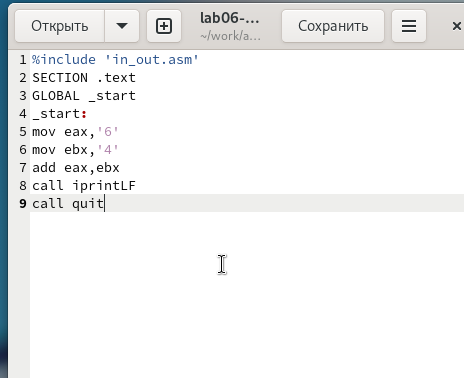


Рис. 6: Программа в файле lab6-2.asm

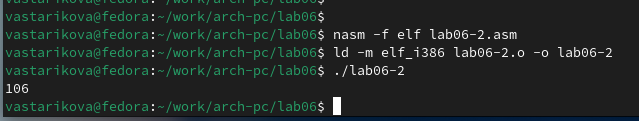


Рис. 7: Запуск программы lab6-2.asm

В результате выполнения программы я получаю число 106. (рис. 7)  
В данном случае, как и в первом примере, 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, я изменила символы на числа. (рис. 8)

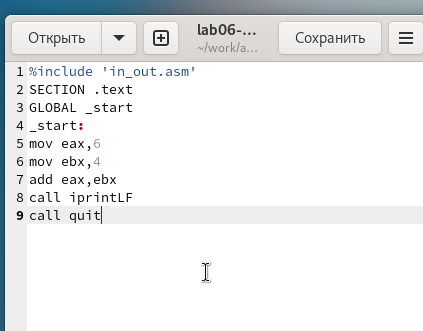


Рис. 8: Программа в файле lab6-2.asm

Функция iprintLF позволяет вывести число, и операндами были числа (а не коды символов).  
Поэтому получаем число 10. (рис. 9)

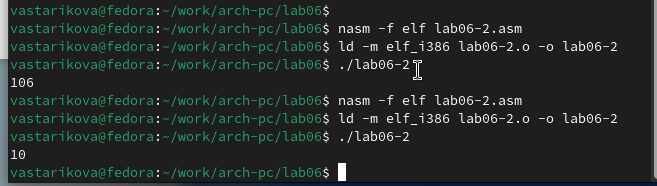


Рис. 9: Запуск программы lab6-2.asm

Я заменила функцию iprintLF на iprint. Создала исполняемый файл и запустила его.  
Вывод отличается тем, что нет переноса строки. (рис. 10)

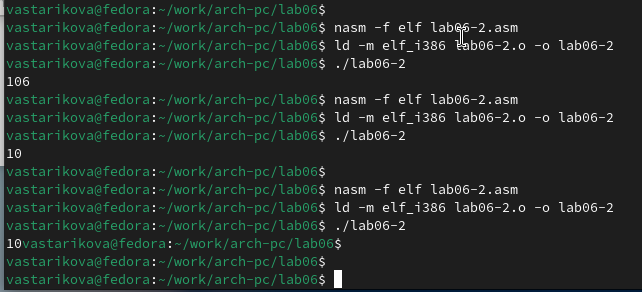


Рис. 10: Запуск программы lab6-2.asm

В качестве примера выполнения арифметических операций в NASM привожу программу вычисления арифметического выражения (рис. 11) (рис. 12)

.

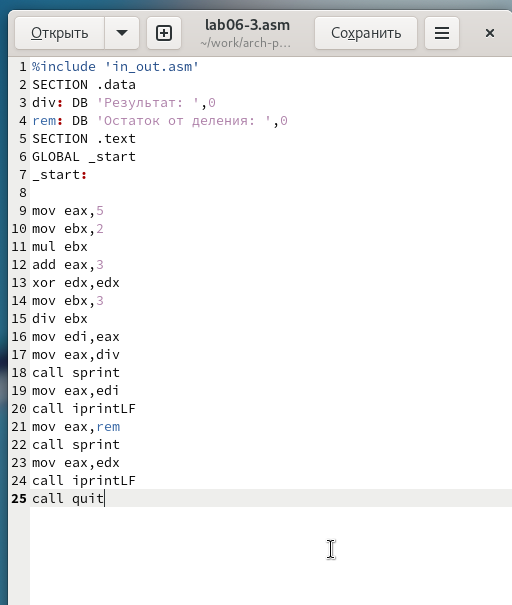


Рис. 11: Программа в файле lab6-3.asm

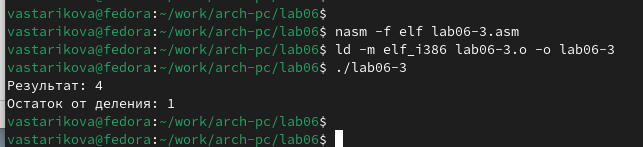


Рис. 12: Запуск программы lab6-3.asm

Я изменила текст программы для вычисления выражения

.  
Создала исполняемый файл и проверила его работу. (рис. 13) (рис. 14)

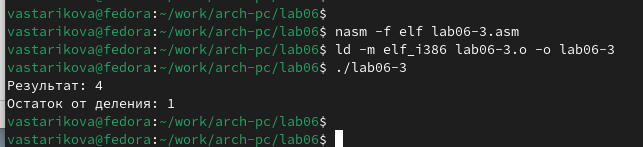


Рис. 13: Программа в файле lab6-3.asm

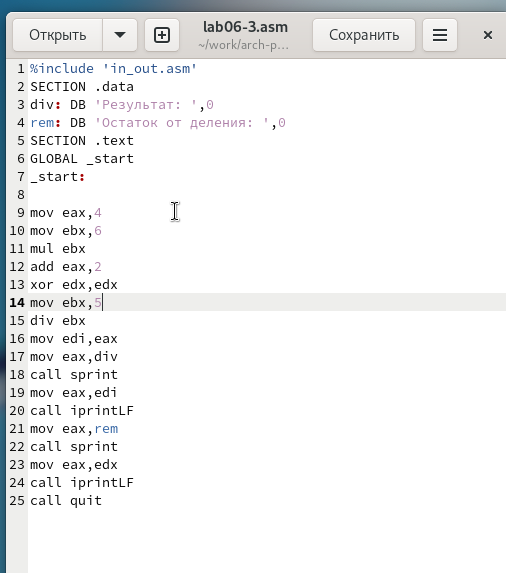


Рис. 14: Запуск программы lab6-3.asm

В качестве другого примера рассматриваю программу вычисления варианта задания по номеру студенческого билета. (рис. 15) (рис. 16)

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.

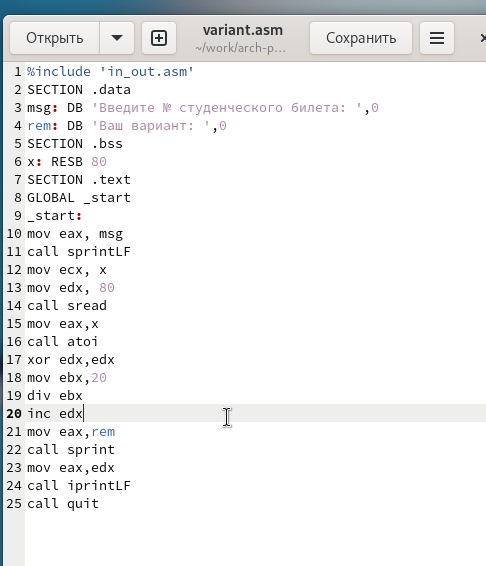


Рис. 15: Программа в файле variant.asm

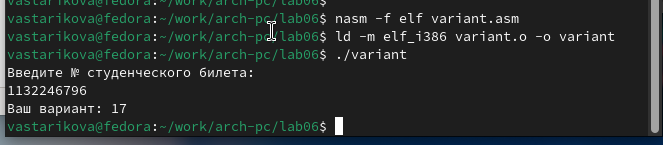


Рис. 16: Запуск программы variant.asm

## 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” перекладывает в регистр значение переменной с фразой “Ваш вариант:”.  
  Строка “call sprint” вызывает подпрограмму вывода строки.

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

* Инструкция “nasm” используется для компиляции кода на языке ассемблера NASM.  
  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.

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, в соответствии с формулой 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.

## 2.2 Самостоятельное задание

Я написала 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) выбрала из таблицы 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 6.3.

Получили вариант 17 -

для

(рис. 17) (рис. 18)

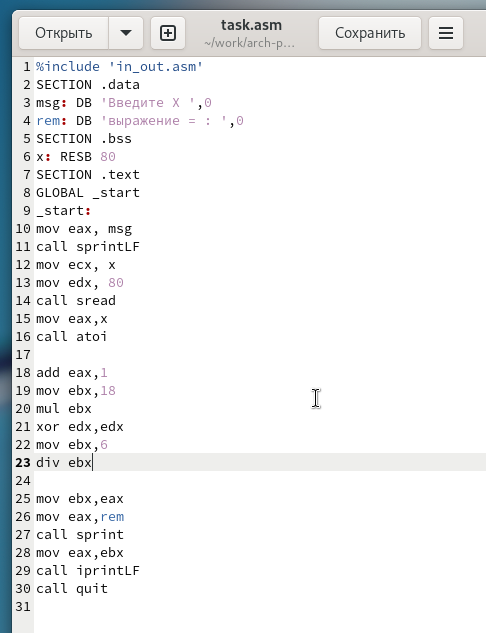


Рис. 17: Программа в файле task.asm

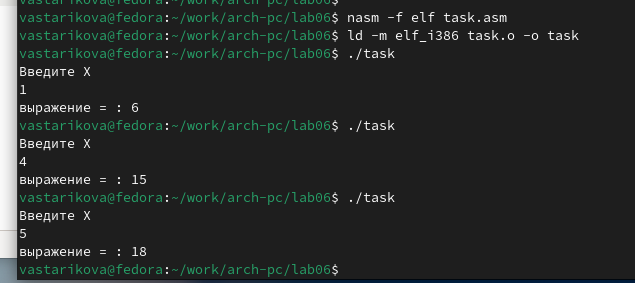


Рис. 18: Запуск программы task.asm

# 3 Выводы

Изучили работу с арифметическими операциями.