Лабораторная работа №6.

Арифметические операции в NASM.

Казначеев Сергей Ильич

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

Для начала я создал папку с названием lab06 и файл lab6-1.asm

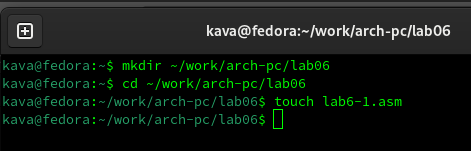


Рис. 1: scr1

Далее заходим в папку и открываем только что созданный файл и вставляем код из листинга 6.1

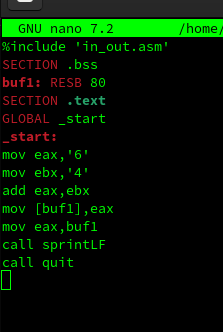


Рис. 2: scr2

После чего копируем файл in\_out.asm

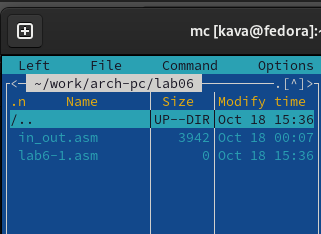


Рис. 3: scr3

Теперь соберем наш файл и запустим его мы увидим что вывелось j а нам нужно вывести сумму 6 и 4, и чтобы вывелось число 10

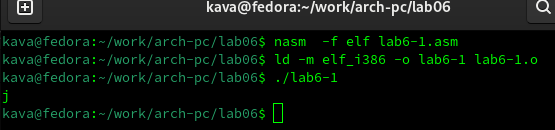


Рис. 4: scr4

Чтобы исправить это нам нужно убрать кавычки,теперь мы будет складывать числа,а не символы

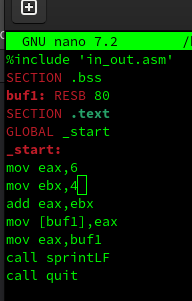


Рис. 5: scr5

После исправлений запустим файл. Увидим, что ничего не вывелось. Это произошло из-за того, что мы выводим символы, а не число.

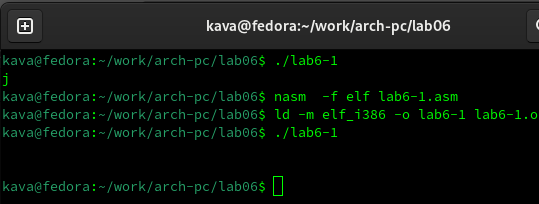


Рис. 6: scr6

Теперь создадим файл lab6-2.asm

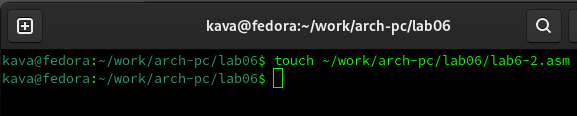


Рис. 7: scr7

После вставим в него код из листинга 6.2

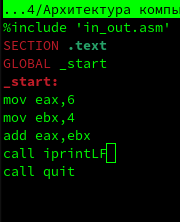


Рис. 8: scr8

Он выведет нам 106 это произойдет, так как у нас числа стоят в кавычках и мы складываем их коды (54+52=106)

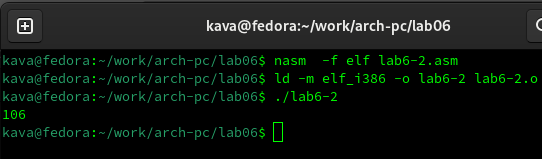


Рис. 9: scr9

Теперь,если мы уберем кавычки то у нас выведется 10

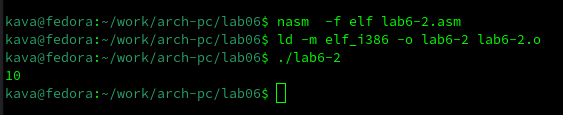
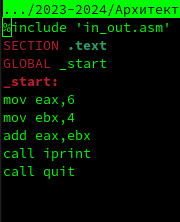


Рис. 10: scr10

Теперь посмотрим в чем разница между iprintLF и iprint

{#fig:011width=70%}

Собираем программу и запускаем

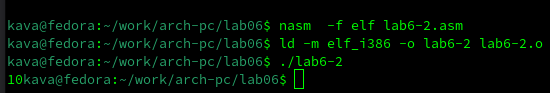


Рис. 11: scr10

Мы увидим, что оперцая iprint не переносит на следующую строку

Теперь создадим третий файл lab6-3

scr10

Рис. 12: scr10

И вставляем код из файла листинга 6.3

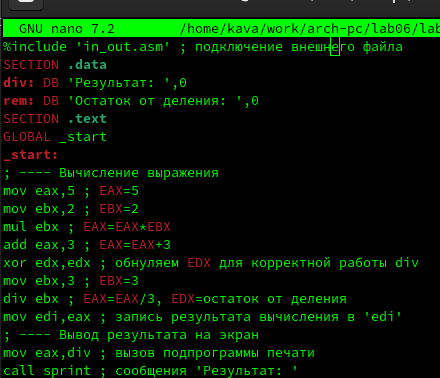
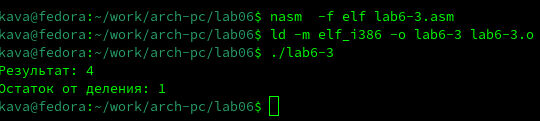


Рис. 13: scr10

Собираем программу и запускаем, и получаем верный результат 

Теперь меняем файл так,чтобы мы могли посчитать значение выражения (4\*6+2)/5

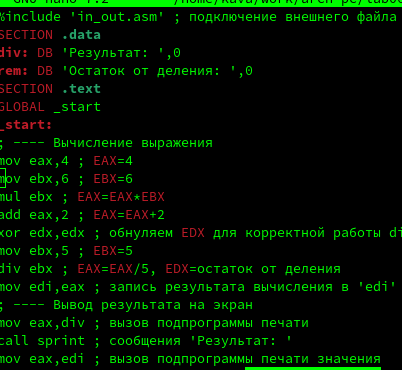


Рис. 14: scr10

Собираем программу и запускаем, и получаем верный результат

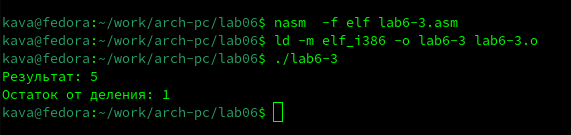


Рис. 15: scr10

Теперь создами файл variat.asm

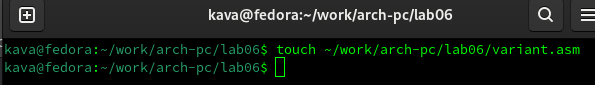


Рис. 16: scr10

И вставляем код из файла листинга 6.4

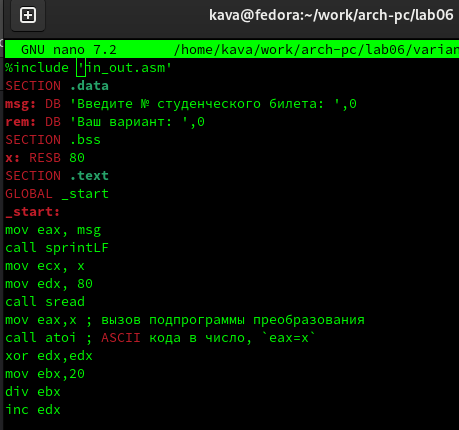


Рис. 17: scr10

Соберем и запустим ее

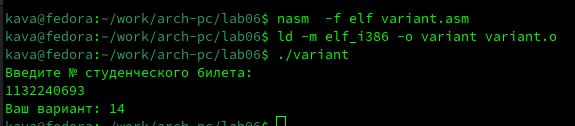


Рис. 18: scr10

И нам выведится число 14,и это действительно так

Ответим на вопросы лабораторной работы

1 Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

За это отвечает 21 строчка кода call sprint перед которой идёт строка mov eax,rem, которая перемещает строку с фразой в регистр eax ,из которого мы считаем данные для вывода

2 Для чего используется следующие инструкции?

mov ecx, x

mov edx, 80

call sread

Эти инструкции используются для того, чтобы записать данные в переменную x

3 Для чего используется инструкция “call atoi”?

Для преобразования ASCII кода в число

4 Какие строки листинга 6.4 отвечают за вычисления варианта?

div ebx

inc edx

Первая делит число x в регистре eax на значение ebx регистра , а вторая прибавляет к значению регистра edx удиницу

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

В регистр edx

6 Для чего используется инструкция “inc edx”?

Для увеличения значения регистра edx на единицу

7 Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

mov eax,edx

call iprintLF

Первая строка переносит значение регистра edx в eax, а вторая вызывает операцию вывода значения регистра eax

Задание для самостоятельной работы

Я написал программу,которая вычисляет пример под номером 10

Предворительно, я создал файл под именем task10.asm и написал следующий код

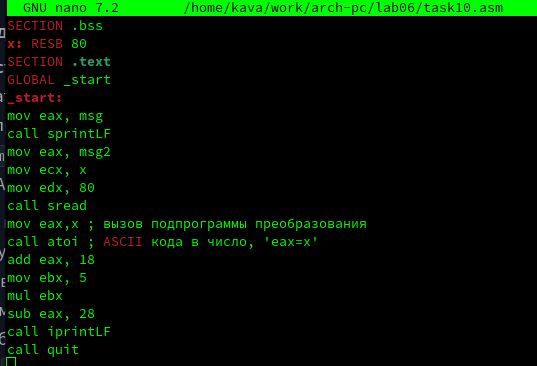


Рис. 19: scr11

И запустил код,в качестве x я указал число 5

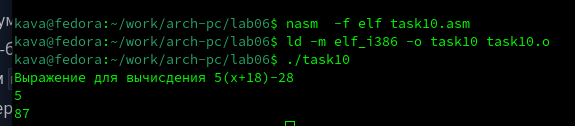


Рис. 20: scr12

Как видим, программа работает исправна и правильно вычисляет выражения.

# 3 Выводы

В результате выполнения лабораторной работы, я освоил арифметические операции которые есть в Ассемблере и как они работают. Здесь кратко описываются итоги проделанной работы.