Отчет по Лабораторной работе №6

Архитектура компьютеров и операционные системы

Бекауов Артур Тимурович НКАбд-01-23

Содержание

# 1 Цель работы

Целью работы является освоение арифметических инструкций языка ассемблера NASM.

# 2 Ход лабораторной работы

**Символьные и численные данные в NASM**

Создаю каталог для программ ЛО№6, перехожу в него cкопирую из папки для ЛО№5 файл “in\_out.asm” и создаю текстовый файл “lab6-1.asm” (Рис. [1](#fig:1)). Открываю файл в mcedit, и ввожу в него текст листинга 6.1 (Рис. [2](#fig:2))

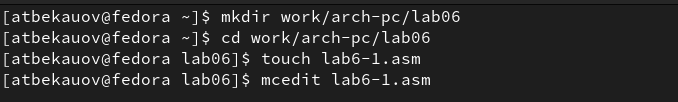


Figure 1: Создание папки для ЛО№6 и файла “lab6-1.asm”

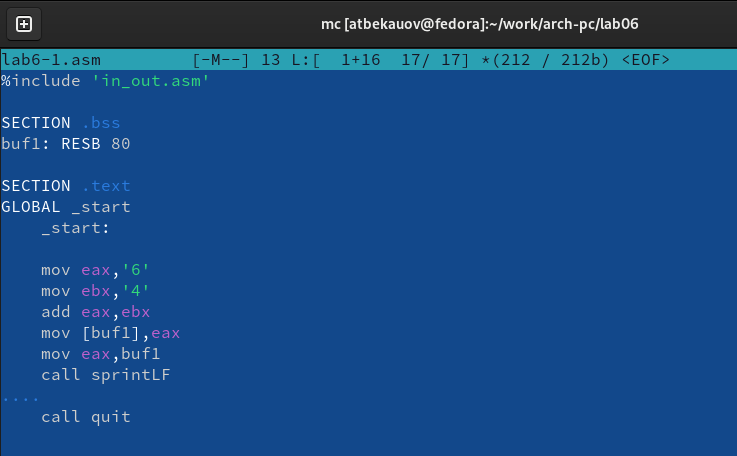


Figure 2: Текст программы “lab6-1.asm”

Создаю исполняемый файл “lab6-1” и запускаю его (Рис. [3](#fig:3))

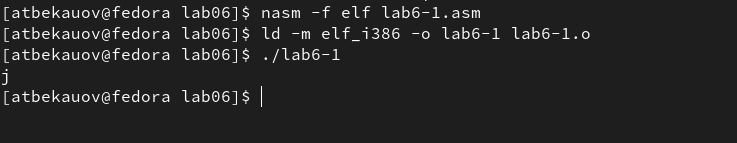


Figure 3: Первый вывод файла “lab6-1”

В качестве ответа, вопреки ожиданиям, на экране появилась буква j, а не число 10. Это происходит потому, что программа посчитала сумму кодов символов 4 и 6 в двоичном представлении и вывела символ j, код которого соответствует этой сумме.

Далее изменю текст программы и вместо символов запишу в регистры числа (Рис. [4](#fig:4)).

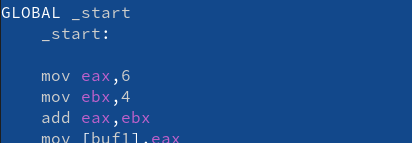


Figure 4: Исправленная часть “lab6-1.asm”

Создаю опять исполняемый файл “lab6-1” и запускаю его (Рис. [5](#fig:5))

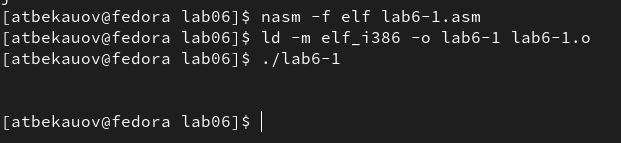


Figure 5: Второй вывод файла “lab6-1”

Как видно, число 10 опять не было выведено на экран. Вместо этого выведен символ с кодом 10, пользуясь таблицей ASCII, вижу что десятичному коду 10 соответствует “line feed” - символ переносящий на новую строку (Рис. [6](#fig:6)). Символ как таковой не видно, но его последствия мы видим - на месте вывода пустая строка.

Figure 6: 10-ый символ таблицы ASCII”

Figure 6: 10-ый символ таблицы ASCII”

Создаю файл “lab6-2.asm” в папке ЛО№6, открываю его в mcedit, и ввожу в него текст листинга 6.2 (Рис. [7](#fig:7)).

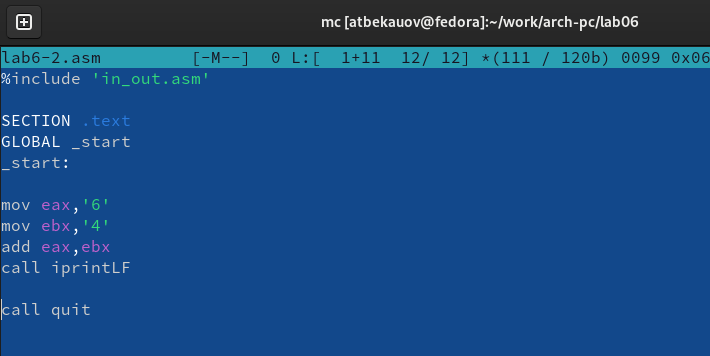


Figure 7: Текст программы “lab6-2.asm”

Создам исполняемый файл “lab6-2” и запущу его (Рис. [8](#fig:8))

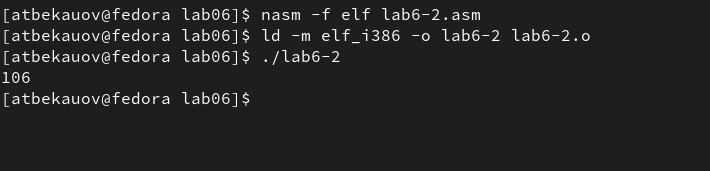


Figure 8: Первый вывод файла “lab6-2”

В результате работы программы мы получили число 106. В данном случае, как и в первом, команда add складывает коды символов ‘6’ и ‘4’ (54+52=106). Однако, в отличии от программы “lab6-1”, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число.

Аналогично предыдущей программе, изменим в тексте программы “lab6-2.asm” символы на числа (Рис. [9](#fig:9))

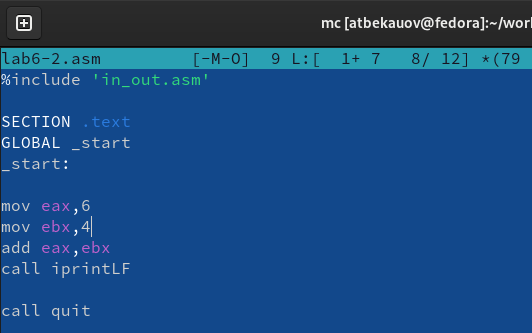


Figure 9: Отредактированный текст “lab6-2.asm”

Создаю опять исполняемый файл “lab6-2” и запускаю его (Рис. [10](#fig:10))

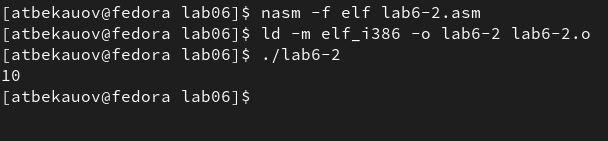


Figure 10: Второй вывод файла “lab6-2”

В этот раз всё получилось, как я и хотел и вывелось 10 - сумма чисел 4 и 6. Заменю функцию iprintLF на iprint в тексте программы “lab6-2.asm” (Рис. [11](#fig:11)) и сравню получившиеся результаты программ “lab6-2.asm” (Рис. [12](#fig:12))



Figure 11: Исправленная часть “lab6-2.asm”

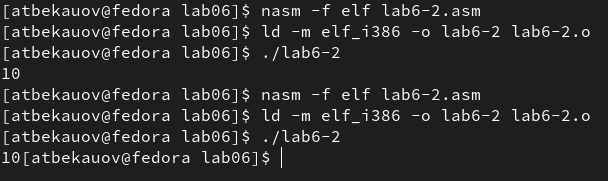


Figure 12: Сравнение второго и третьего выводов “lab6-2”

Как можно увидеть, разница в выводе двух программ заключается в том, что после команды iprintLF совершается переход на новую строчку, а во-втором случае - после iprint переход не совершается, из-за чего вывод программы “lab6-2.asm” оказался на одной строчке с вызовом новой команды терминалом.

**Выполнение арифметических операций в NASM**

В качестве примера выполнения арифметических действий создам программу вычисления арифметического выражения f(x)=(5\*2+3)/3. Для этого создаю файл “lab6-3.asm” в папке ЛО№6, открываю его в mcedit, и ввожу в него текст листинга 6.3 (Рис. [13](#fig:13)).

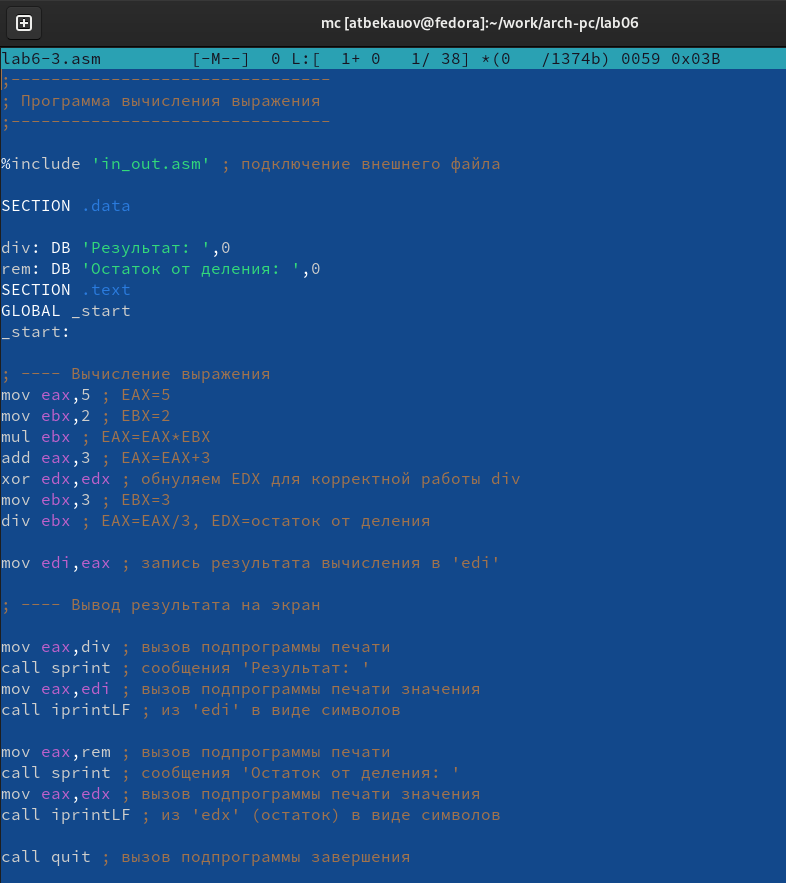


Figure 13: Текст программы “lab6-3.asm”

Создам исполняемый файл “lab6-3” и запущу его (Рис. [14](#fig:14))

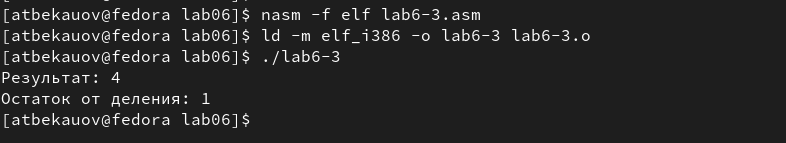


Figure 14: Первый вывод файла “lab6-3”

Вывод файла совпал с результатами предоставленными в методичке, также правильность его легко проверить, просто посчитав значение выражения.

Исправлю текст программы “lab6-3.asm” так, чтобы он вычислял выражение f(x)=(4\*6+2)/5 (Рис. [15](#fig:15)).

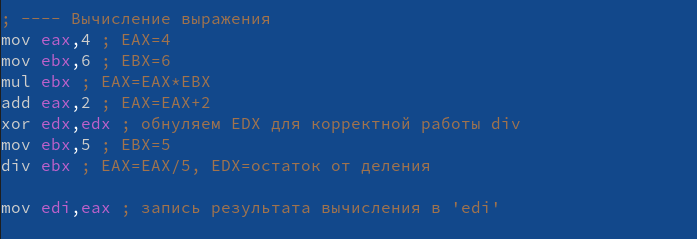


Figure 15: Исправленная часть “lab6-3.asm”

Создаю опять исполняемый файл “lab6-2” и запускаю его (Рис. [16](#fig:16))

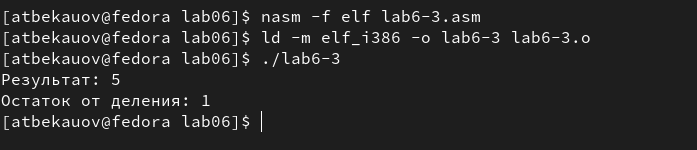


Figure 16: Второй вывод файла “lab6-3”

Вывод файла совпадает с моими рассчетами.

В качестве другого примера создам программу вычисления варианта задания по номеру студенческого билета. Для этого создаю файл “variant.asm” в папке ЛО№6, открываю его в mcedit, и ввожу в него текст листинга 6.4 (Рис. [17](#fig:17)).

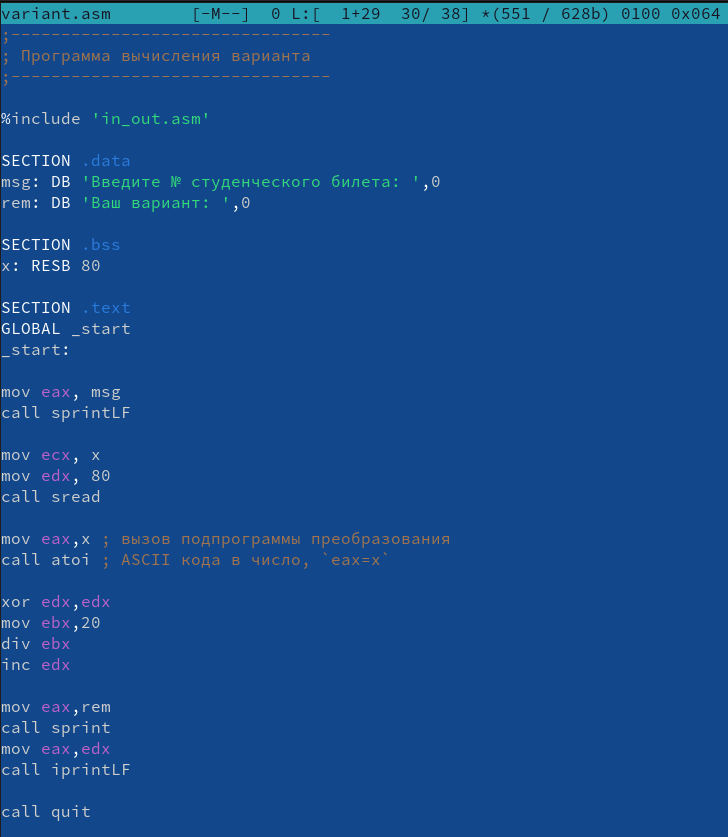


Figure 17: Текст программы “variant.asm”

Создам исполняемый файл “variant” и запущу его (Рис. [18](#fig:18))

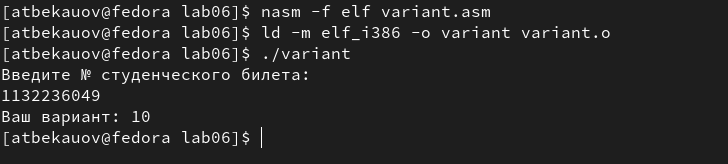


Figure 18: Вывод файла “variant”

Вывод программы совпадает с моими вычислениями (1132236049 mod 20 = 9, 9 + 1 = 10). Мой вариант 10.

**Вопросы к ходу лабораторной работы:**

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

mov eax,rem call sprint

1. Для чего используется следующие инструкции? mov ecx, x mov edx, 80 call sread -Эти инструкции запрашивают ввод (студенческого билета)
2. Для чего используется инструкция “call atoi”? -Для преобразования символьного текста введенного с клавиатуры (номера студенческого билета) в число.
3. Какие строки листинга 6.4 отвечают за вычисления варианта?

* mov ebx,20 div ebx inc edx

1. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? ebx”? -В регистр eax.
2. Для чего используется инструкция “inc edx”? -Для того,чтобы увеличить значение регистра edx на 1.
3. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

* mov eax,edx call iprintLF

# 3 Ход самостоятельной работы

Самостоятельное задание для моего варианта заключается в том, чтобы создать файл, получающий с клавиатуры значение x и вычисляющий значение выражения f(x)=5\*(x+18)-28, а затем проверить результат выполнения программы на x1=2, x2=3. Для этого создаю файл “lab6-4.asm” в папке ЛО№6, открываю его в mcedit, и ввожу в него следующий текст программы (Рис. [19](#fig:19)).

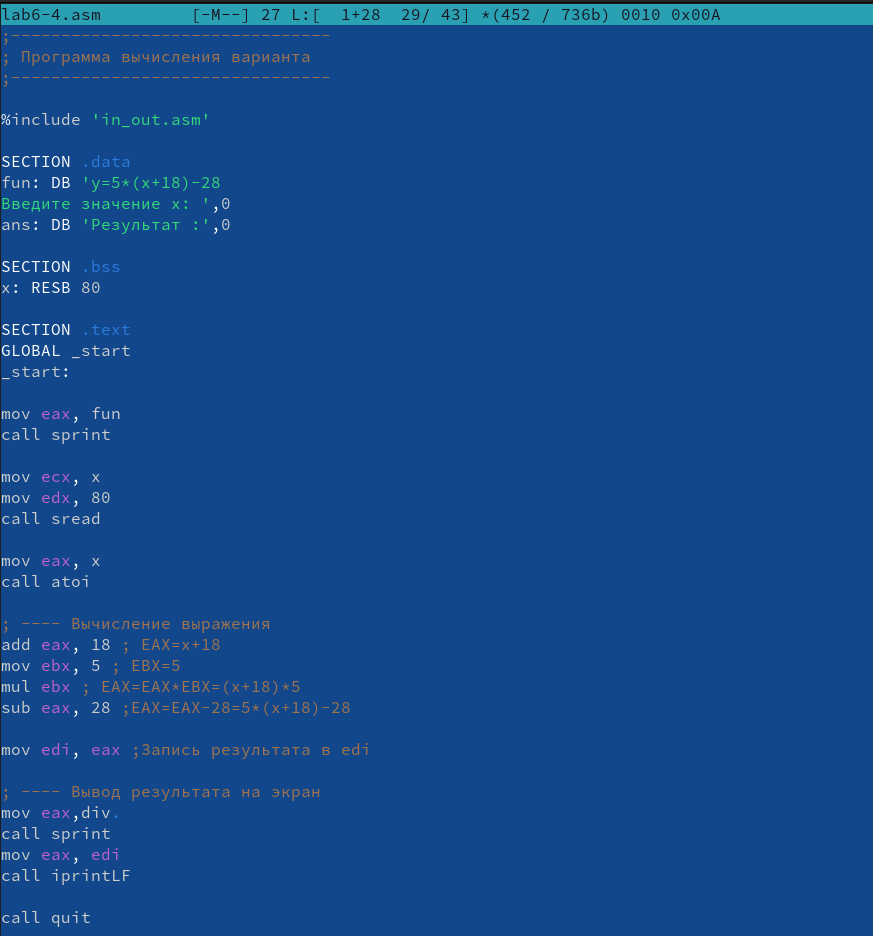


Figure 19: Текст программы “lab6-4.asm”

Создам исполняемый файл “lab6-4” и запущу его дважды, вводя значения x1 и x2(Рис. [20](#fig:20))

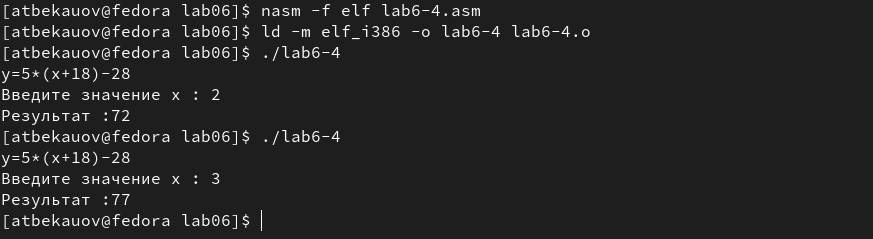


Figure 20: Выводы программы “lab6-4”

# 4 Выводы

В ходе лабораторной работы я освоил арифметические инструкций языка ассемблера NASM.