Отчёта по лабораторной работе 6

Освоение арифметических инструкций языка ассемблера NASM.

Галиева Аделина Руслановна

Содержание

# 1 Цель работы

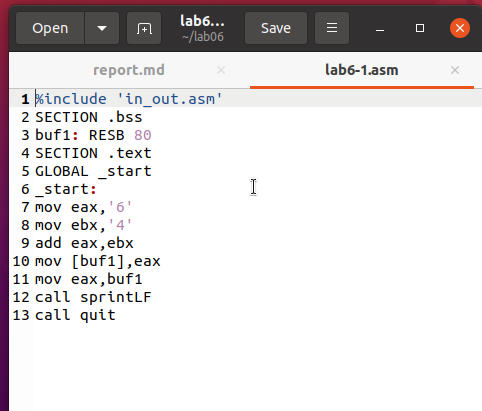
Освоение арифметических инструкций языка ассемблера NASM.

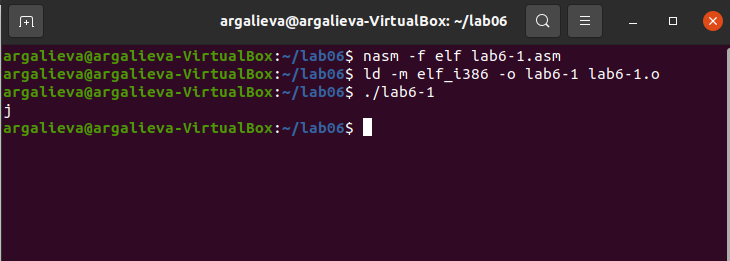
# 2 Задание

1. Изучить примеры программ.
2. Написать программу вычисления выражения в соответсвии с вариантом.
3. Загрузить файлы на GitHub.

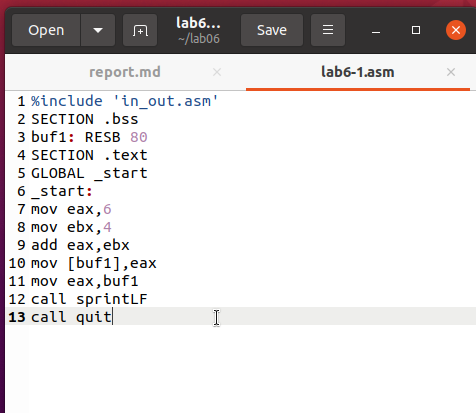
# 3 Выполнение лабораторной работы

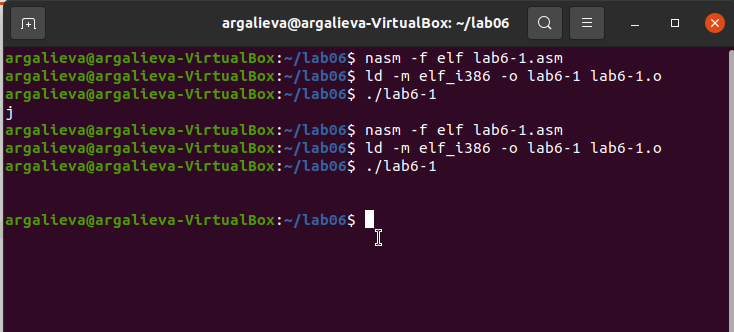
1. Создаём каталог для программам лабораторной работы № 6, переходим в него и создаем файл lab7-1.asm:
2. Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения, записанные в регистр eax.





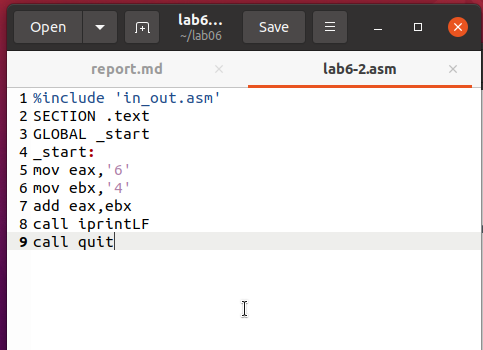
1. Далее меняем текст программы и вместо символов, запишем в регистры числа. Исправляем текст программы (Листинг 1) следующим образом

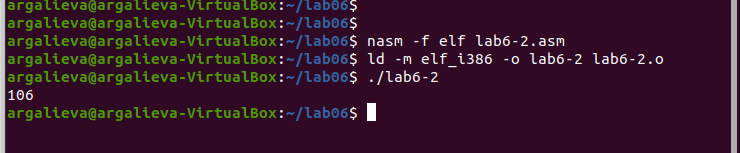




Никакой символ не виден, но он есть. Это возврат каретки LF.

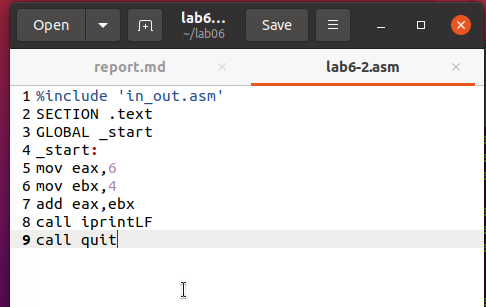
1. Как отмечалось выше,для работы с числами в файле in\_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из Листинга 7.1 с использованием этих функций.

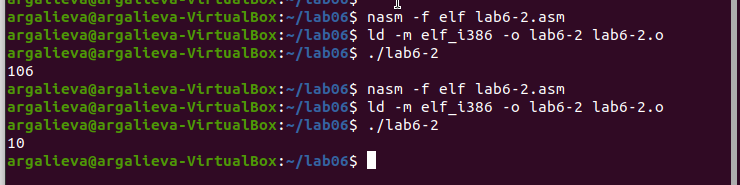




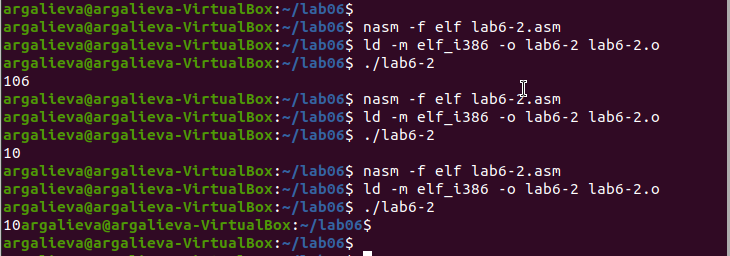
В результате работы программы мы получим число 106. В данном случае, как и в первом, команда add складывает коды символов ‘6’ и ‘4’ (54+52=106). Однако, в отличии от программы из листинга 7.1, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число.

1. Аналогично предыдущему примеру изменим символы на числа. Создаём исполняемый файл и запускаем его. Какой результат будет получен при исполнении программы? – получим число 10



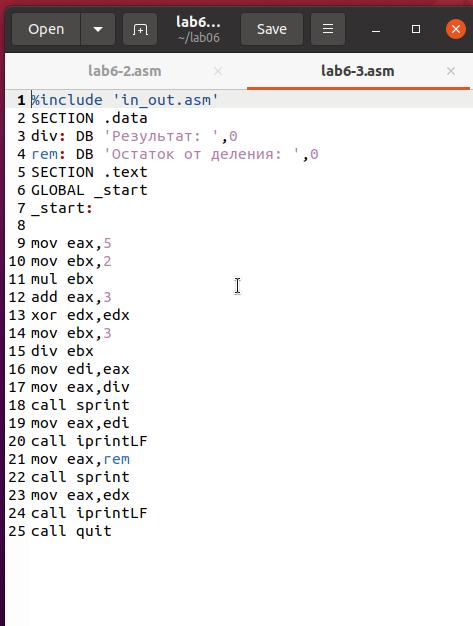


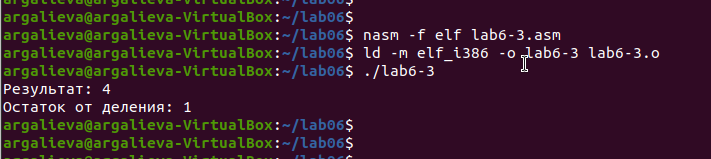
Заменяем функцию iprintLF на iprint. Создаем исполняемый файл и запускаем его. Чем отличается вывод функций iprintLF и iprint? - Вывод отличается что нет переноса строки.



1. В качестве примера выполнения арифметических операций в NASM приводим программу вычисления арифметического выражения

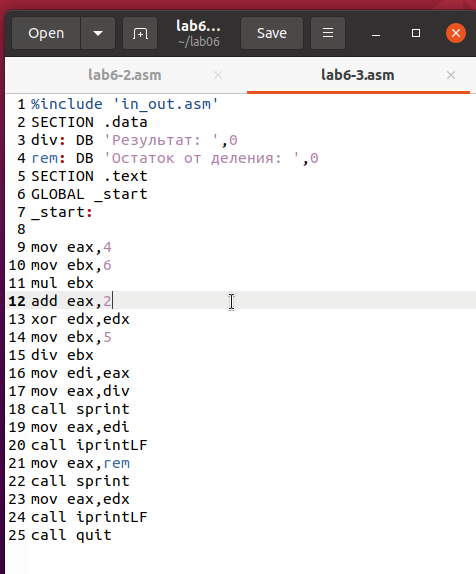
* .

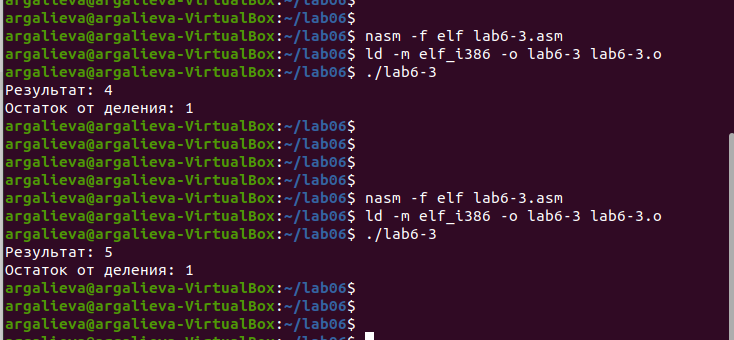




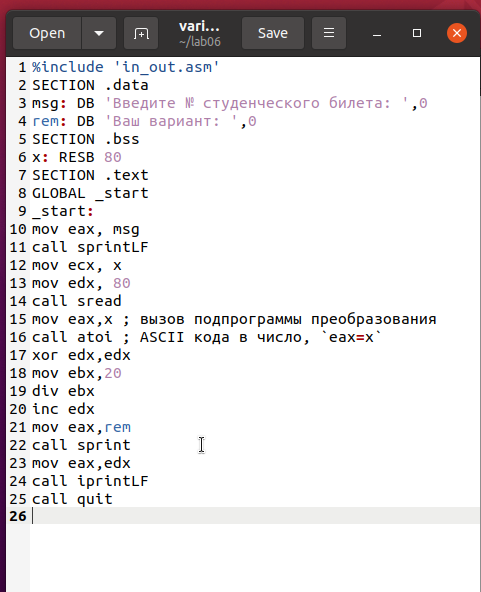
Изменияем текст программы для вычисления выражения

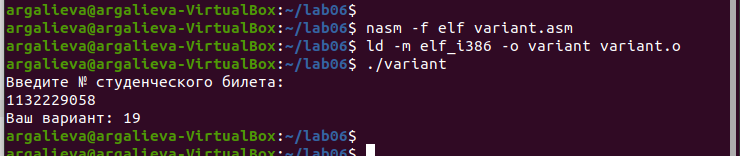
. Создаем исполняемый файл и проверяем его работу.





1. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму:





* Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? – mov eax,rem – перекладывает в регистр значение переменной с фразой ‘Ваш вариант:’ call sprint – вызов подпрограммы вывода строки
* Для чего используется следующие инструкции? nasm mov ecx, x mov edx, 80 call sread

Считывает значение студбилета в переменную Х из консоли

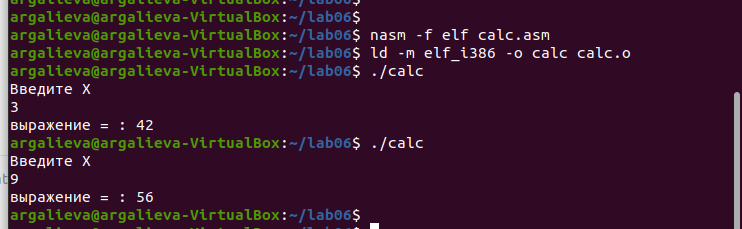
* Для чего используется инструкция “call atoi”? - эта подпрограмма переводит введенные символы в числовой формат
* Какие строки листинга 7.4 отвечают за вычисления варианта? xor edx,edx mov ebx,20 div ebx
* В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? 1 байт AH 2 байта DX 4 байта EDX – наш случай
* Для чего используется инструкция “inc edx”? по формуле вычисления варианта нужно прибавить единицу
* Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений mov eax,edx – результат перекладывается в регистр eax call iprintLF – вызов подпрограммы вывода

1. Написать 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) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создаем исполняемый файл и проверяем его работу для значений x1 и x2 из 6.3.

Вариант 19 -

для х=3 и 9

!{](image/16.png){ #fig:016 width=70%, height=70% }



# 4 Выводы

Я изучила работу с арифметическими операциями.