Лабораторная работа №6

Арифметические операции в NASM

Колонтырский Илья Русланович

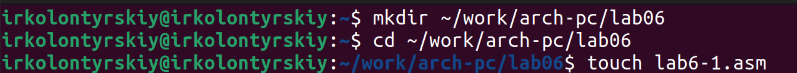
Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

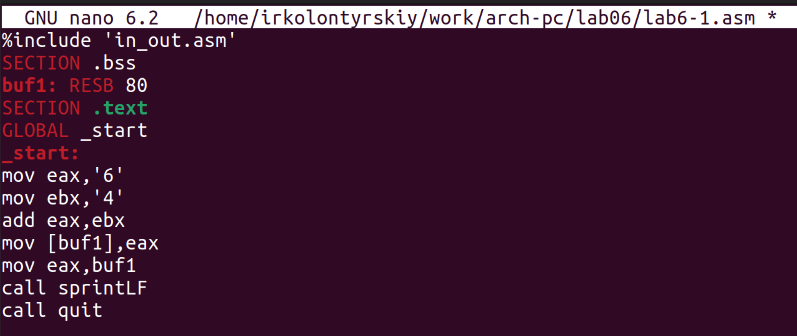
# 2 Выполнение лабораторной работы

Создадим папку рабочего каталога и файл lab6-1.asm (Рис. 2.1)



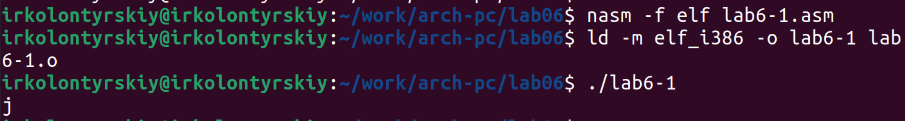
Создание файла и рабочей директории

Вставим в созданный файл код из листинга (Рис. 2.2)



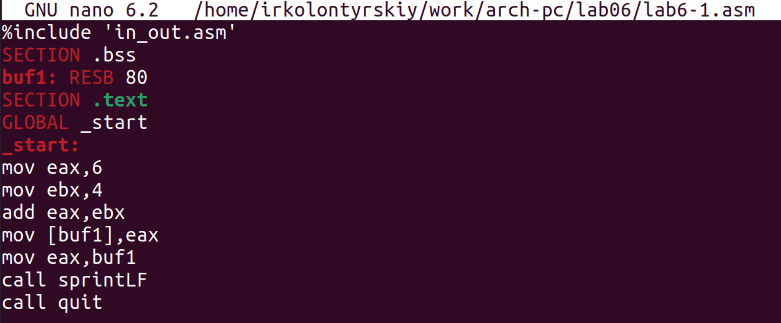
Запись кода из листинга

Теперь соберём файл и запустим его (Рис. 2.3)



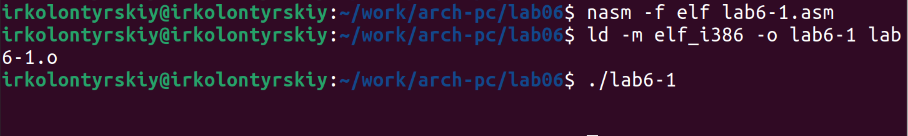
Сборка исполняемого файла и запуск

Вывело символ j. Изменим файл (Рис. 2.4)



Редактирование файла

Мы убрали кавычки и теперь складываем числа. Соберём и запустим файл (Рис. 2.5)



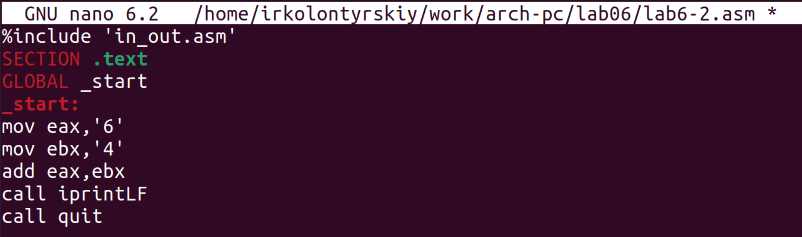
Запуск программы и сборка

Ничего не вывелось. Но на самом деле вывелся символ перевода строки, и поэтому мы его не видим. Создадим файл под названием lab6-2.asm (Рис. 2.6)

Создание файла lab6-2.asm

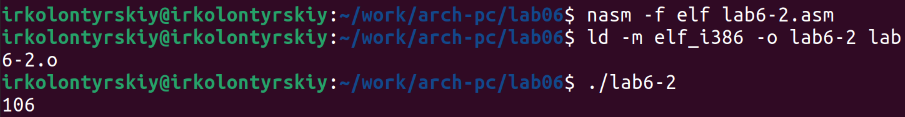
Создание файла lab6-2.asm

Вставим в него следующий код (Рис. 2.7)



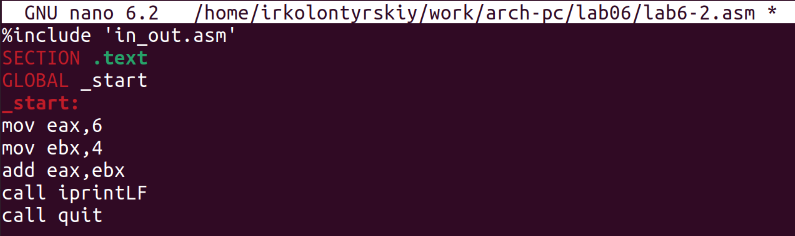
Вставка кода в файл lab6-2.asm

Вместо sprintLF тут iprintLF. Соберём файл и запустим (Рис. 2.8)



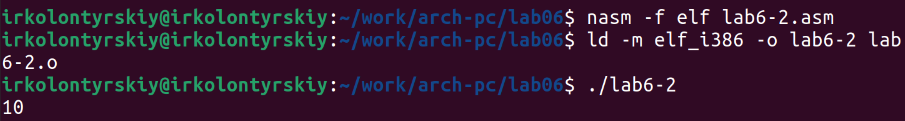
Запуск файла и результат

Вывело число 106. Уберём кавычки у чисел в коде (Рис. 2.9)



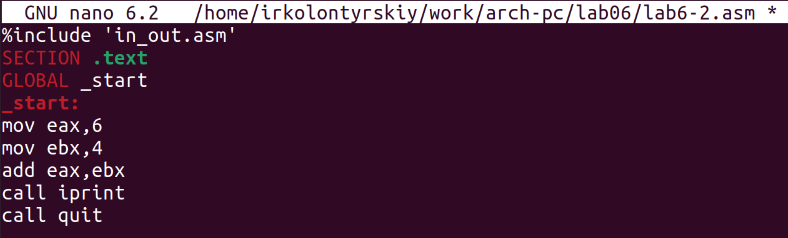
Изменение файла

Соберём и запустим программу (Рис. 2.10)



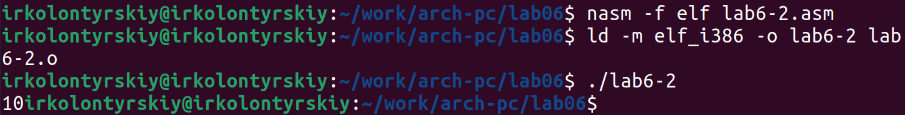
Сборка файла и результат

Вывод корректный. Заменим iprintLF на iprint (Рис. 2.11)



Редактирование файла

Соберём и запустим (Рис. 2.12)



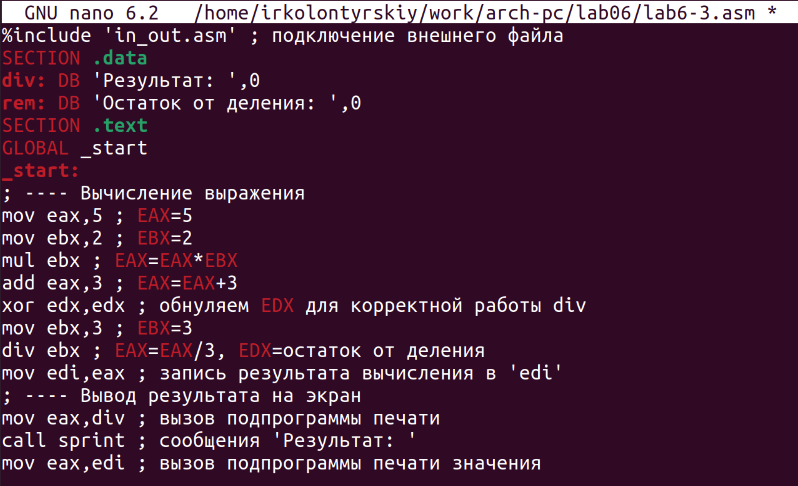
Сборка и результат работы отредактированного файла

iprint не использует перенос на новую строку. Создадим ещё 1 файл (Рис. 2.13)

Создание файла lab6-3.asm

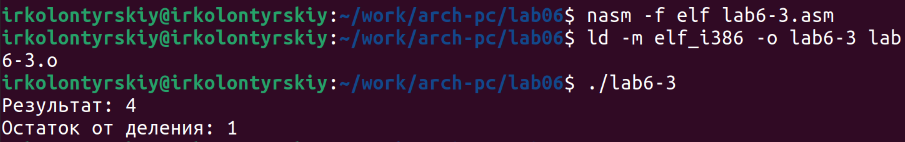
Создание файла lab6-3.asm

Он считает выражение (5\*2+3)/3. Вставим в него следующий код (Рис. 2.14)



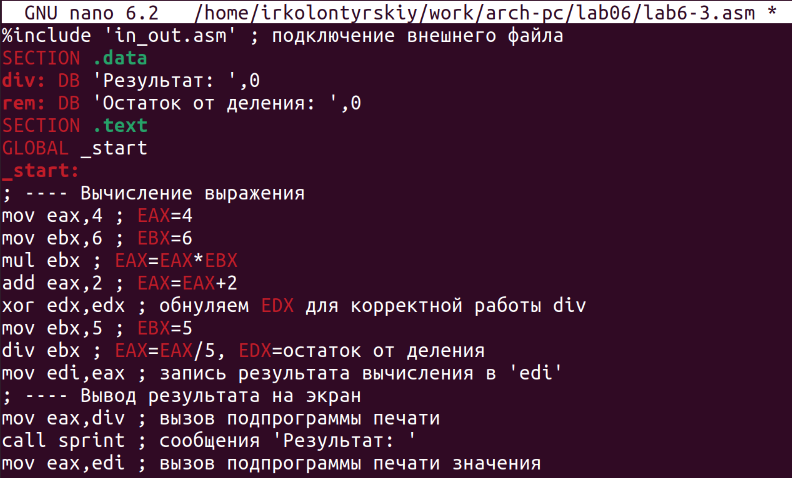
Вставка кода в файл

Попробуем запустить этот программу, предварительно её собрав (Рис. 2.15)



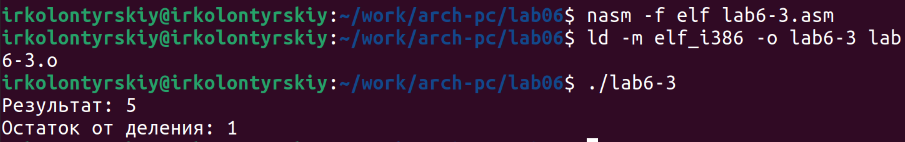
Сборка и запуск

Вычисления верные. Сделаем так, чтобы код вычислял функцию f(x)=(4\*6+2)/5. Внесём следующие изменения (Рис. 2.16)



Изменение файла

Соберём и запустим (Рис. 2.17)



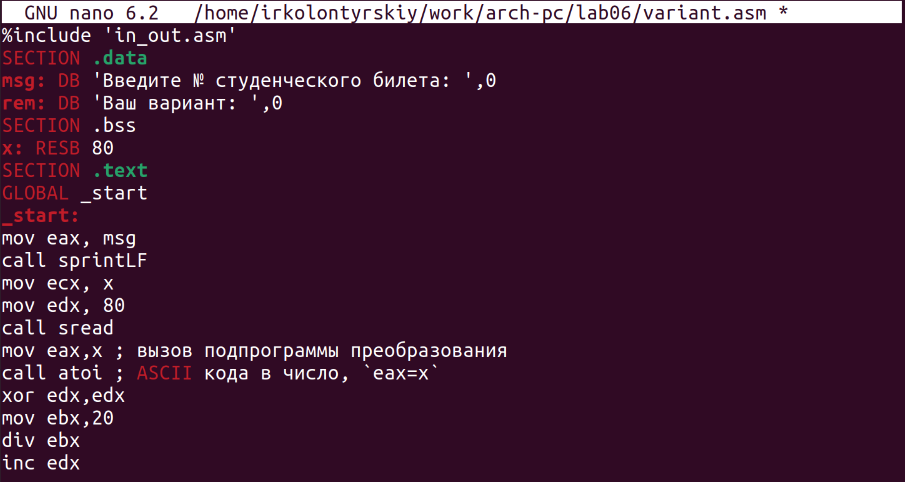
Повторная сборка и запуск

Ответ верный. Создадим файл variant.asm. Он должен вычислять номер варианта по студенческому билету (Рис. 2.18)

Создание файла variant.asm

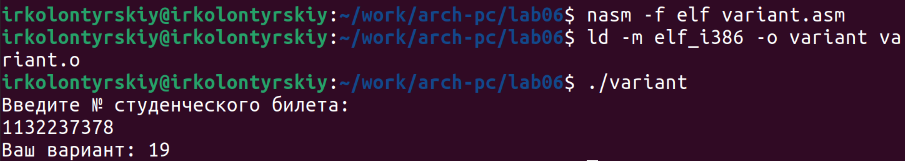
Создание файла variant.asm

Вставим в него следующий код (Рис. 2.19)



Вставка кода

Скомпилируем и запустим файл. Укажем в вводе студенческий билет (Рис. 2.20):



Компиляция и запуск

Пересчитав вручную, находим, что это верный ответ

Ответим на вопросы:

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

Строчка под номером 25

2. Для чего используется следующие инструкции?

mov ecx, x

mov edx, 80

call sread

Они записывают ввод с клавиатуры в переменную х

3. Для чего используется инструкция “call atoi”?

Для преобразования кода символа в число

4. Какие строки листинга 6.4 отвечают за вычисления варианта?

div ebx

inc edx

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

В edx

6. Для чего используется инструкция “inc edx”?

Прибавляет к edx единицу

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

mov eax,edx

call iprintLF

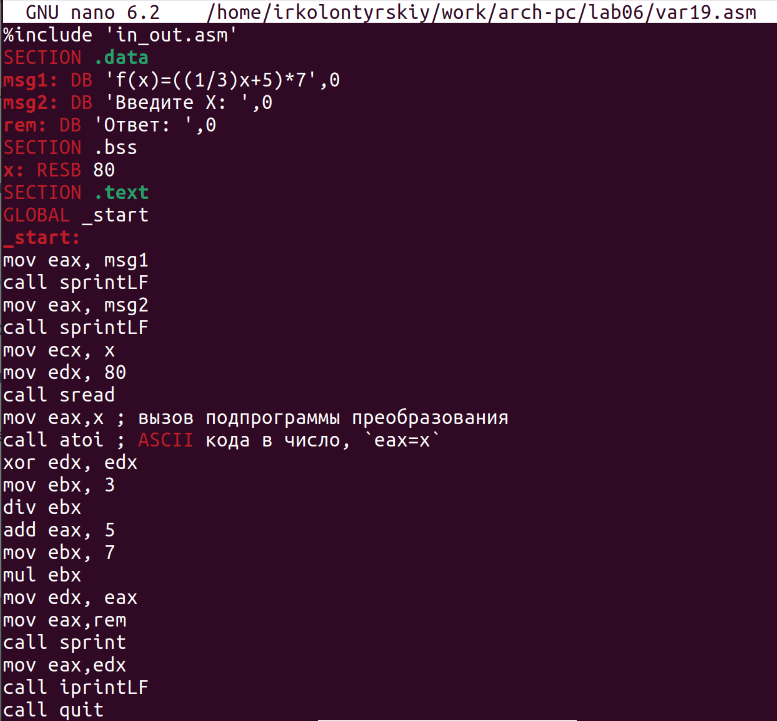
Самостоятельная работа

Создадим файл самостоятельной работы (Рис. 2.21)

Создание файла

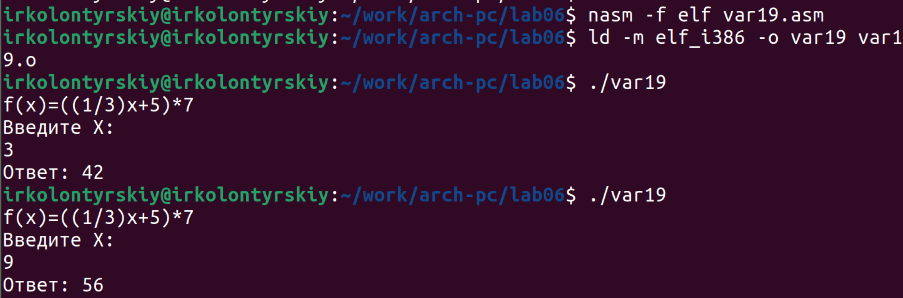
Создание файла

Теперь напишем следующий код (Рис. 2.22)



Код программы

Проверим работу программы, собрав и запустив её (Рис. 2.23)



Запуск программы и проверка её корректной работы

Программа дала верные ответы

# 3 Выводы

Были получены навыки работы с арифметическими инструкциями языка ассемблера