Арифметические операции NASM

Дисциплина: Архитектура компьютера

Швед Карина Дмитриевна

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

# 2 Ход лабораторной

Я создала каталог для программ лабораторной работы № 6, перешла в него и создала файл lab6-1.asm: mkdir ~/work/arch-pc/lab06 cd ~/work/arch-pc/lab06 touch lab6-1.asm (рис. 1).

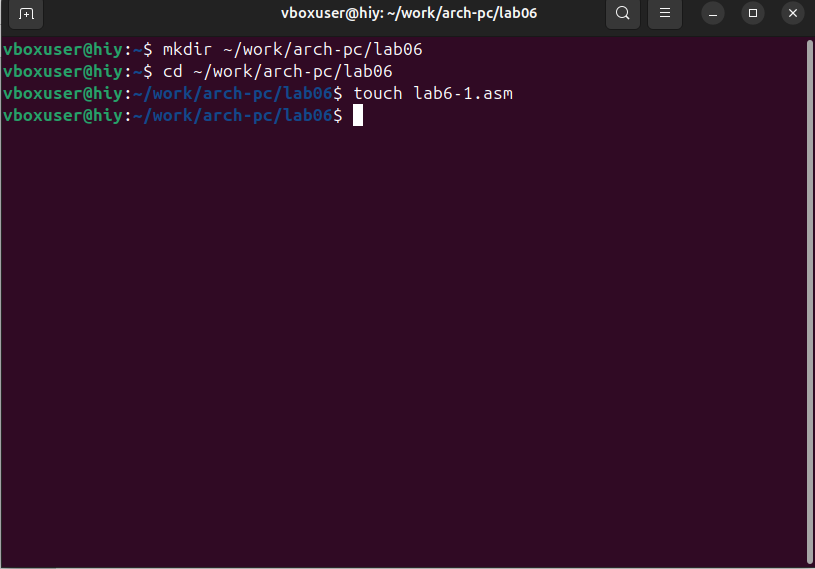


Рис. 1: Создание каталога для лабораторной 6 и файла asm

Далее я ввела в файл lab6-1.asm текст программы (рис. 2)из листинга 6.1. Перед созданием исполнямого файла я создала копию файла in\_out.asm в каталоге ~/work/arch-pc/lab06.Далее cоздала исполняемый файл и запустила его.(рис. 3)

На экране я увидела символ j, так как код символа 6 равен 00110110 вдвоичном представлении, а код символа 4 – . Команда add eax,ebx запишет в регистр eax сумму кодов – 01101010,что в свою очередь является кодом символа j согласно таблице ASCII.

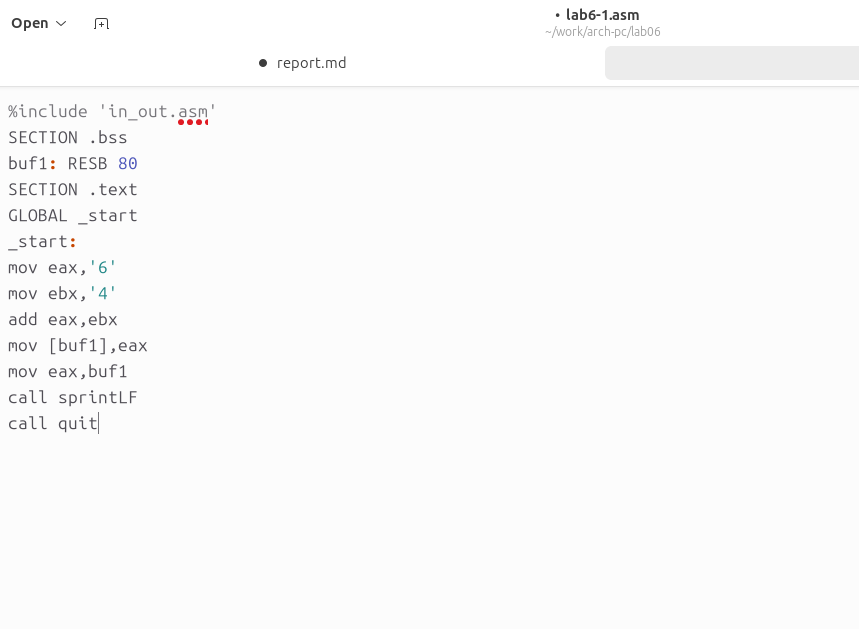


Рис. 2: код программы lab6-1.asm

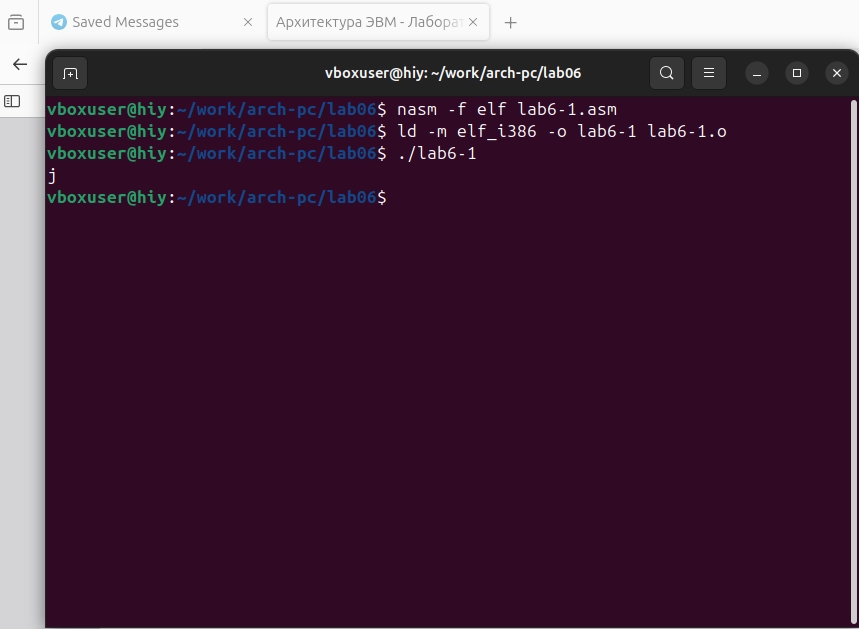


Рис. 3: создание исполняемого файла lab6-1.asm

Далее я изменила текст программы и вместо символов записала в регистры числа. Я заменила строки:

mov eax,‘6’ mov ebx,‘4’ на строки mov eax,6 mov ebx,4

Создала исполняемый файл и запустила его (рис. 4). Как и в предыдущем случае, при выполнении программы я не получила число 10. Вместо этого выводится символ с кодом 10, который представляет собой символ конца строки (возврат каретки). Этот символ не отображается на экране, но он добавляет пустую строку.

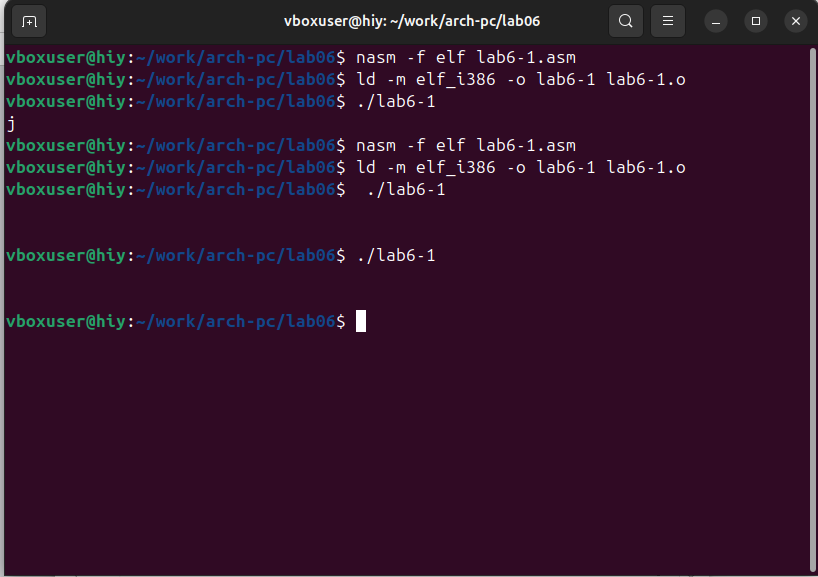


Рис. 4: работа программы lab6-1.asm

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 из Листинга 6.1 с использованием этих функций. Создала файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввела в него текст программы из листинга 6.2 (рис. 5), создала исполняемый файл и запустила его (рис. 6).

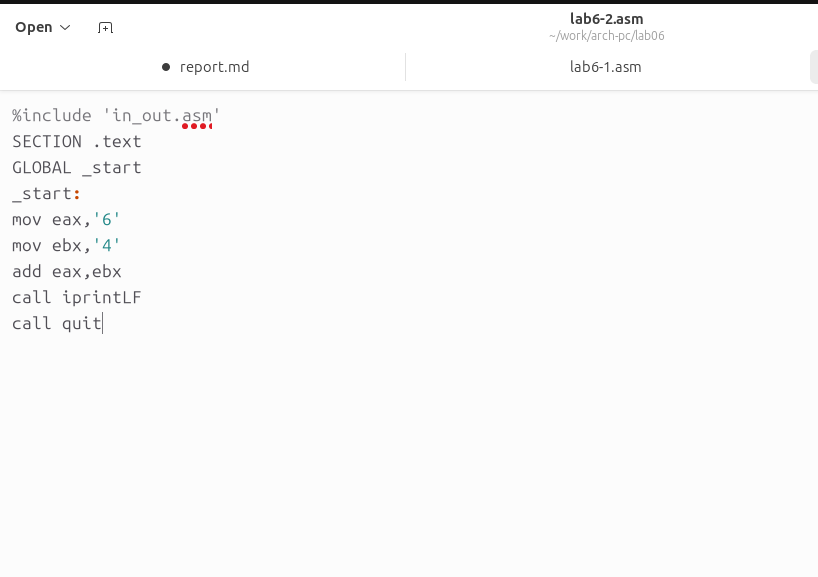


Рис. 5: код программы lab6-2.asm

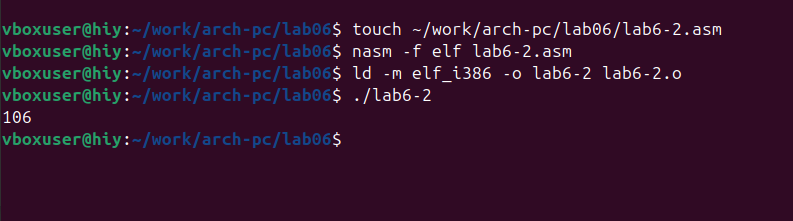


Рис. 6: запуск программы lab6-2.asm

В результате работы программы я получила 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 6.1, функция iprintLF позволяет вывести число, а не символ, кодом которого является это число.

Аналогично предыдущему примеру я изменила символы на числа. Я заменила строки (рис. 7).

mov eax,‘6’ mov ebx,‘4’ на строки mov eax,6 mov ebx,4

Создала исполняемый файл и запустила его (рис. 8). Функция iprintLF позволяет вывести число и операндами были числа (а не коды символов). Поэтому получаем число 10.

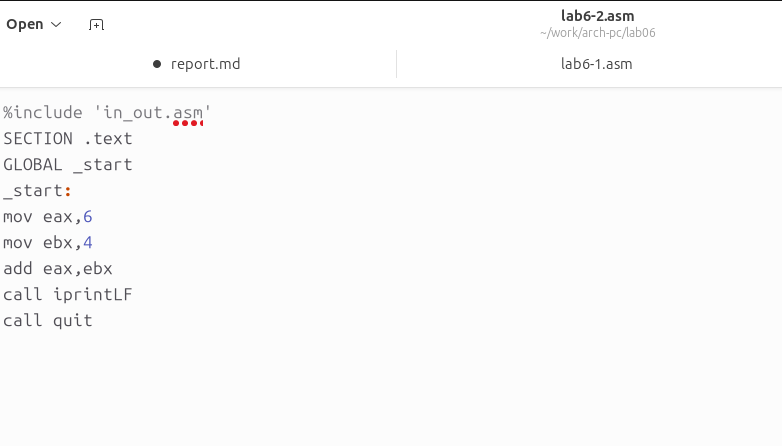


Рис. 7: программа в файле lab6-2.asm

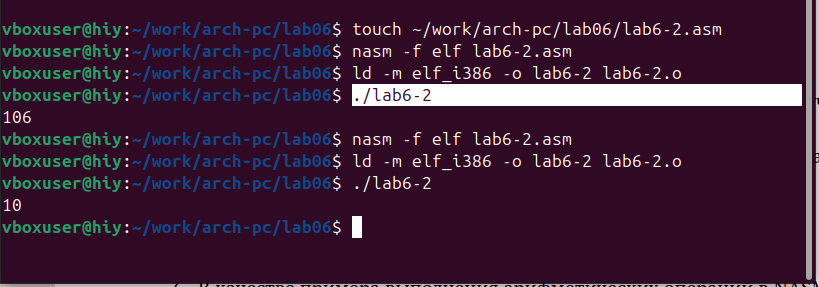


Рис. 8: работа программы lab6-2.asm

Я заменила функцию iprintLF на iprint. Создала исполняемый файл и запустила его (рис. 9). Вывод отличается тем, что нет переноса строки

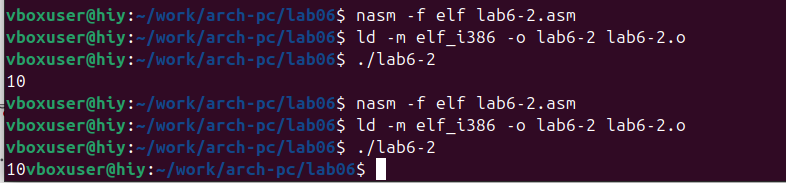


Рис. 9: работа программы lab6-2.asm

В качестве примера выполнения арифметических операций в NASM приведу программу вычисления арифметического выражения 𝑓(𝑥) = (5 ∗ 2 + 3)/3. Я создала файл lab6-3.asm в каталоге ~/work/arch-pc/lab06: touch ~/work/arch-pc/lab06/lab6-3.asm Внимательно изучила текст программы из листинга 6.3 и ввела в lab6-3.asm. Создала исполняемый файл и запустила его. Результат работы программы получился следующим (рис. 10).

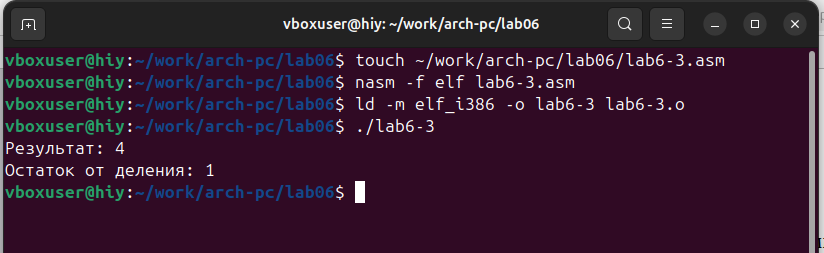


Рис. 10: работа программы lab6-3.asm

Далее я изменила текст программы (рис. 11)для вычисления выражения f (x) = (4 ∗ 6 + 2)/5. Создала исполняемый файл и проверила его работу ( (рис. 12)

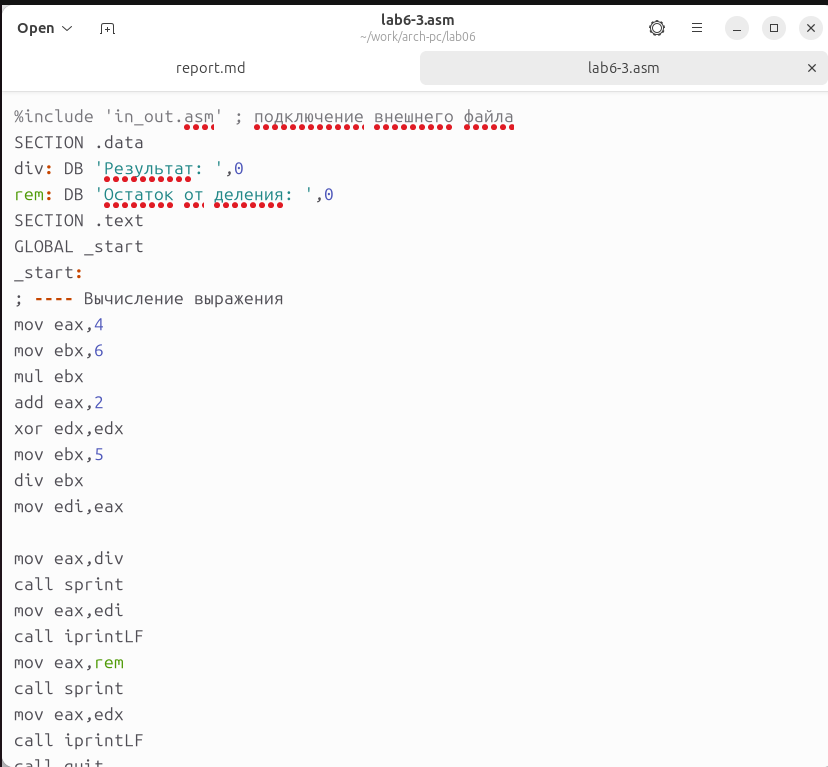


Рис. 11: код программы lab6-3.asm

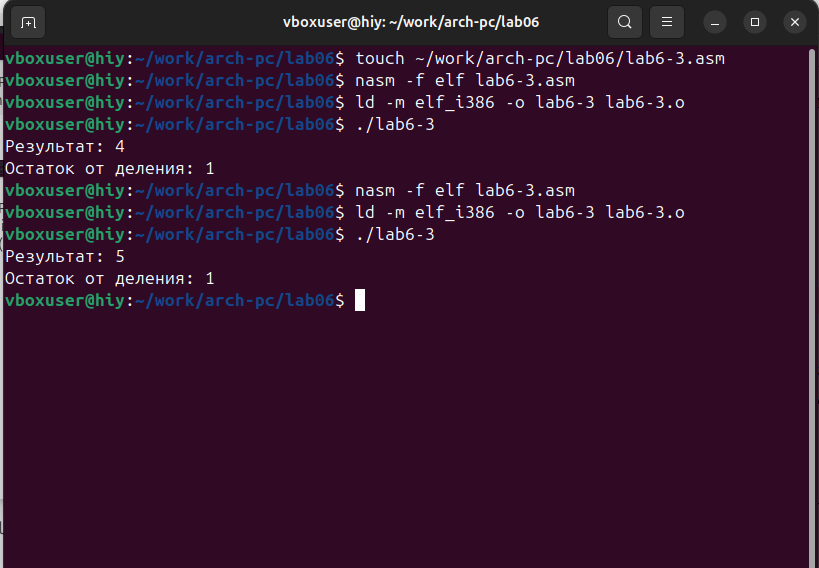


Рис. 12: работа программы lab6-3.asm

В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета.В данном случае число, над которым необходимо проводить арифметические операции,вводится с клавиатуры. 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.

Я создала файл variant.asm в каталоге ~/work/arch-pc/lab06: touch ~/work/arch-pc/lab06/variant.asm Внимательно изучила текст программы из листинга 6.4 и ввела в файл variant.asm. (рис. 13). Создала исполняемый файл и запустила его. Проверила результат работы программы вычислив номер варианта аналитически (рис. 14)

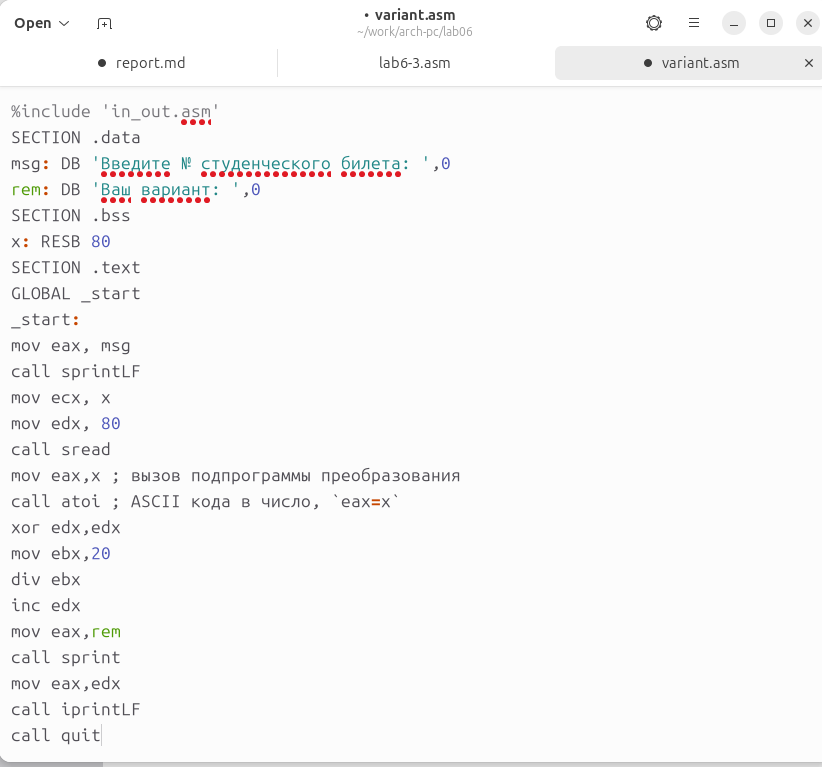


Рис. 13: код программы variant.asm

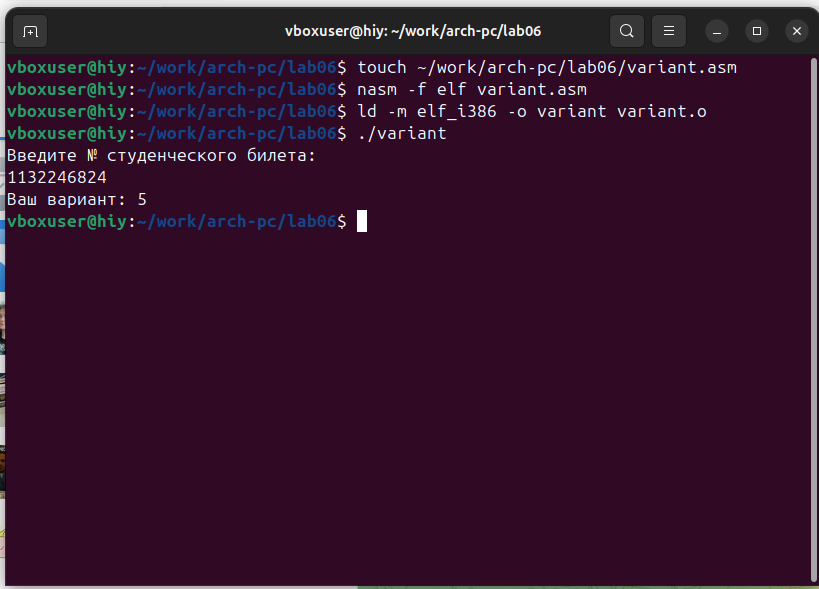


Рис. 14: работа программы variant.asm

# 3 Ответы на вопросы

1. Какие строки листинга 6.4 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
2. Для чего используются следующие инструкции? mov ecx, x Загружает адрес переменной x в регистр ECX. mov edx, 80 Загружает значение 80 в регистр EDX. call sread Вызывает подпрограмму для считывания значения студенческого билета из консоли
3. Для чего используется инструкция “call atoi”? Инструкция “call atoi” используется для преобразования введенных символов в числовой формат
4. Какие строки листинга 6.4 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
5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? Остаток от деления записывается в регистр edx
6. Для чего используется инструкция “inc edx”? Инструкция “inc edx” используется для увеличения значения в регистре edx на 1, в соответствии с формулой вычисления варианта
7. 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

# 4 Задание для самостоятельной работы

Формулировка задания: 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. Создайте исполняемый файл и проверьте его работу для значений x1 и x2 из 6.3

Я получила вариант 5:

(9𝑥 − 8)/8

для х1=8 х2=64

Сначала я создала отдельный файл task.asm и ввела код программы (рис. 15), затем получила исполняемый файл и проверила работу программы,введя значения x1 и x2 (рис. 16)

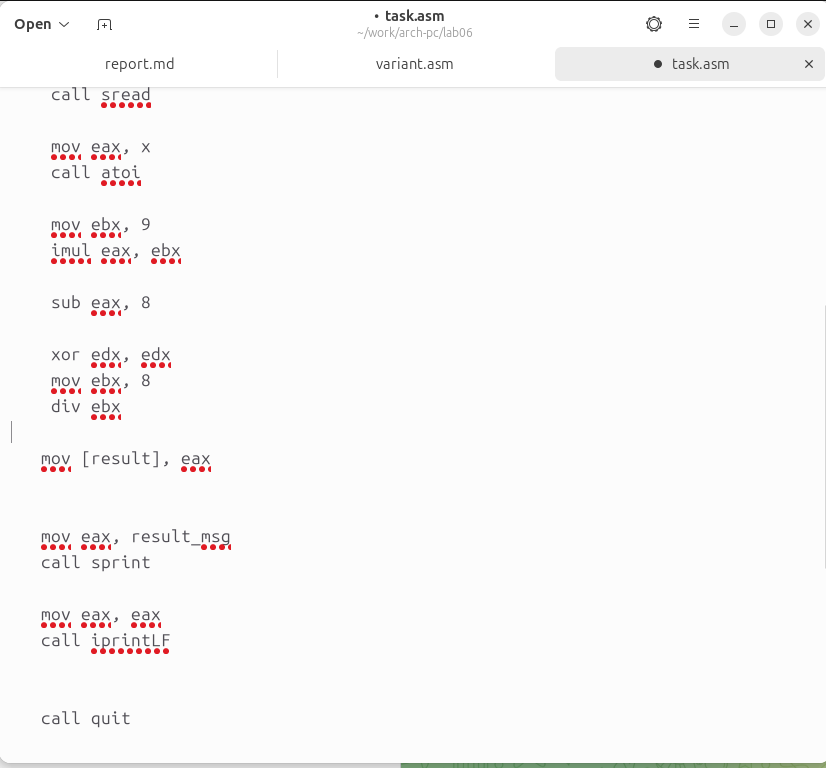


Рис. 15: код программы task.asm

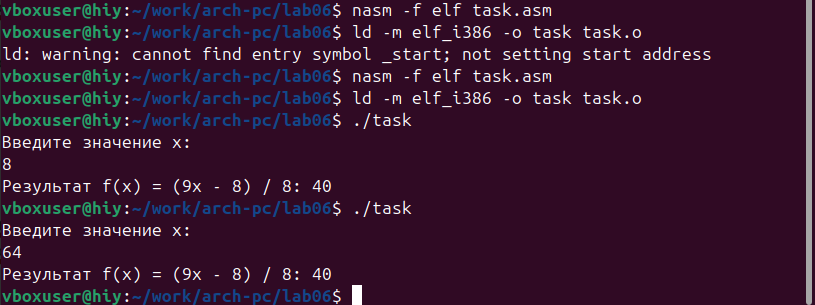


Рис. 16: работа программы task.asm

# 5 Выводы

Я освоила арифметические инструкции языка ассемблера NASM