ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №7

Лабораторная работа №7. Арифметические операции в NASM.

Кудряшов Артём Николаевич

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

Для выполнения данной лабораторнной работы необходимо освоить принцип работы с арифметическими инструкциями языка ассемблера NASM. Для этого следует разобрать различные операторы, используемые для сложения, вычитания, умножения, деления, а также перевода символьного типа в числовой.

# 3 Выполнение лабораторной работы

## 3.1 Символьные и численные данные в NASM

Откроем терминал, создадим каталог для програм в папке arch-pc и с помощью команды touch создадим первый файл lab7-1.asm (рис. 1).

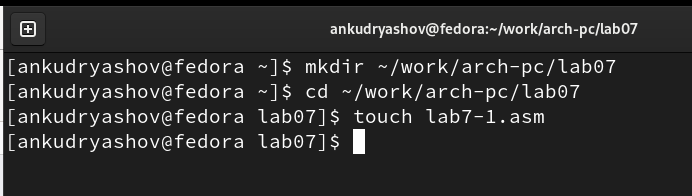


Рис. 1: Новый каталог, файл lab7-1.asm

Введём в файл lab7-1.asm текст программы из листинга 7.1 (рис. 2).

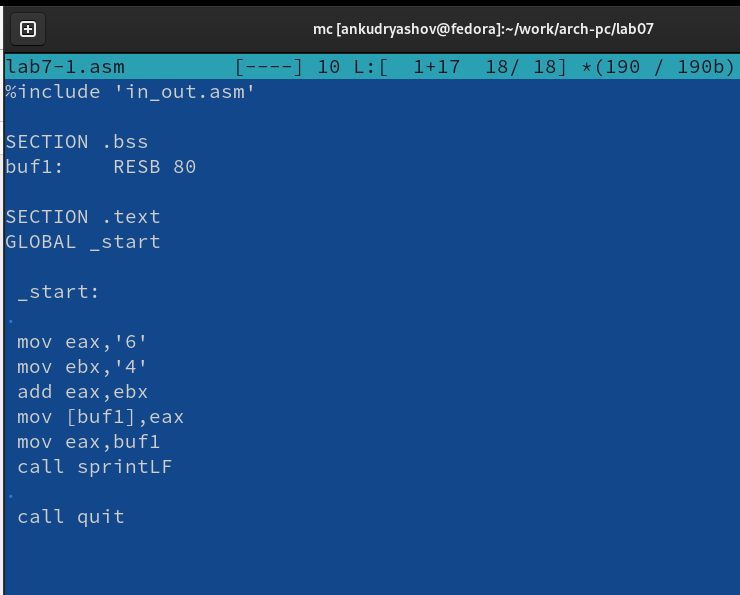


Рис. 2: Текст программы из листинга 7.1

Создадим исполняемый файл и запустим его. Как мы видим, программа вывела на экран символ ‘j’ (рис. 3).

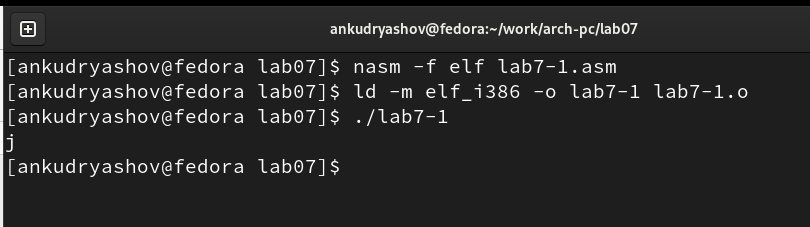


Рис. 3: Первая версия программы – вывод ‘j’

Немного изменим текст программы, записав в регистры числа, а не символы (поменяем mov eax,‘6’ на mov eax,6 и т.д.). Создадим исполняемый файл и запустим его. В этот раз на экран вывелся символ LF, т.е. перевод строки, потому что его номер в таблице символов ASCII равен 10 (т.к 6+4) (рис. 4).

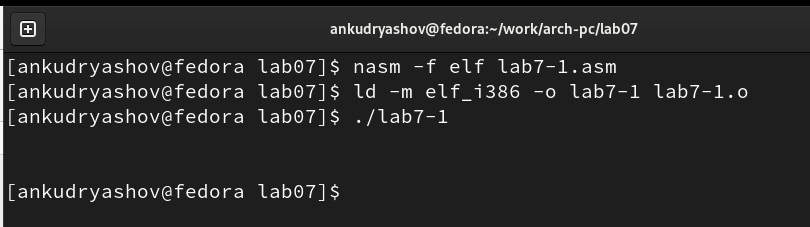


Рис. 4: Вторая версия программы – вывод перевода строки

В нашем каталоге создадим файл lab7-2.asm. Введём в него текст программы из листинга 7.2 (рис. 5).

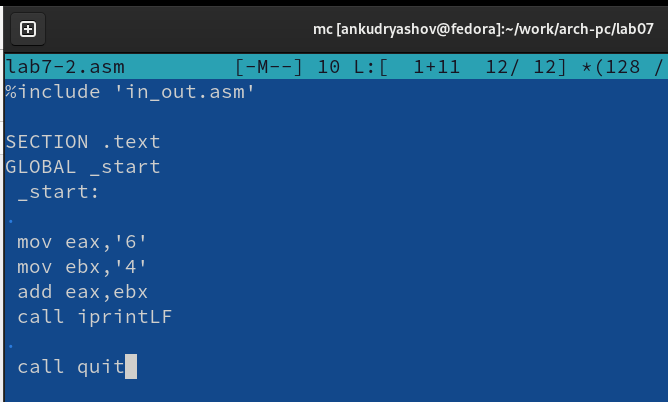


Рис. 5: Текст программы lab7-2.asm

Вновь создадим исполняемый файл и запустим его. На этот раз программа выводит на экран число 106 (сумма ASCII кодов 54 + 52) (рис. 6).

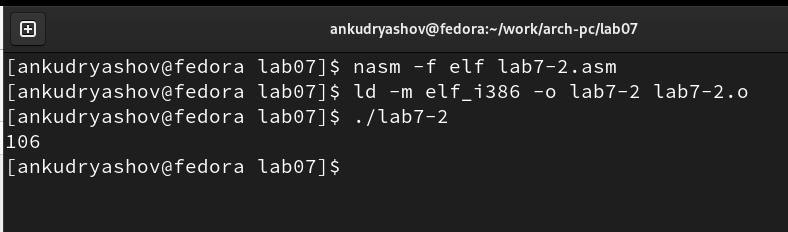


Рис. 6: Третья версия программы – вывод числа 106

Аналогично предыдущеиу примеру изменим символы на числа. В этот раз программа выведет на экран число 10, т.е. сумму чисел 6 и 4 (рис. 7).

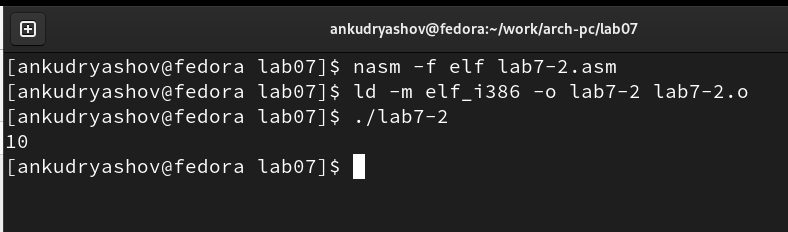


Рис. 7: Четвёртая версия программы – вывод числа 10

Заменим функцию iprintLF на iprint. Создадим и запустим исполняемый файл. Теперь программа выводит число 10, но не ставит после него перевод строки (рис. 8).

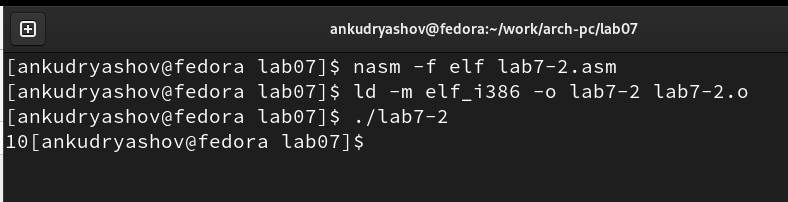


Рис. 8: Та же программа, но без перевода строки в конце

## 3.2 Выполнение арифметических операций в NASM

Вычислим значение функции f(x) = (5\*2 + 3)/3. Создадим файл lab7-3.asm и скопируем в него текст программы из листинга 7.3 (рис. 9).

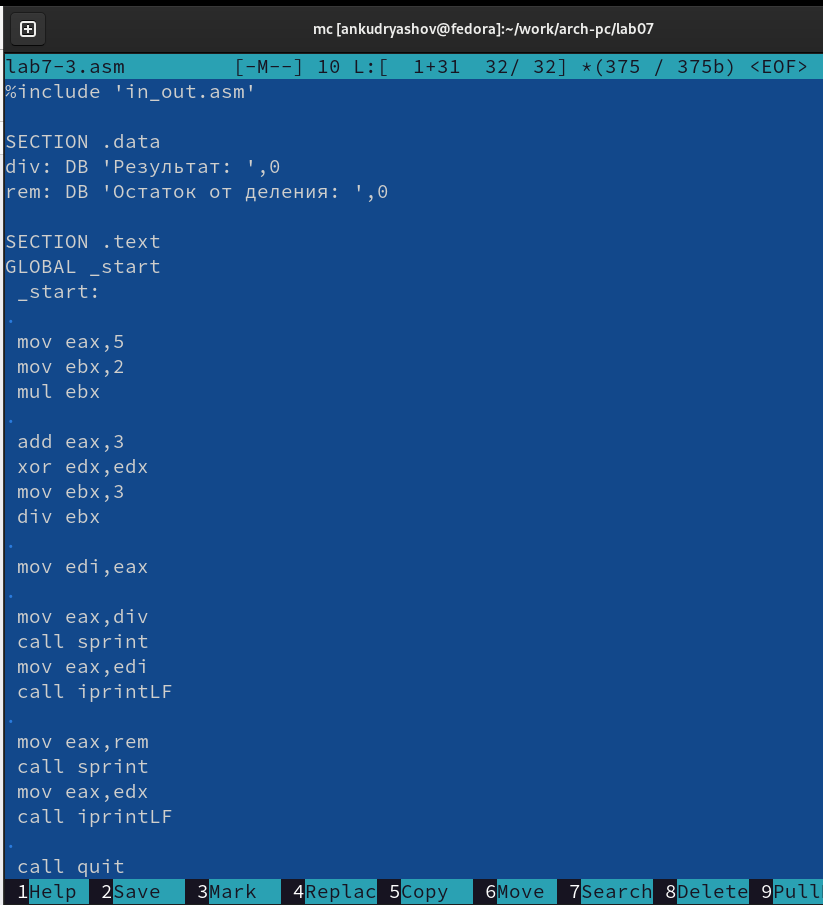


Рис. 9: Текст программы lab7-3.asm

Создадим исполняемый файл и запустим его. Увидим в выводе целую часть ответа 4 и остаток от деления 1 (рис. 10).

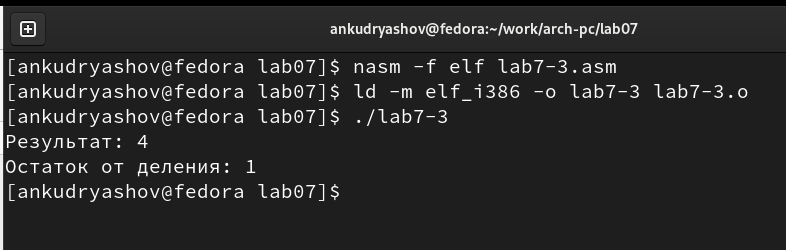


Рис. 10: Результат выполнения программы lab7-3

Изменим текст программы так, чтобы она вычисляла f(x) = (4\*6 + 2)/5 (рис. 11).

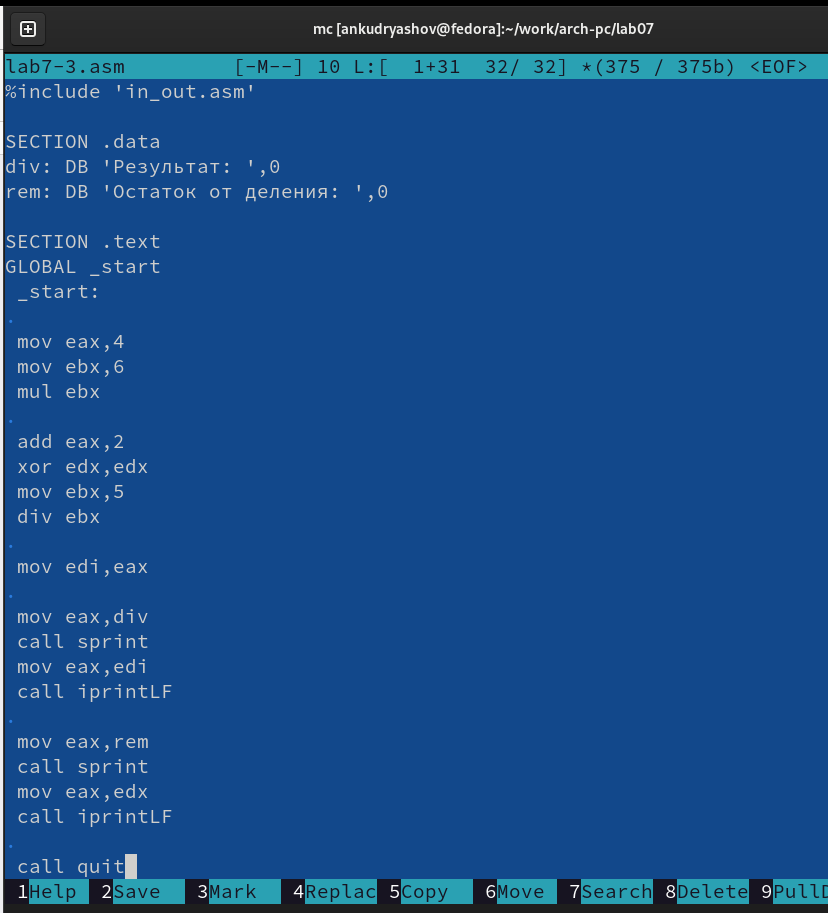


Рис. 11: Текст отредактированной программы для f(x) = (4\*6 + 2)/5

Создадим и запустим исполняемый файл (рис. 12).

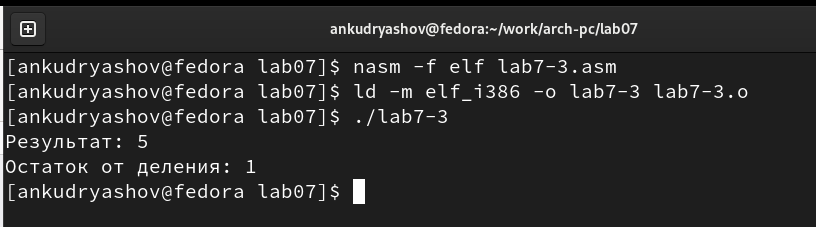


Рис. 12: Запуск отредактрованного файла lab7-3

В качестве еще одного примера создадим программу вычисления варианта задания по номеру студенческого билета. Скопируем в файл variant.asm текст программы из листинга 7.4 (рис. 13).

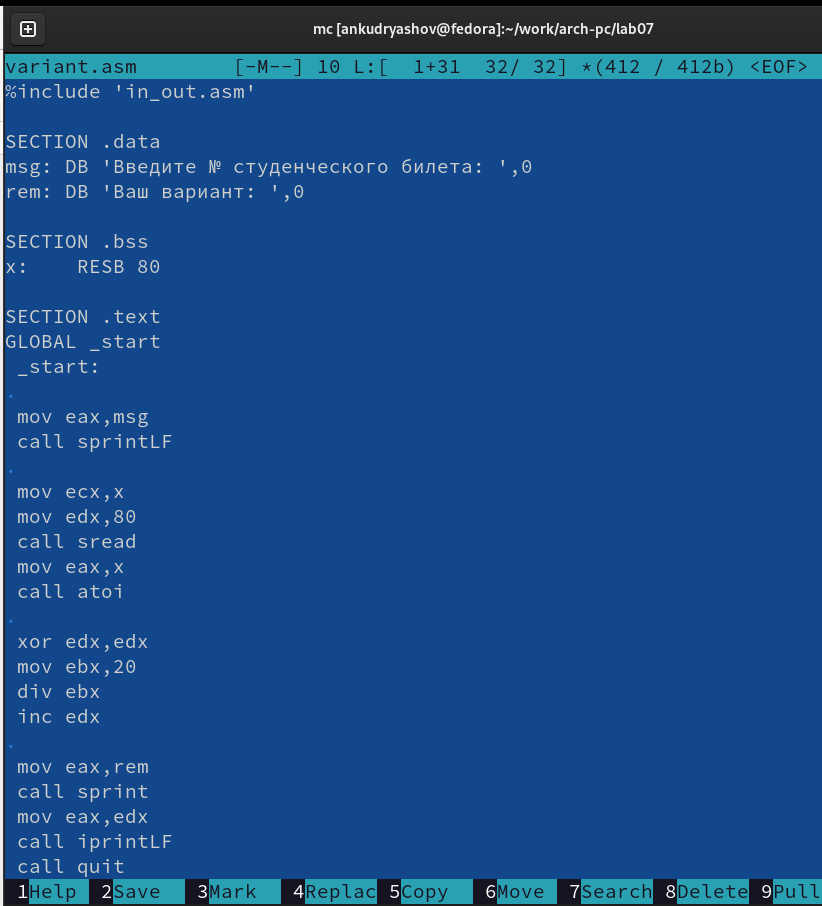


Рис. 13: Текст программы variant.asm

В результате выполнения получим вариант №14. Проверим: 1132226433 % 20 = 13, 13 + 1 = 14, верно (рис. 14).

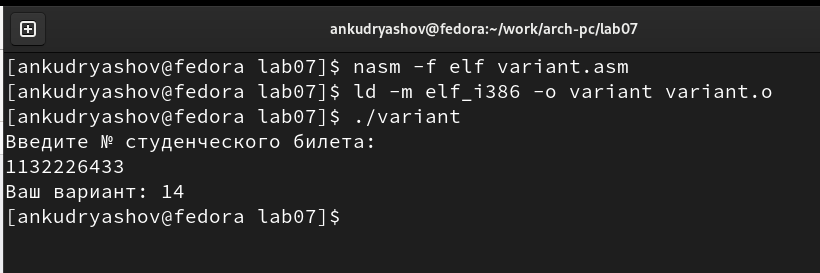


Рис. 14: Результат работы variant.asm

### 3.2.1 Ответы на вопросы

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? Переменная rem хранит текст ‘Ваш вариант:’, в строке mov eax,rem адрес rem помещается в eax, затем в строке call sprint значение из eax выводится на экран.
2. Для чего используется следующие инструкции? nasm mov ecx, x mov edx, 80 call sread Для считывания значения переменной x от пользователя. Первая инструкция указывает, что записать результат считывания нужно в x, вторая инструкция указывает максимальную длину строки, последняя инструкция вызывает функцию, считывающую значение, введённое пользователем.
3. Для чего используется инструкция “call atoi”? Для преобразования ASCII кода в число
4. Какие строки листинга 7.4 отвечают за вычисления варианта? mov ebx,20 div ebx inc edx
5. В какой регистр записывается остаток от деления при выполнении ин- струкции “div ebx”? В регистр edx.
6. Для чего используется инструкция “inc edx”? Для увеличения значения, хранящегося в edx на 1.
7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычис- лений? mov eax,rem call sprint mov eax,edx call iprintLF Здесь первые две строки выводят на экран сообщение ‘Ваш вариант:’, а вторые две строки выводят сам номер варианта, посчитанный в ходе работы программы.

## 3.3 Задание для самостоятельной работы

Получили вариант №14. Функция имеет вид f(x) = (x/2 + 8)\*3, а x1 и x2 равны 1 и 4 соответственно. Создадим файл lab7-sr.asm и запишем туда текст программы, выполняющей необходимые вычисления (рис. 15).

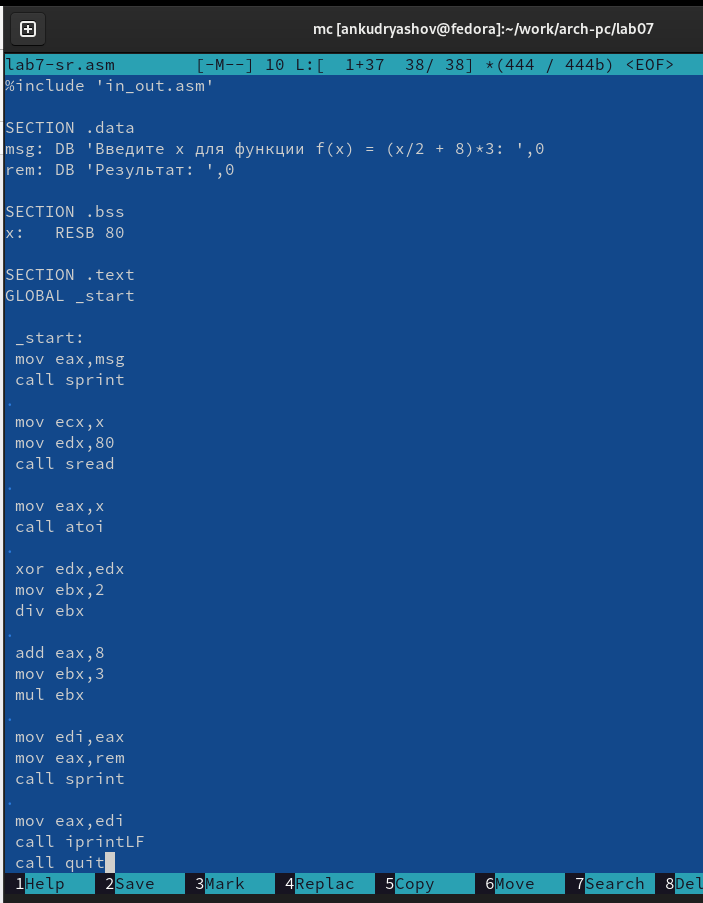


Рис. 15: Текст программы для самостоятельной работы

Создадим и запустим исполняемый файл lab7-sr. Введём значение 1. Получим (1/2 + 8)•3 = 8•3 = 24. Теперь введём значение 4. Получим (4/2 + 8)•3 = 10•3 = 30. Всё верно (рис. 16).

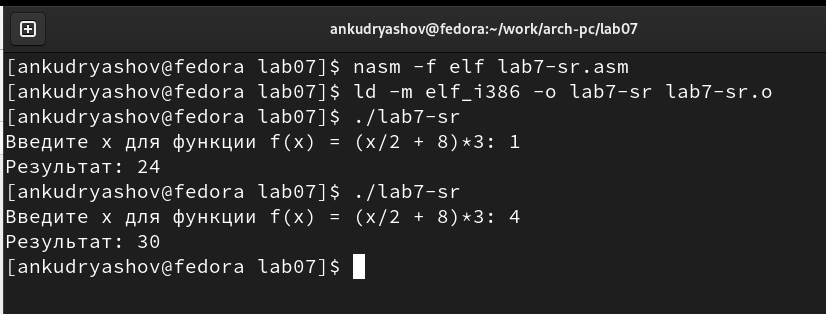


Рис. 16: Запуск файла программы для самостоятельной работы

# 4 Выводы

В результате выполнения лабораторной работы удалось в полной мере освоить работу с арифметическими операциями и переводом из символьного в числовой тип на языке ассемблера NASM.