Отчёт по лабораторной работе №6

Дисциплина: Архитектура компьютера

Буриева Шахзода Акмаловна

Содержание

# 1 Цель работы

Освоить арифметических инструкций языка ассемблера NASM.

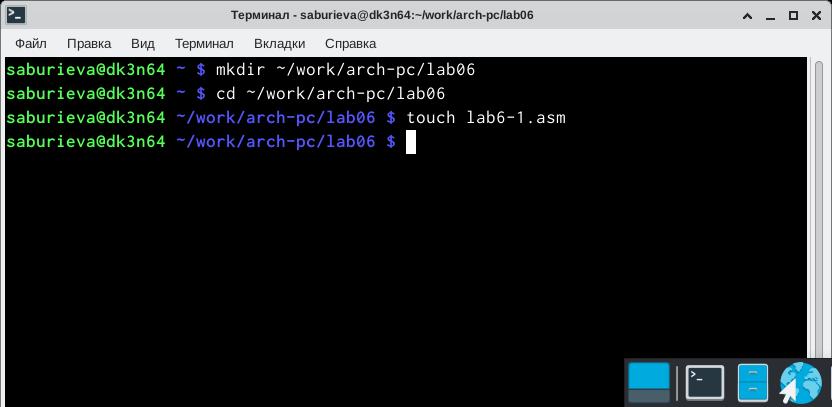
# 2 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес опе- ранда предоставляет место, где хранятся данные, подлежащие обработке. Это могут быть данные хранящиеся в регистре или в ячейке памяти. Далее рассмотрены все существующие способы задания адреса хранения операндов – способы адресации. Схема команды целочисленного сложения add (от англ. addition - добавление) выполняет сложение двух операндов и записывает результат по адресу первого операнда. Довольно часто при написании программ встречается операция прибавления или вычита- ния единицы. Прибавление единицы называется инкрементом, а вычитание — декрементом. Для этих операций существуют специальные команды: inc (от англ. increment) и dec (от англ. decrement), которые увеличивают и уменьшают на 1 свой операнд.

Ввод информации с клавиатуры и вывод её на экран осуществляется в символьном виде. Кодирование этой информации производится согласно кодовой таблице символов ASCII. ASCII – сокращение от American Standard Code for Information Interchange (Американский стандартный код для обмена информацией). Согласно стандарту ASCII каждый символ кодируется одним байтом. Расширенная таблица ASCII состоит из двух частей. Первая (символы с кодами 0-127) является универсальной (см. Приложение.), а вторая (коды 128-255) предназначена для специальных символов и букв национальных алфавитов и на компьютерах разных типов может меняться. Среди инструкций NASM нет такой, которая выводит числа (не в символьном виде). По- этому, например, чтобы вывести число, надо предварительно преобразовать его цифры в ASCII-коды этих цифр и выводить на экран эти коды, а не само число. Если же выводить число на экран непосредственно, то экран воспримет его не как число, а как последовательность ASCII-символов – каждый байт числа будет воспринят как один ASCII-символ – и выведет на экран эти символы. Аналогичная ситуация происходит и при вводе данных с клавиатуры. Введенные дан- ные будут представлять собой символы, что сделает невозможным получение корректного результата при выполнении над ними арифметических операций. Для решения этой проблемы необходимо проводить преобразование ASCII символов в числа и обратно.

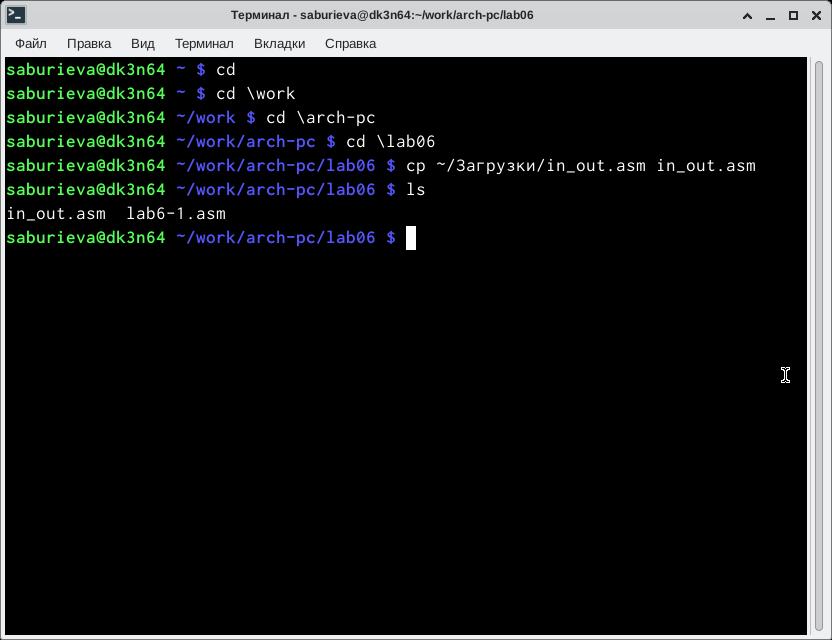
# 3 Выполнение лабораторной работы

Создала каталог для программам лабораторной работы № 6, перешла в него и создала файл lab6-1.asm



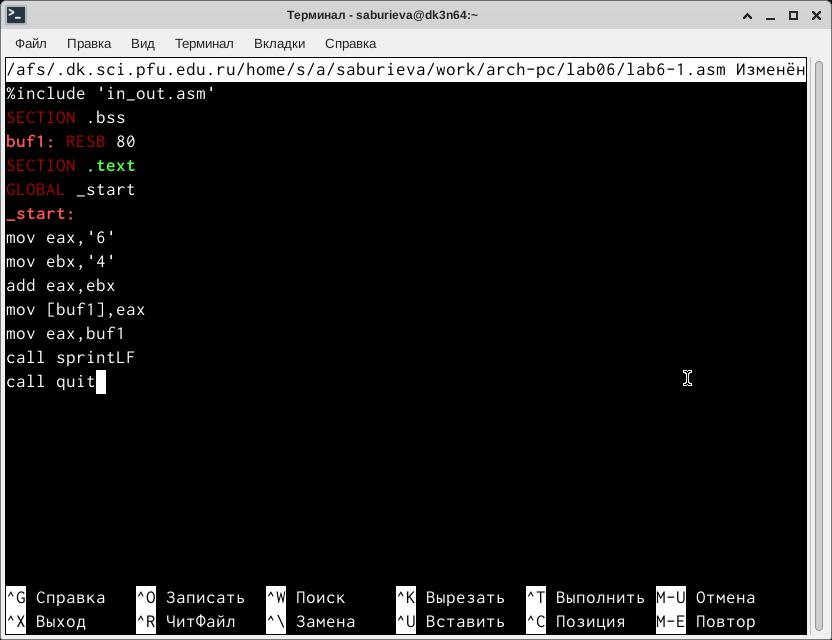
Создание каталога и файла в нём

Перед созданием исполняемого файла создала копию файла in\_out.asm в каталоге ~/work/arch-pc/lab06.



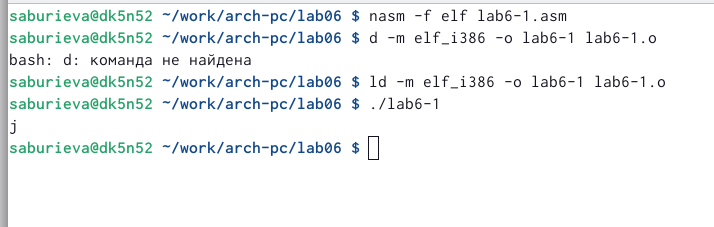
Копирование файла in\_out.asm в каталог

Ввела в файл lab6-1.asm текст программы из данного листинга и сохранила его.



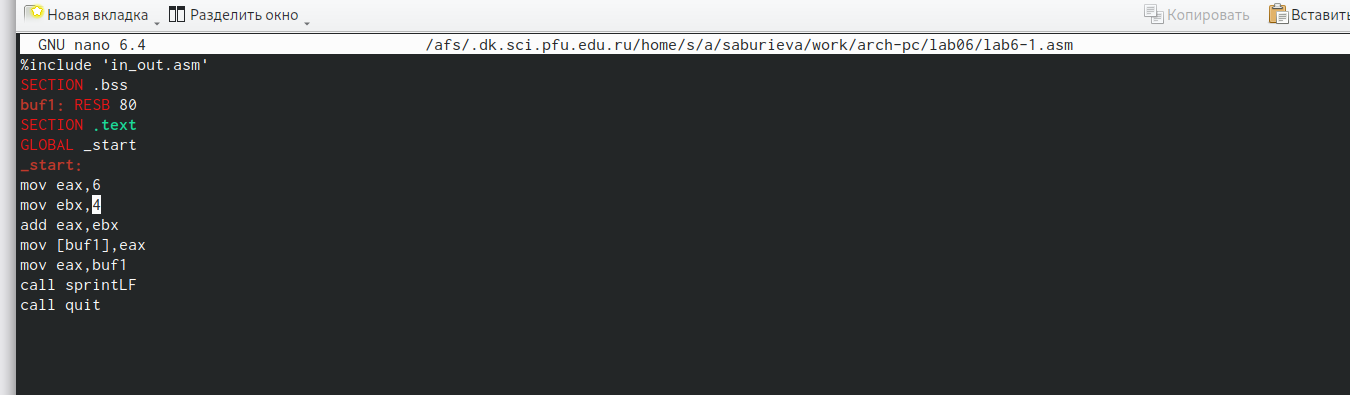
Ввод текста программы

Создала исполняемый файл и запустила его.Программа вывела символ j,потому что он соответствует сумме двоичных кодов символов 4 и 6 по системе ASCII.



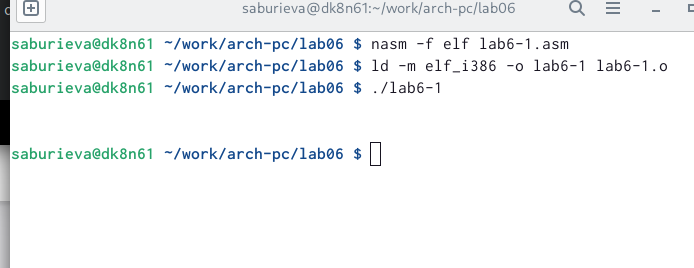
Создание исполняемого файла и его запуск

Изменила текст программы.



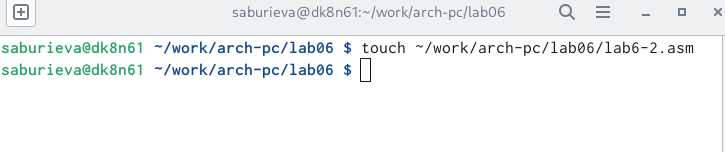
Изменение программы

Создала изменённый исполняемый файл и запустила его.Программа вывела символ перевода строки, потому что он соответствует коду 10=6+4 по системе ASII.



Создание исполняемого файла и его запуск

Создала файл lab6-2.asm в каталоге ~/work/arch-pc/lab06.



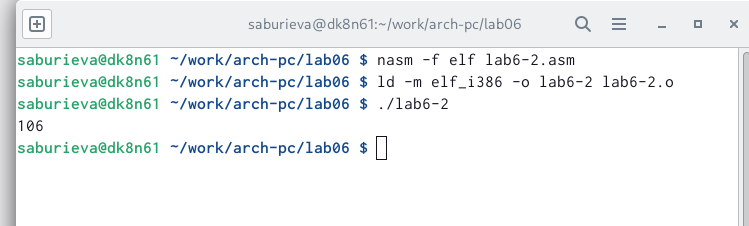
Создание файла lab6-2.asm

Ввела в созданный файл lab6-2.asm текст программы из листинга.



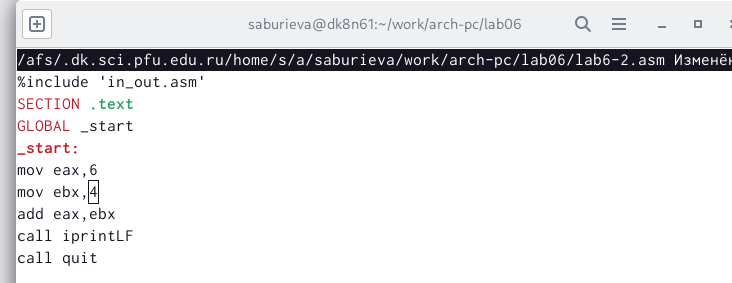
Ввод текста программы

Создала исполняемый файл и запустила его.Программа вывела, а именно сложение кодов символов “6” и “4”.В итоге получилось 106.



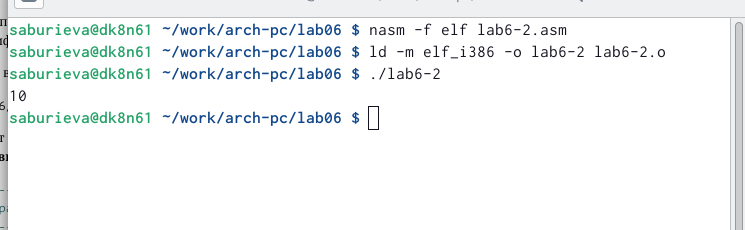
Создание исполняемого файла и его запуск

Далее опять изменила текст программы lab6-2.asm.



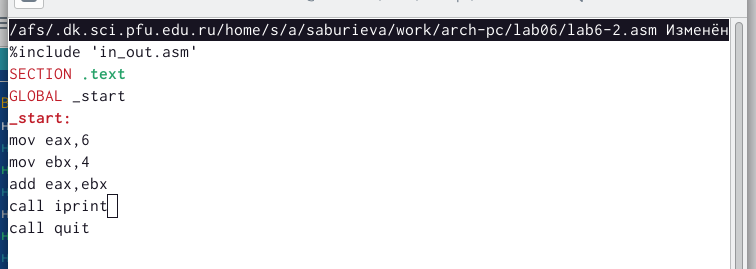
Изменение текста программы

Создала изменённый исполняемый файл lab6-2.asm и запустила его. Программа вывела сложение именно цифр.



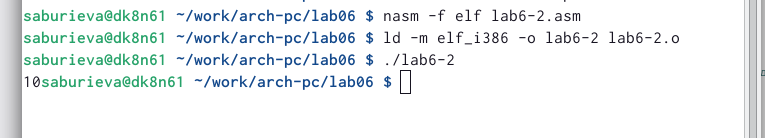
Создание исполняемого файла lab6-2.asm и его запуск

Заменила функцию iprintLF на iprint в данном файле.



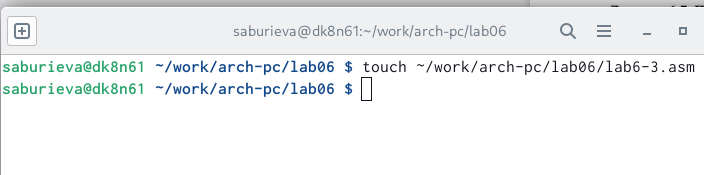
Замена на iprint

Создала изменённый исполняемый файл lab6-2.asm и запустила его.Вывод остался таким же, потому что символьный переносстрок не отображался, когда программа выполнялась с ограничением iprintLF, а iprint не добавлял к завершению символьный перенос строк, в отличие от iprintLF.



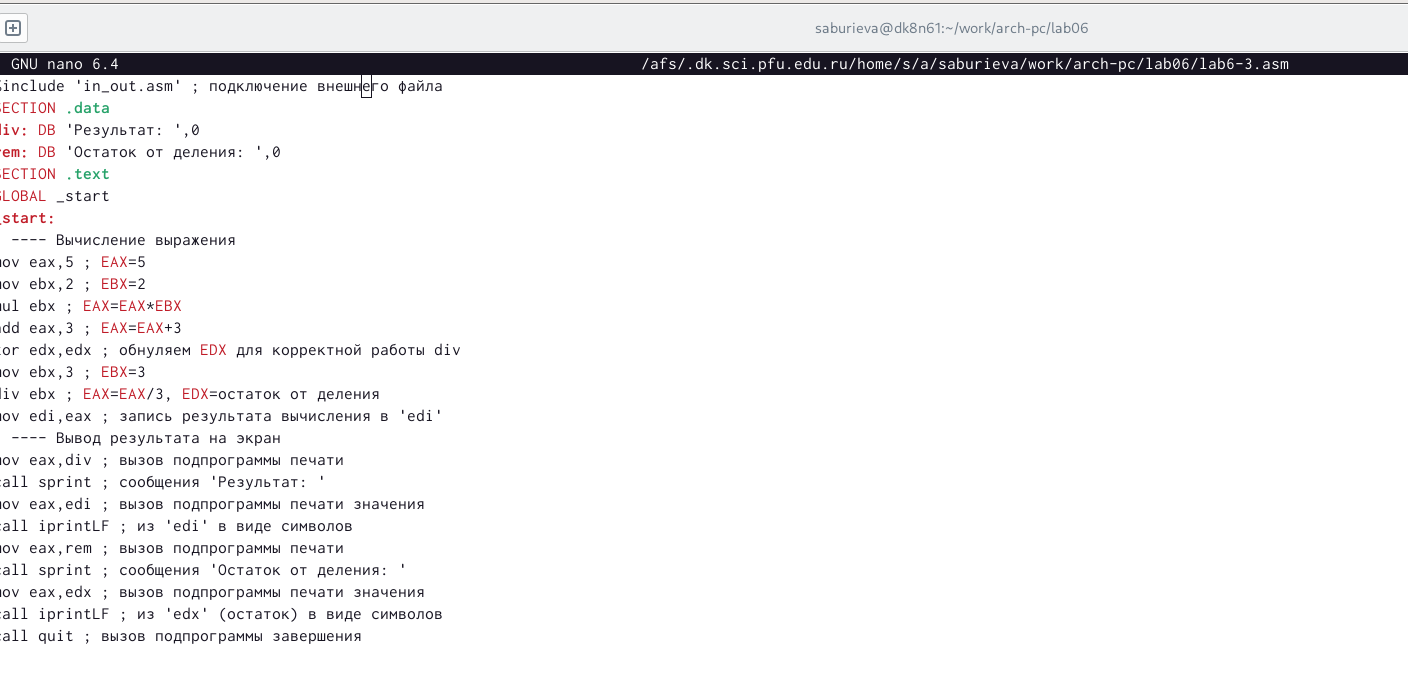
Создание исполняемого файла lab6-2.asm и его запуск

Создала файл lab6-3.asm в каталоге ~/work/arch-pc/lab06.



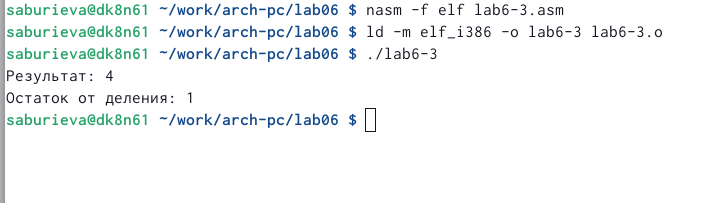
Создание файла lab6-3.asm

Внимательно изучила текст программы из листинга и ввела в lab6-3.asm.



Ввод текста программы в файл lab6-3.asm

Создала исполняемый файл и запустила его.На экран вывелся ответ.



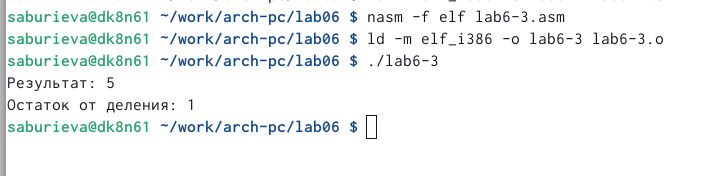
Создание исполняемого файла lab6-3.asm и его запуск

Изменила текст программы для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5.



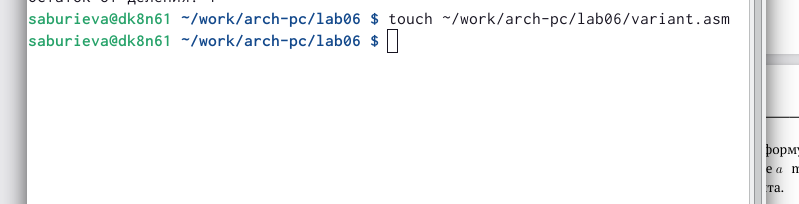
Изменение текста программы для алгебраического вычисления

Создала изменённый исполняемый файл для алгебраического вычисления и запустила его.



Создание изменённого исполняемого файла для алгебраического вычисления и его запуск

Создала файл variant.asm в каталоге ~/work/arch-pc/lab06.



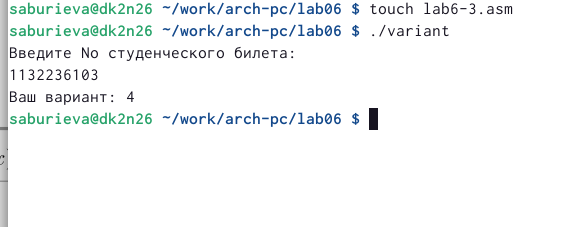
Создание файла variant.asm и его запуск

Внимательно изучила текст программы из листинга и ввела в файл variant.asm.



Ввод текста программы в файл variant.asm

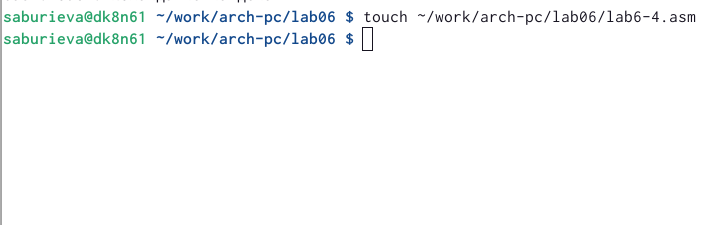
Создала файл variant.asm и запустила его.На запрос ввести студенческий билет,ввела номер своего студенческого билета.И мне выпало число 4.



Создание исполняемого файла variant.asm и его запуск

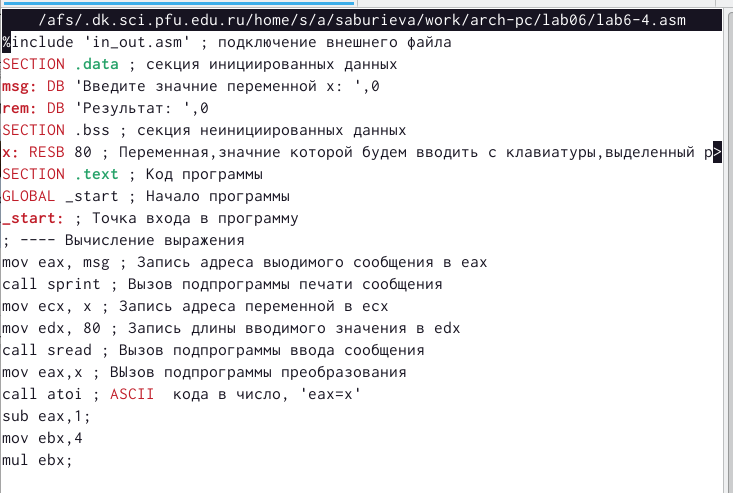
# 4 Выполнение заданий для самостоятельной работы

Создала файл lab6-4.asm в каталоге ~/work/arch-pc/lab06.



Создание файла lab6-4.asm

Ввела текст программы для алгебраического расчёта.



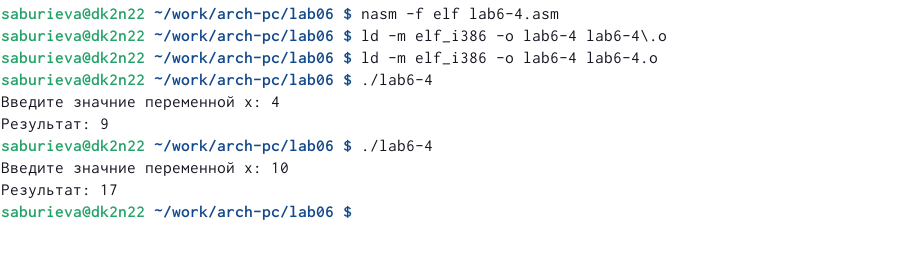
Ввод текста программы в файл lab6-4.asm

На рисунке показан сам текст моей программы,который я вводила.



Текст моей программы в редакторе

Создала исполняемый файл и запустила его.На запрос “Введите значение переменной x” ввела число 4.А затем ввела число 10.На экран вывелись оба значения x.



Создание исполняемого файла lab6-3.asm и его запуск

# 5 Ответы на вопросы.

1. Для вывода сообщения “Ваш вариант” предоставляются следующие строки кода.

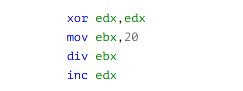


Код для вывода сообщения “Ваш вариант”

2)Инструкция mov ecx,x используется для того чтобы поставить адрес вводимой строки x в регистр, ecx mov edx,80-это запись в регистр edx длины вводимой строки, call sread-это вызов подпрограммы из внешнего файла, вводы сообщений с клавиатуры.

3)Инструкция “call atoi” используется для вызова подпрограмм из внешнего файла,преобразующий символы ASII-кода в имена чисел и записывающая результат в регистр eax.

4)Эти строки листинга 6.4 отвечают за вычисления варианта.

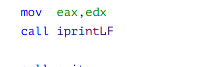


Код для вычисления варинта

5)При выполнении инструкции остаток деления div ebx записывается в регистр edx.

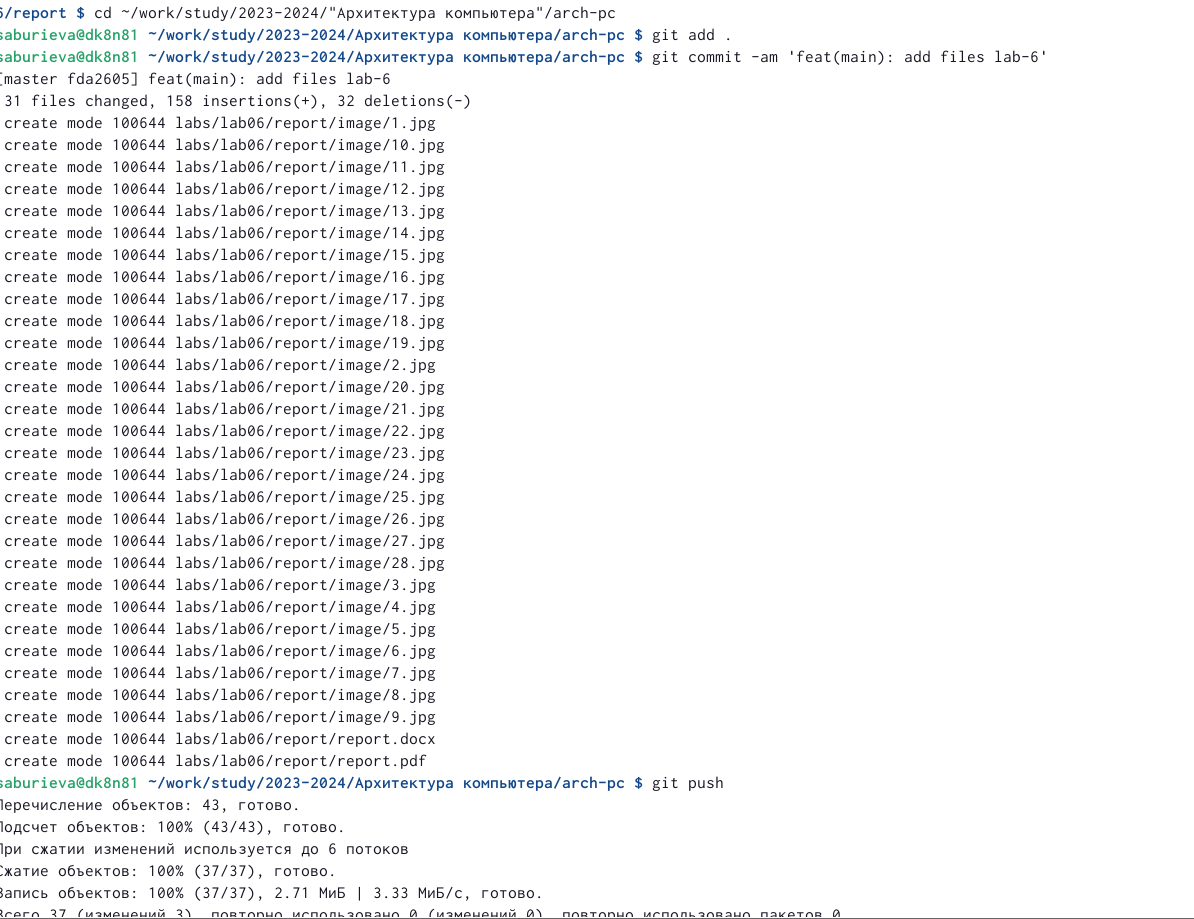
1. Инструкция “inc edx” используется для увеличения значения регистра edx на 1.

7)Для вывода результатов на экране используются эти коды.



Код для вывода резльтатов

Отправила все файлы на github.



Отправка на github

# 6 Выводы

Я освоила арифметических инструкций языка ассемблера NASM.

# Список литературы