Отчёт по лабораторной работе №6

дисциплина: архитектура компьютеров

Ведьмина Александра Сергеевна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

1. Ознакомиться с теорией
2. Изучить простые арифметические операции NASM.
3. Выполнить задания для самостоятельной работы.

# 3 Теоретическое введение

Большинство инструкций на языке ассемблера требуют обработки операндов. Адрес операнда - это место, где хранятся данные.

Способы адресации: 1. Регистровая 2. Непосредственная 3. Адресация памяти

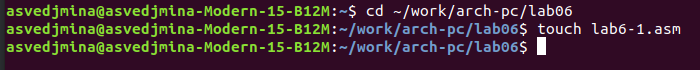
Команда целочисленного сложения add (от англ. addition - добавление) выполняет сложение двух операндов и записывает результат по адресу первого операнда. sub отвечает за вычитание.

Для команд умножения один из сомножителей указывается в команде и должен находиться в регистре или в памяти, но не может быть непосредственным операндом. Второй сомножитель в команде явно не указывается и должен находиться в регистре EAX,AX или AL, а результат помещается в регистры EDX:EAX, DX:AX или AX, в зависимости от размера операнда.

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.

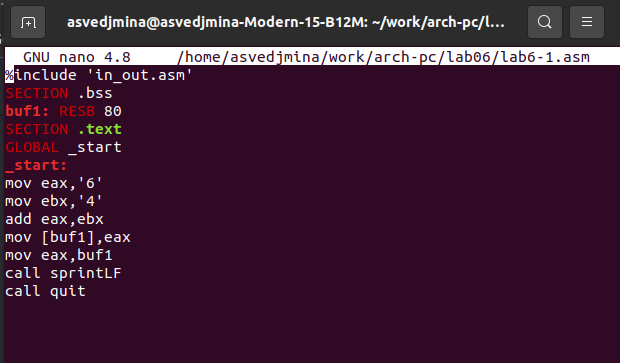
# 4 Выполнение лабораторной работы

Создаю каталог lab6, перехожу в него и создаю файл lab6-1.asm.



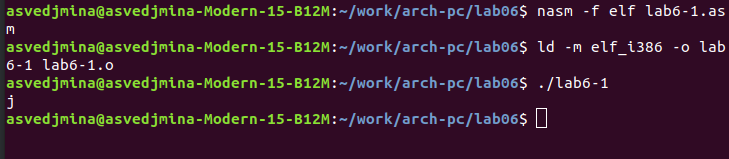
Создание файла lab6-1.asm

Ввожу в данный файл текст требуемой программы.



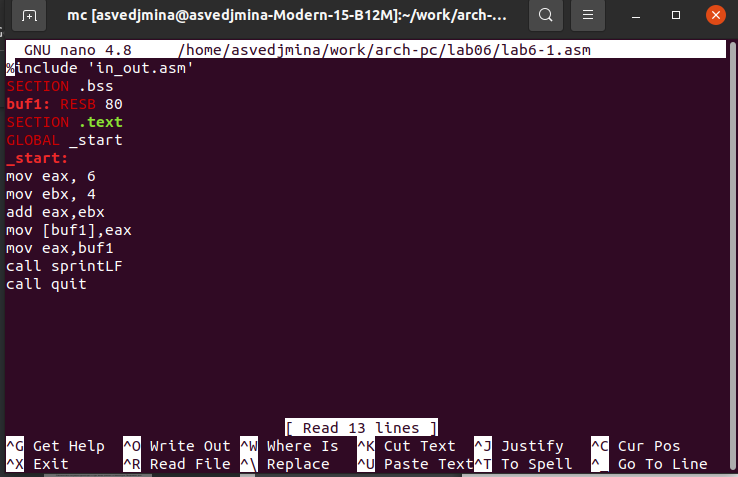
Ввод программы в lab6-1.asm

Создаю исполняемый файл и запускаю его.



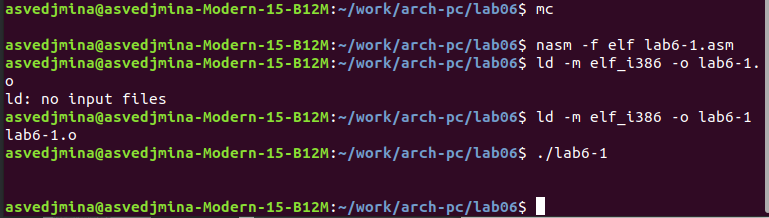
Запуск файла lab6-1

Затем в тексте программы заменяю строки mov eax,‘6’, mov ebx,‘4’ на строки mov eax,6,mov ebx,4.



Замена строк в lab6-1.asm

Создаю исполняемый файл и запускаю его.



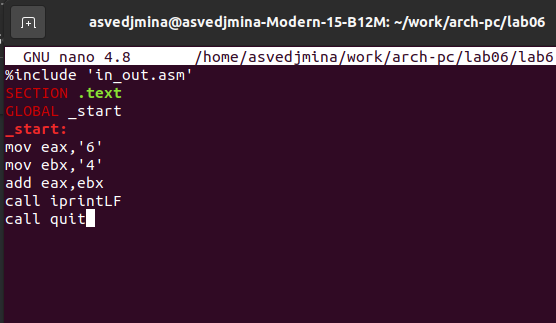
Запуск файла lab6-1

Символ, полученный в результате программы, не отображается. По таблице ASCII коду 10 соответствует символ, изображающий круг в квадрате.

Создаю файл lab6-2.asm и ввожу в него предложенный текст проограммы.

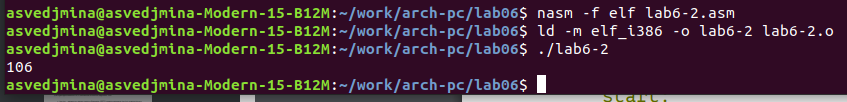
Создание файла lab6-2.asm

Создание файла lab6-2.asm



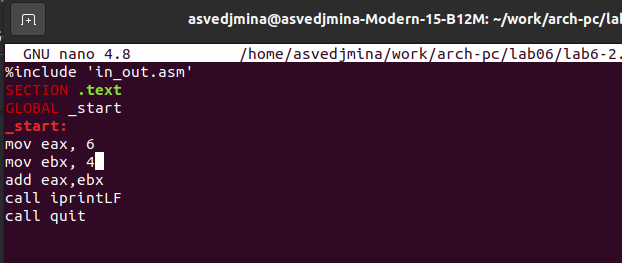
Ввод программы в lab6-2.asm

Создаю исполняемый файл и запускаю его.



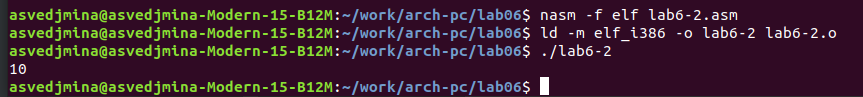
Запуск файла lab6-2

Заменяю mov eax,‘6’ и mov ebx,‘4’ на mov eax,6 и mov ebx,4 в файле lab6-2.asm.



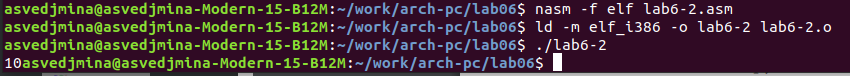
Замена строк в файле lab6-2.asm

Создаю исполняемый файл и запускаю его.



Запуск файла lab6-2

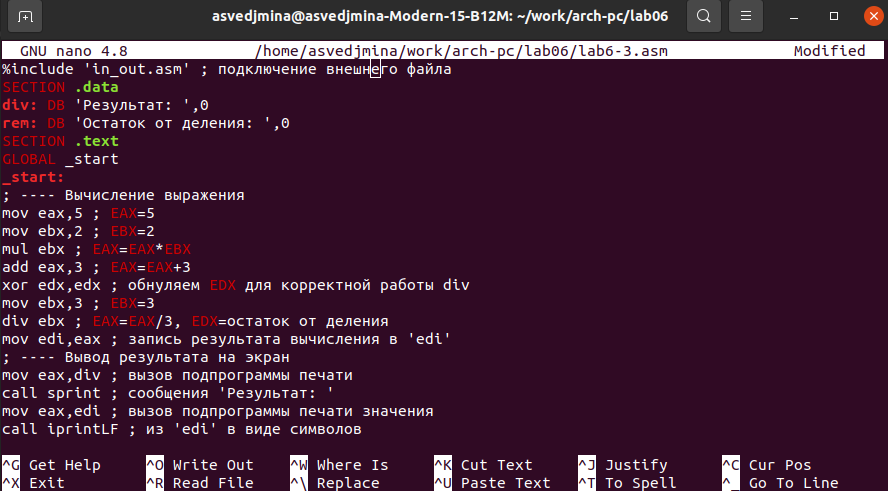
При исполнении программы получен результат 10. Заменяю функцию iprintLF на iprint в файле lab6-2.asm, затем создаю исполняемый файл и запускаю его.



Запуск изменённого файла lab6-2

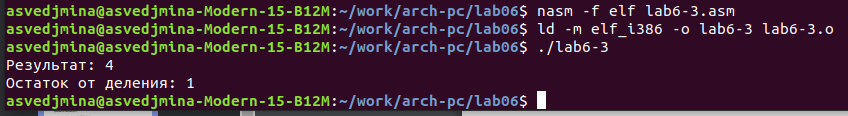
Функция iprint в отличие от iprintLF предлагает нам ввести следующую команду на той же строке, на которой был выведен результат выполнения программы.

Далее создаю файл lab6-3.asm и ввожу программу вычисления выражения f(x)=(5\*2+3)/3.



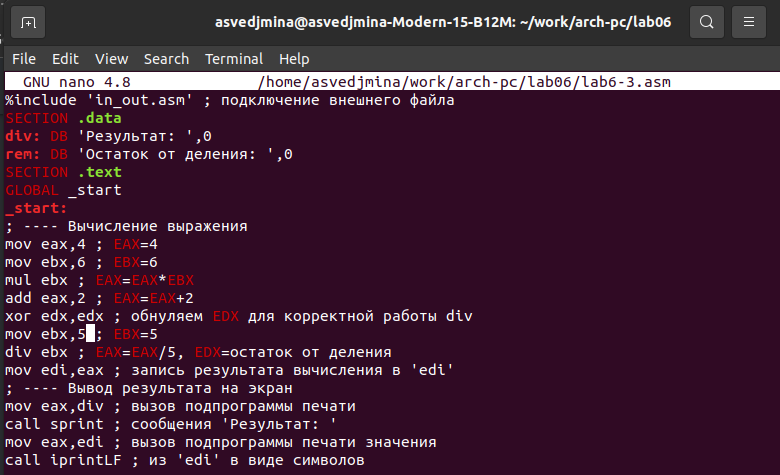
Ввод программы в файл lab6-3.asm

Создаю исполняемый файл и запускаю его.



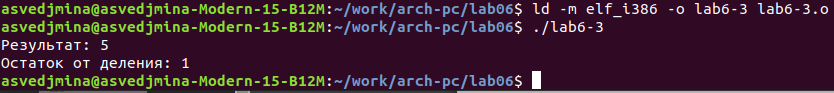
Запуск файла lab6-3

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



Изменение текста программы lab6-3.asm

Создаю исполняемый файл и проверяю его работу.



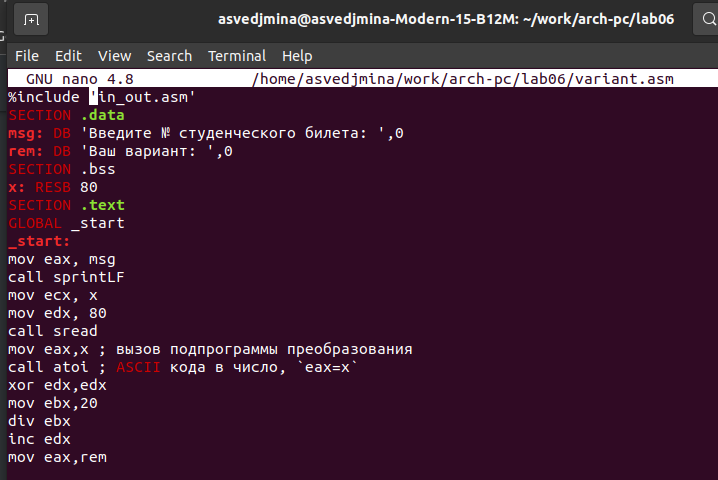
Запуск файла lab6-3

Создаю файл variant.asm.

Создание файла variant.asm

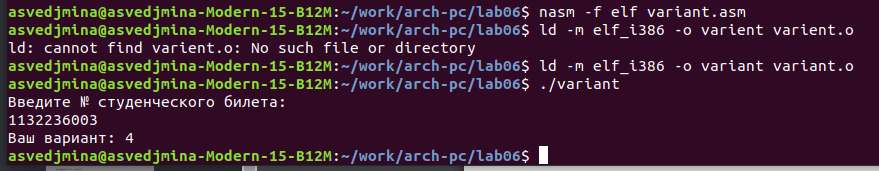
Создание файла variant.asm

Ввожу в данный файл программу для вычисления варианта задания.



Ввод программы variant.asm

Создаю исполняемый файл и запускаю его.



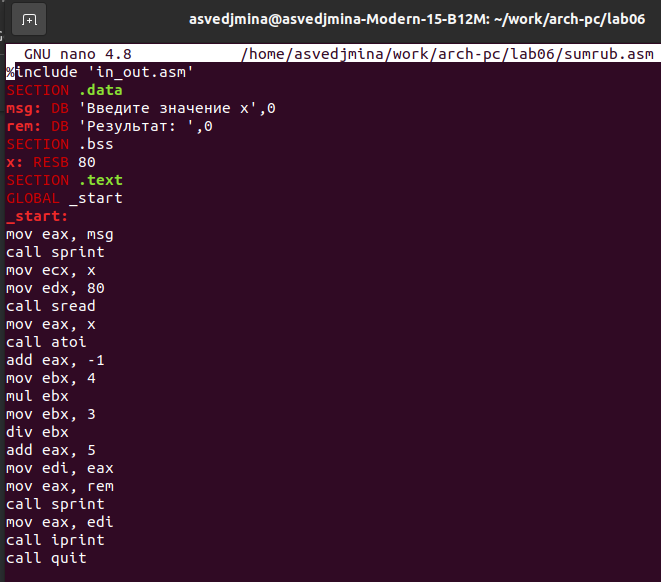
Запуск файла variant

Ответы на вопросы по листингу 6.4:

1. За вывод на экран сообщения “Ваш вариант:” в листинге 6.4 отвечают строки mov eax,rem и call sprint.
2. mov ecx,x прокладывает адрес введённой строки в ecx; mov edx,80 обозначает запись в регистр edx; call sread используется для вызова подпрограммы, которая считывает текст с клавиатуры.
3. call atoi нужна для вызова подпрограммы, которая преобразовывает код символа из таблицы ASCII в сам символ и записыывает его в eax.
4. За вычисление варианта отвечают строки xor edx, edx; mov ebx,20; div ebx; inc edx.
5. Остаток от деления от div ebx записывается в edx.
6. inc edx увеличивает значение регистра на один.
7. За вывод на экран результата вычислений отвечают mov eax, edx и call iprintLF.

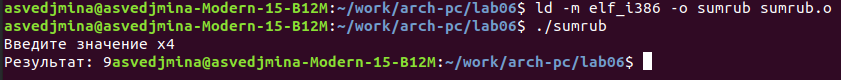
# 5 Выполнение заданий для самостоятельной работы

Номер моего варианта - 4, поэтому я буду реализовывать функцию f(x) = 4/3(x-1)+5. Для выполнения задания создаю файл sumrub.asm и записываю в него необходимую программу.

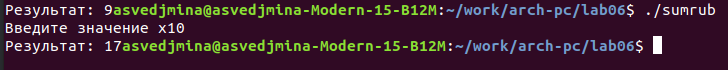


Программа для вычисления функции

Создаю исполняемый файл и запускаю программу.



Запуск файла sumrub при х = 4



Запуск файла sumrub при х = 10

# 6 Выводы

В ходе выполнения лабораторной работы я освоила арифметические операции в NASM.