Отчёт по лабораторной работе №6

Дисциплина: архитектура компьютера

Худдыева Дженнет

Содержание

# 1 Цель работы

Цель данной лабораторной раюоты - освоение арефметических инструкций языка ассемблера NASM.

# 2 Задание

1.Символьные и численные данные в NASM. 2.Выполнение арифметических операций в NASM. 3.Выполнение заданий для самостоятельной работы.

# 3 Теоретическое введение

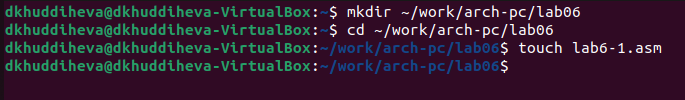
Большинство инструкций на языке ассемблера требуют обработки операндов.Адрес операнда предоставляет место,где хранятся данные, подлежащие обработке.Это могут быть данные хранящиеся в регистре или в ячейке памяти. - Регистровая адресация - операнды хранятся в регистрах и в команде используются имена этих регистров,например: mov ax,bx. - Непосредственная адресация - значение операнда задаётся непосредственно в команде,например: mov ax,2. - Адресация памяти - операнд задаёт адрес в памяти.В команде указывается символическое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию.

Ввод информации с клавиатуры и вывод её на экран осуществляется в символьном виде.Кодирование этой информации производится сог- ласно 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.Среди инструкций NASM нет такой,которая выводит числа(не в символьном виде).Поэтому,например,чтобы вывести число,надо предворительно преобразовать его цифры в ASCII-коды этих цифр и выводит 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 число будет воспринят как один ASCII- -символ и выведет на экран эти символы.

# 4 Выполнение лабораторной работы

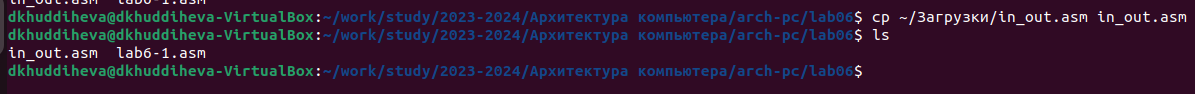
## 4.1 Символьные и численные данные в NASM

С помощью утилиты mkdir создаю директорию,в которой буду создавать файлы с программами для лабораторной работы №6 (рис.[??]) С помощью утилиты touch создаю файл lab6-1.asm



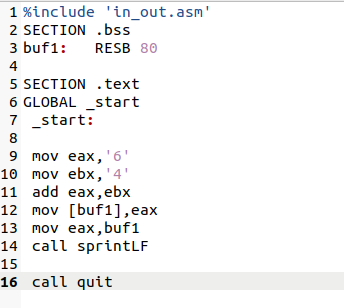
Создание директории

Копирую в текущий каталог файл in\_out.asm с помощью утилиты cp (рис.[??]).



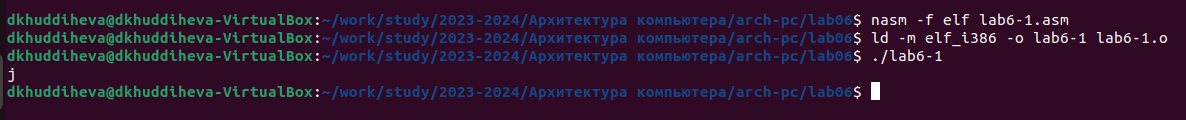
Создание копии файла

Открываю созданный файл lab6-1.asm,вставляю в него программу вывода значения регистра eax (рис.[??]).



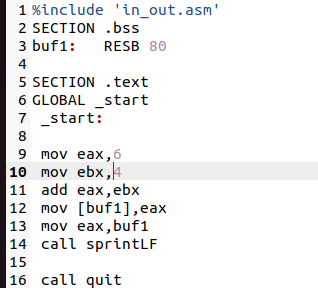
Редактирование файла

Создаю исполняемый файл программы и запускаю его (рис.[??]).Вывод программы: символ j потому что программа вывела символ, соответствующий по системе ASCII сумме двоичных кодов символов 4 и 6.



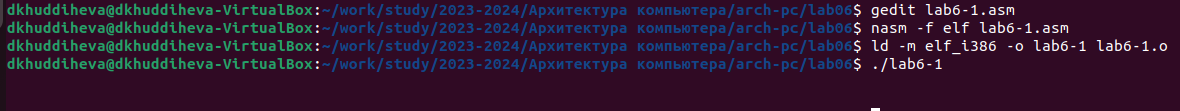
Запуск исполняемого файла

Изменяю в тексте программы “6” и “4” на цифру 6 и 4 (рис.[??]).



Редактирование файла

Создаю новый исполняемый файл программы и запускаю его (рис.[??]).Теперь вывелся символ 10,это символ перевода строки,этот символ не отображается при выводе на экран.



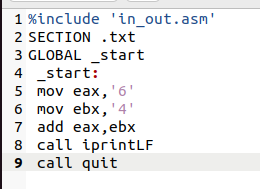
Запуск исполняемого файла

Создаю новый файл lab6-2.asm с помощью утилиты touch (рис.[??]).

Создание файла

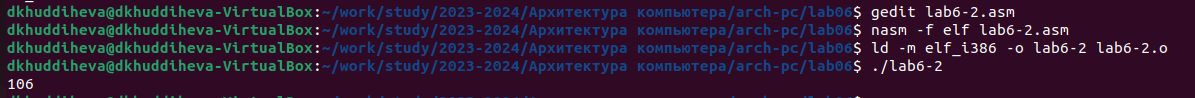
Создание файла

Ввожу в файл текст другой программы для вывода значения регистра eax (рис.[??]).



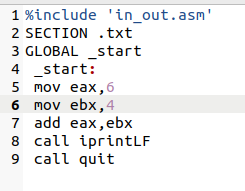
Редактирование файла

Создаю и запускаю исполняемый файл lab6-2 (рис.[??]).Теперь вывод числа 106,потому что программа позволяет вывести именно число,а не символ,хотя всё ещё происходит именно сложение кодов символов “6” и “4”.



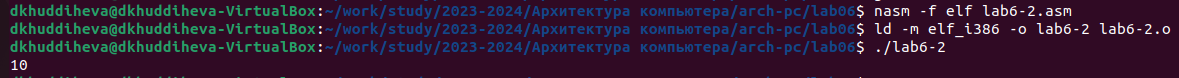
Запуск исполняемого файла

Заменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на число 6 и 4 (рис.[??]).



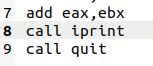
Редактирование файла

Создаю и запускаю новый исполняемый файл (рис.[??]).Теперь программа складывает не соответствующие символом коды в системе ASCII,а сами числа,поэтому вывод 10.



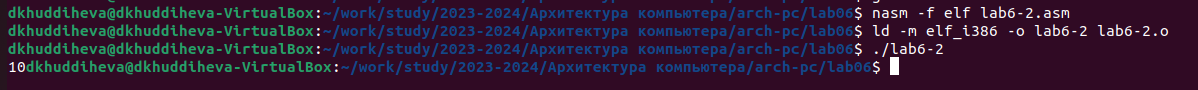
Запуск исполняемого файла

Заменяю в тексте программы функцию iprintLF на iprint (рис.[??]).



Редактирование файла

Создаю и запускаю новый исполняемый файл (рис.[??]).Вывод не изменился,потому что символ переноса строки не отображался, когда программа исполнялась с функцией iprintLF,а iprint не добавляет к выводу символ переноса строки, в отличие от iprintLF.



Запуск исполняемого файла

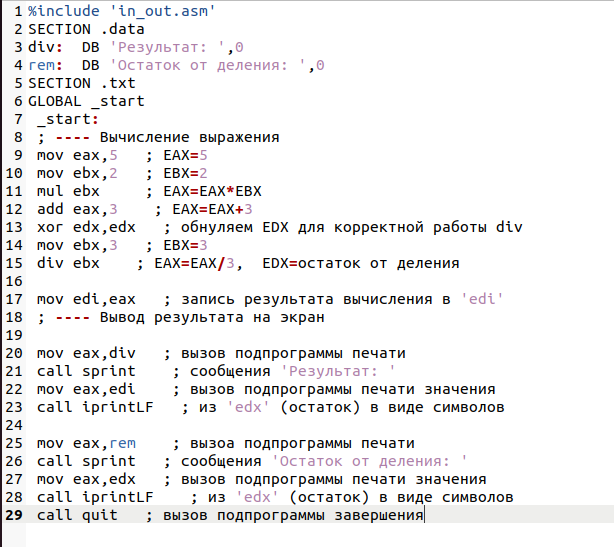
## 4.2 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm с помощью утилиты touch (рис.[??]).

Создание файла

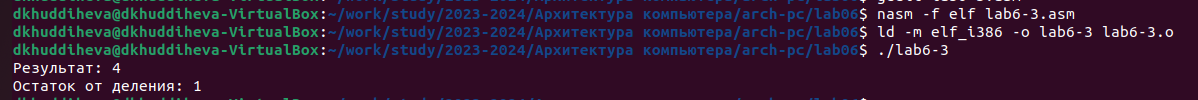
Создание файла

Ввожу в созданный файл текст программы для вычисления значения выражения f(x)=(5\*2+3)/3 (рис.[??]).



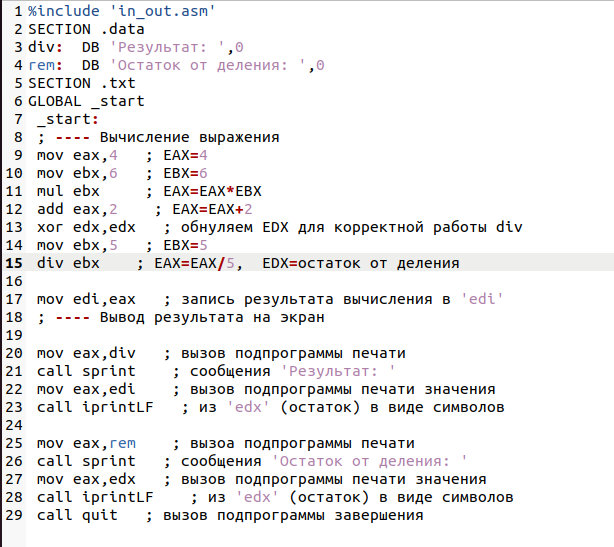
Редактирование файла

Создаю исполняемый файл и запускаю его (рис.[??]).



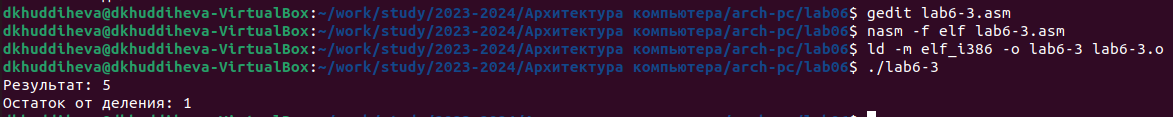
Запуск исполняемого файла

Изменяю программу так,чтобы она вычисляла значение выражения (рис.[??]).



Изменение программы

Создаю и запускаю новый исполняемый файл (рис.[??]).



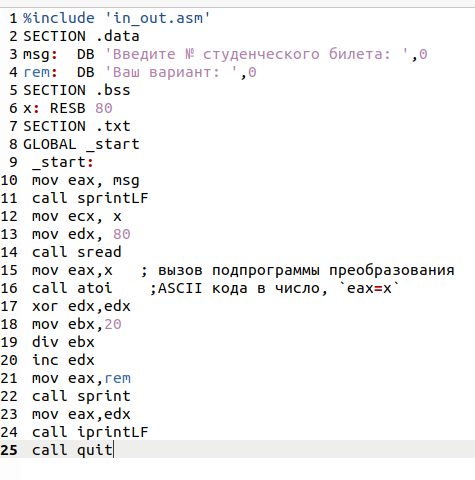
Запуск исполняемого файла

Создаю файл variant.asm с помощью утилиты touch (рис.[??]).

Создание файла

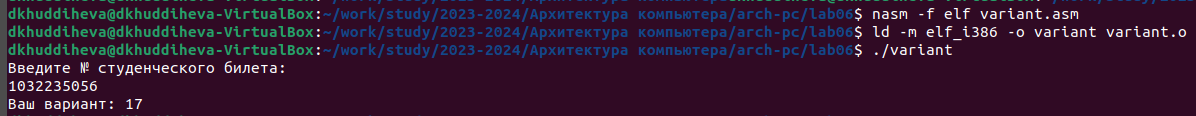
Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис.[??]).



Редактирование файла

Создаю и запускаю исполняемый файл (рис.[??]).Ввожу номер своего студ.билета с клавиатуры,программа вывела что мой вариант-17



Запуск исполняемого файла

### 4.2.1 Ответы на вопросы по программе

1.За вывод сообщения “Ваш вариант” отвечают строки кода:

mov eax,rem  
 call sprint

2.Инструкция mov ecx, x используется,чтобы положит адрес вводимой строки x в регистр ecx mov edx, 80 - запись в регистр еdx длины вводимой строки call sread - вызов программы из внешнего файла,обеспечивающей ввод сообщения с клавиатуры

3.call atoi используется для вызова подпрограммы из внешнего файла,которая преобразует ASCII -код символа в целое и записывает результат в регистр eax

4.За вычисления варианта отвечают строки:

xor edx,edx ; обнуление edx для корректной работы div  
mov ebx,20 ; ebx=20  
div ebx ; eax=eax/20, edx-остаток от деления  
inc edx ; edx=edx+1

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

6.Инструкция inc edx увеличивает регистра edx на 1

7.За вывод на экран результатов вычислений отвечают строки:

mov eax,edx  
call iprintLF

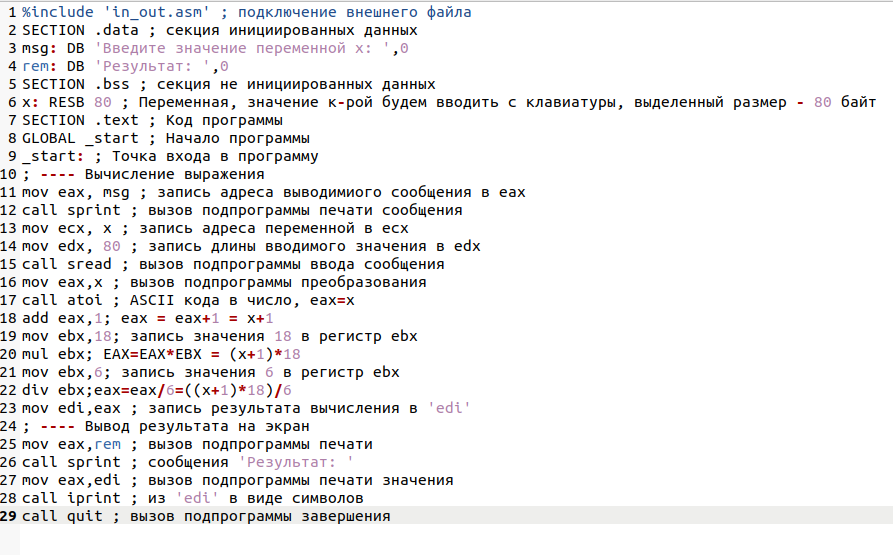
## 4.3 Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm с помощью утилиты touch (рис.[??])

Создание файла

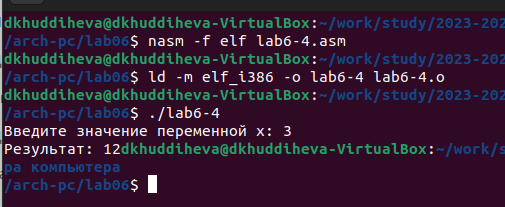
Создание файла

Открываю созданной файл для редактирования,ввожу в него текст программы для вычисления значения выражения 18\*(х+1)/6.Это выражение было под вариантом 17 (рис.[??])



Написание программы

Создаю и запускаю исполняемый файл (рис.[??])



Запуск исполняемого файла

\*\*Листинг 4.1. Программа для вычисления значения выражения 18\*(х+1)/6.\*\*

# 5 Выводы

При выполнении данной лабораторной работы я освоила арифметические инструкции языка ассемблера NASM