Отчет по лабораторной работе №6

Арифметические операции в NASM

Ашуров Захид Фамил оглы

Содержание

# 1 Цель работы

Освоить арифметические инструкции языка ассемблера NASM

# 2 Задание

Символьные и численные данные в NASM

Выполнение арифметических операций в NASM

# 3 Теоретическое введение

Адресация в NASM

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. Существует три основных способа адресации:

• Регистровая адресация – операнды хранятся в регистрах и в команде используются имена этих регистров, например: mov ax,bx. • Непосредственная адресация – значение операнда задается непосредственно в ко- манде, Например: mov ax,2. • Адресация памяти – операнд задает адрес в памяти. В команде указывается символи- ческое обозначение ячейки памяти, над содержимым которой требуется выполнить операцию.

Например, определим переменную intg DD 3 – это означает, что задается область памяти

размером 4 байта, адрес которой обозначен меткой intg. В таком случае, команда

mov eax,[intg]

копирует из памяти по адресу intg данные в регистр eax. В свою очередь команда mov [intg],eax

запишет в память по адресу intg данные из регистра eax.

Также рассмотрим команду

mov eax,intg

В этом случае в регистр eax запишется адрес intg. Допустим, для intg выделена память начиная с ячейки с адресом 0x600144, тогда команда mov eax,intg аналогична команде mov eax,0x600144 – т.е. эта команда запишет в регистр eax число 0x600144.

Целочисленное сложение add.

Схема команды целочисленного сложения add (от англ. addition - добавление) выполняет

сложение двух операндов и записывает результат по адресу первого операнда. Команда add работает как с числами со знаком, так и без знака и выглядит следующим образом:

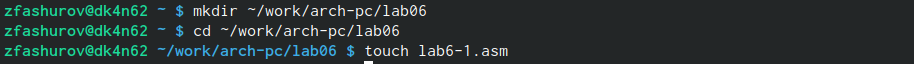
Допустимые сочетания операндов для команды add аналогичны сочетаниям операндов

для команды mov. Так, например, команда add eax,ebx прибавит значение из регистра eax к значению из регистра ebx и запишет результат в регистр eax.

Примеры: add ax,5 ; AX = AX + 5 add dx,cx ; DX = DX + CX add dx,cl ; Ошибка: разный размер операндов.

# 4 Выполнение лабораторной работы

Создаем каталог для программ лабораторной работы №6, переходим в него и создаем файл lab6-1.asm (Рис. ??).



Создание каталога для программ для лабораторной работы №6 и переходим в него и создаем файл lab6-1.asm

Вписываем в lab6-1.asm текст из листинга 6.1 (Рис. ??).



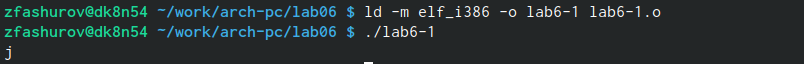
Вписываем в lab6-1.asm

Создаем файл lab6-1.asm (Рис. ??).

Создание файла

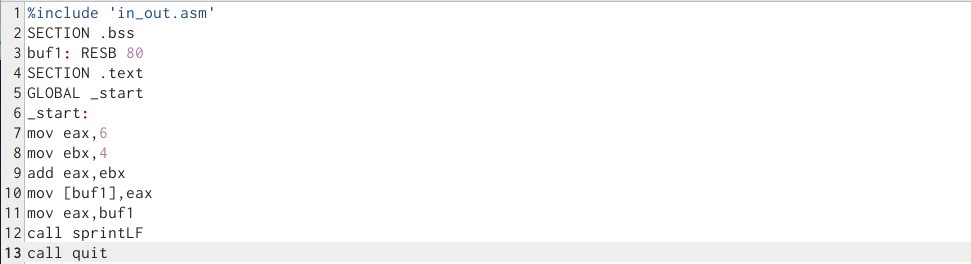
Создание файла

Запускаем файл lab6-1.asm (Рис. ??).



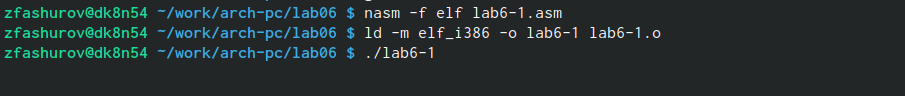
Запуска файла

Редактируем текст файла lab6-1.asm (Рис. ??).



Редактирование текста файла lab6-1.asm

Создаем и запускаем измененный файл lab6-1.asm (Рис. ??).



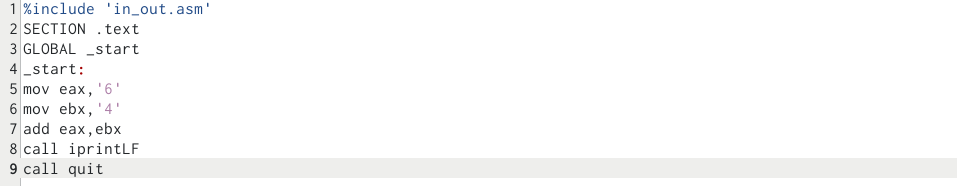
Создание и запускание измененного файла lab6-1.asm

Создаем файл lab6-2.asm (Рис. ??).

Создание файла lab6-2.asm

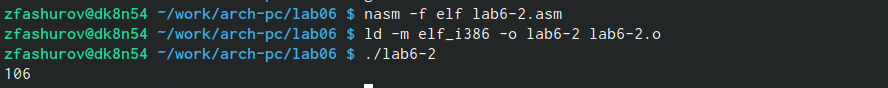
Создание файла lab6-2.asm

Вписываем текст в lab6-2.asm (Рис. ??).



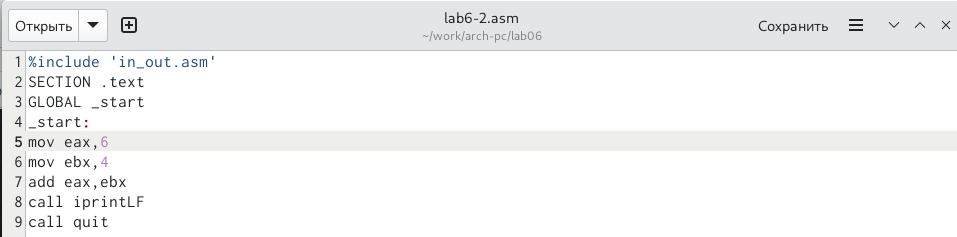
Вписывание в текст lab6-2.asm

Создаем и запускаем файл lab6-2.asm (Рис. ??).



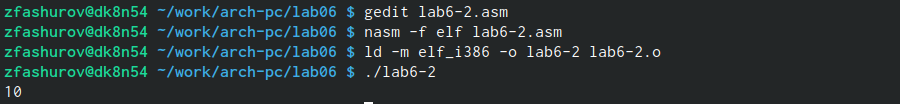
Создание и запускание файла lab6-2.asm

Редактируем lab6-2.asm (Рис. ??).



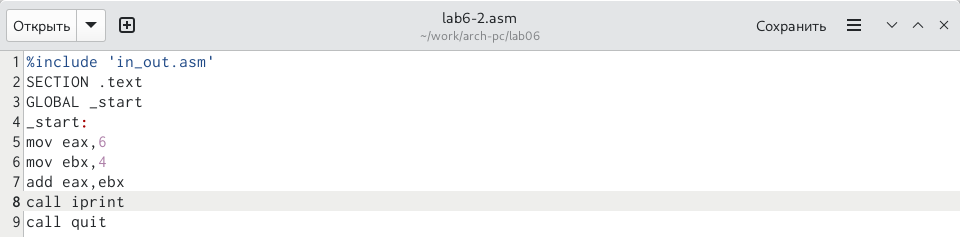
Редактирование lab6-2.asm

Создаем и запускаем измененный файл lab6-2.asm (Рис. ??).



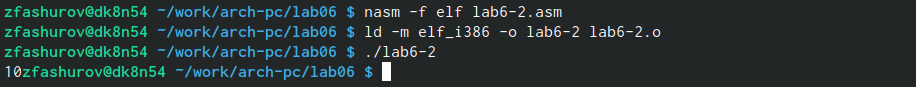
Создание и запускание измененного файла lab6-2.asm

Меняем в тексте с inprintLF на inprint (Рис. ??).



Смена с inprintLF на inprint

Создаем и запускаем файл lab6-2.asm (Рис. ??).



Создание и запускание файл lab6-2.asm

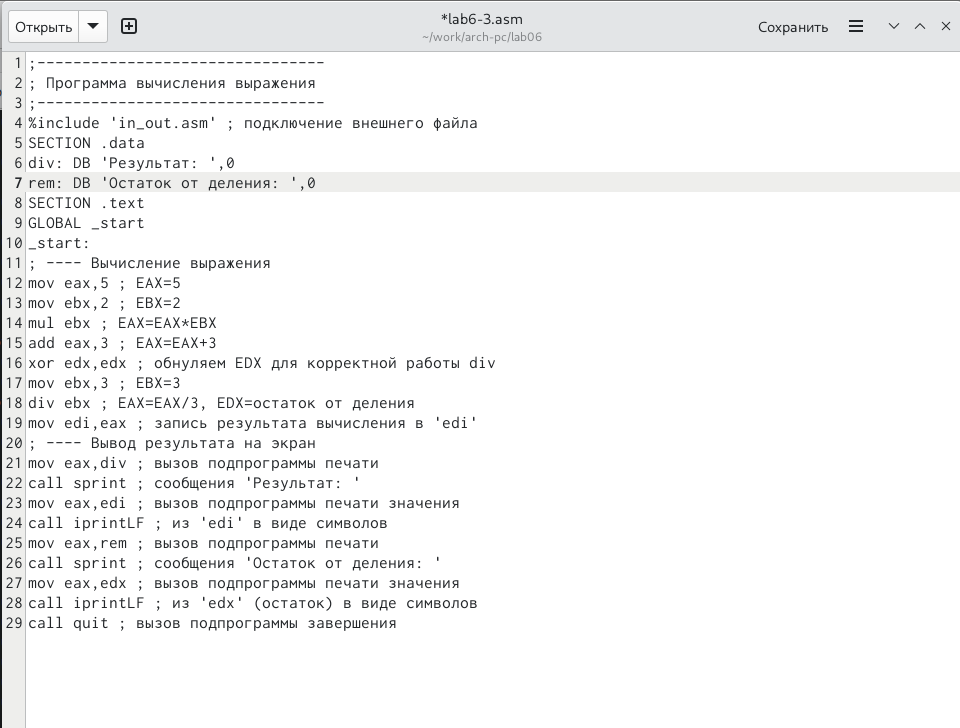
* Разница заключается в том что 10 в первом случае была записана отдельно от строки, а вторая слипнулась с 10.

Создаем файл lab6-3.asm (Рис. ??).

Создание файла lab6-3.asm

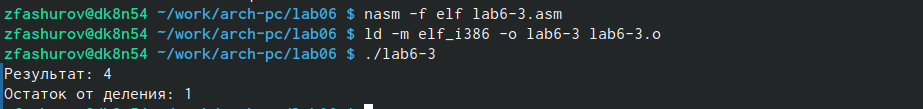
Создание файла lab6-3.asm

Вписываем в lab6-3.asm (Рис. ??).



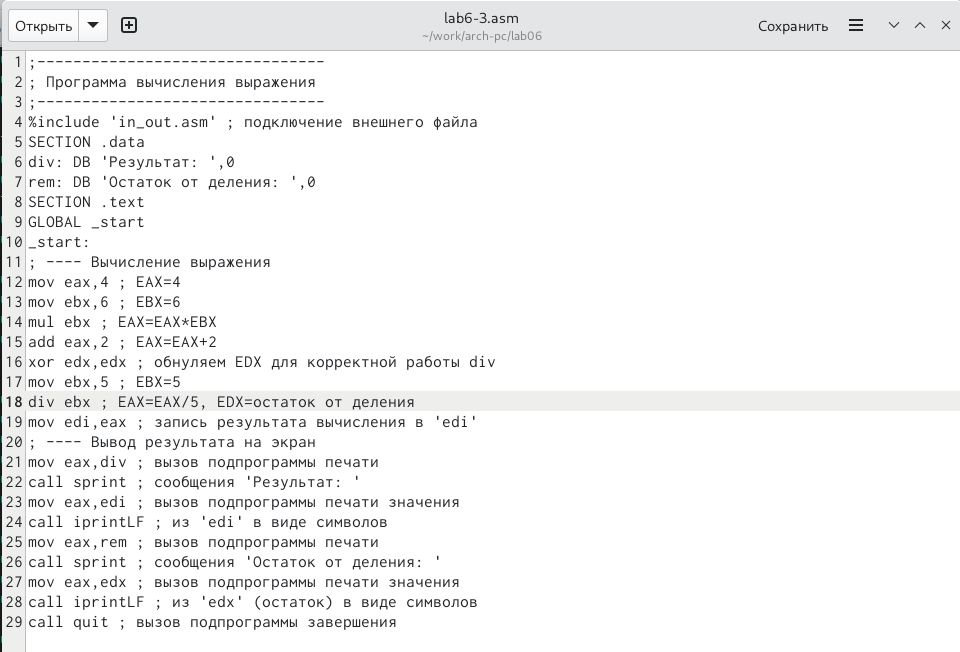
Вписывание в lab6-3.asm

Создаем и запускаем файл lab6-3.asm (Рис. ??).



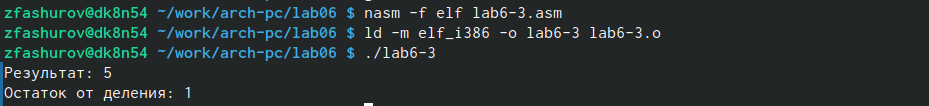
Создание и запускание lab6-3.asm

Редактируем файл lab6-3.asm (Рис. ??).



Редактирование файла lab6-3.asm

Создаем и запускаем файл lab6-3.asm(Рис. ??).



Создание и запускание файла lab6-3.asm

Создаем файл variant.asm(Рис. ??).

Создание файла variant.asm

Создание файла variant.asm

# 5 Ответы на вопросы по программе

Вопрос 1: За вывод сообщения “Ваш вариант” отвечает строка кода: mov eax,rem call sprint

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

Вопрос 3: call atoi исользуется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax.

Вопрос 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

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

Вопрос 6: Инструкция inc edx увеличивает значение регистра edx на 1

Вопрос 7: За вывод на экран результатов вычислений отвечают строки: mov eax,edx call iprintLF

# 6 Выводы

Я освоил арифметические инструкции языка ассемблера NASM

# Список литературы