Отчёт по лабораторной работе №6

Дисциплина: архитектура компьютера

Кайнова Алина Андреевна

Содержание

# 1 Цель работы

Освоить арифметические инструкции 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 задается непосредственно в команде, Например: mov ax,2. - Адресация памяти – операнд задает 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 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. Среди инструкций NASM нет такой, которая выводит числа (не в символьном виде). Поэтому, 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 данные 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.

# 4 Выполнение лабораторной работы

##Символьные и численные данные в NASM

Создаю каталог для лабораторной работы №7 и перехожу в него

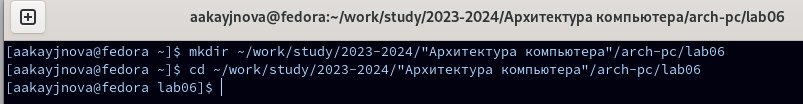


Figure 1: Создание директории

Создаю файл lab6-1.asm

Figure 2: Создание файла

Figure 2: Создание файла

Копирую файл in\_out.asm в текущий каталог

Figure 3: Копирование файла

Figure 3: Копирование файла

Открываю созданный файл и копирую в него программу вывода значения регистра eax

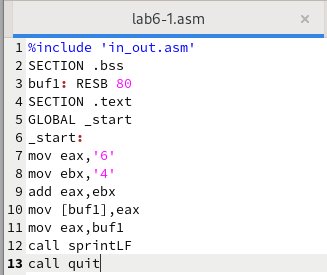


Figure 4: Редактирование файла

Создаю исполняемый файл и запускаю его

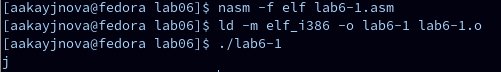


Figure 5: Запуск исполняемого файла

Изменяю в тексте программы в файле lab6-1.asm символы “6” и “4” на цифры 6 и 4 соответственно

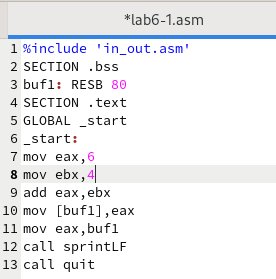


Figure 6: Редактирование файла

Создаю новый исполняемый файл программы и запускаю его

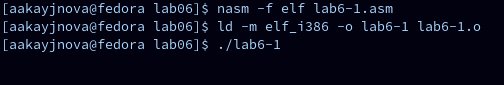


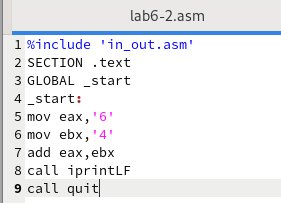
Figure 7: Запуск исполняемого файла

Создаю новый файл lab6-2.asm

Figure 8: Создание файла

Figure 8: Создание файла

Вставляю в файл текст другой программы для вывода значения eax

https://esystem.rudn.ru/pluginfile.php/2089086/mod\_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%966.%20%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8%20%D0%B2%20NASM..pdf

Создаю и запускаю исполняемый файл lab6-2

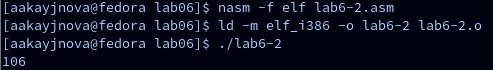


Figure 9: Запуск исполняемого файла

Изменяю в тексте программы в файле lab6-2.asm символы “6” и “4” на цифры 6 и 4 соответственно

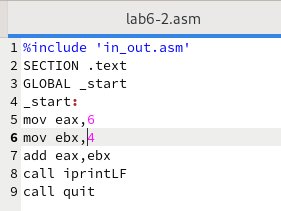


Figure 10: Редактирование файла

Создаю и запускаю исполняемый файл

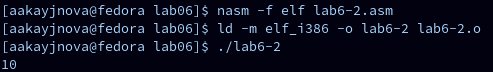


Figure 11: Запуск исполняемого файла

Заменяю функцию iprintLF на iprint в тексте программы. Вывод не изменился, так как символ переноса строки не отображался, когда программа иполнялась с функцией iprintLF, а iprint не добавляет к выводу символ переноса строки, в отличие от iprintLF

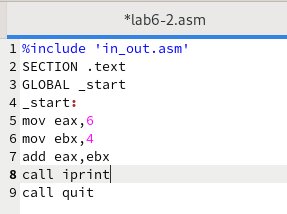


Figure 12: Редактирование файла

Создаю и запускаю новый исполняемый файл

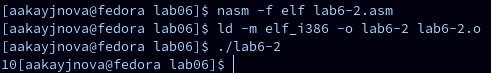


Figure 13: Запуск исполняемого файла

## 4.1 Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm

Figure 14: Создание файла

Figure 14: Создание файла

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

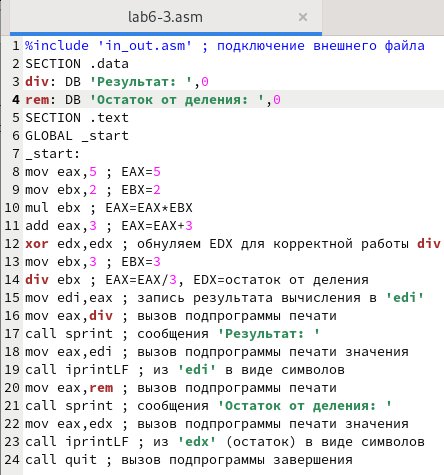


Figure 15: Редактирование файла

Создаю исполняемый файл и запускаю его

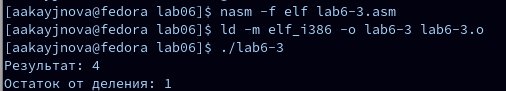


Figure 16: Запуск исполняемого файла

Изменяю текст программы, чтобы вычислить значение выражения f(x)=(4\*6+2)/5

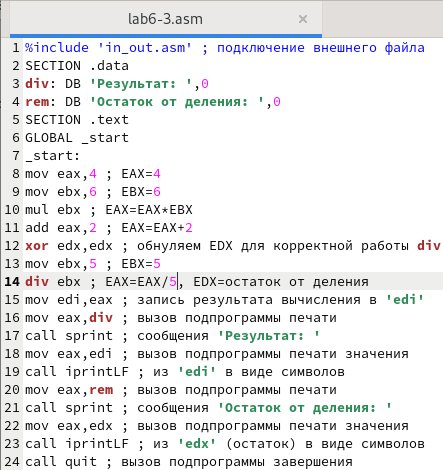


Figure 17: Изменение программы

Создаю и запускаю новый исполняемый файл

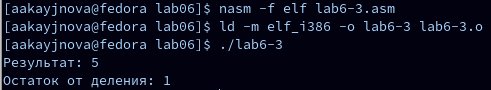


Figure 18: Запуск исполняемого файла

Создаю файл variant.asm

Figure 19: Создание файла

Figure 19: Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета

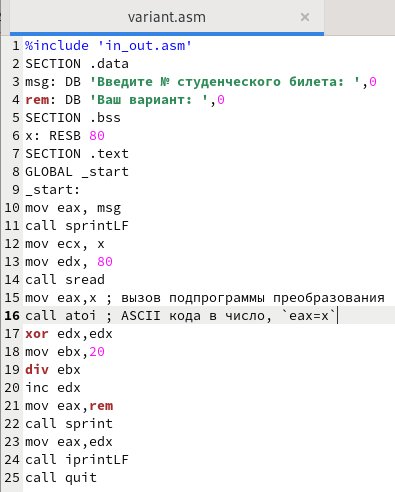


Figure 20: Редактирование файла

Создаю и запускаю исполняемый файл, ввожу номер своего студ. билета

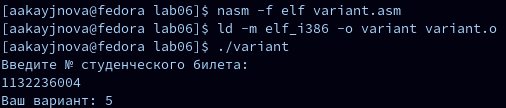


Figure 21: Запуск исполняемого файла

### 4.1.1 Ответы на вопросы по программе

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

mov eax, rem call sprint

1. Инструкция mov ecx,x используется, чтобы включить 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, обеспечивающей ввод сообщения с клавиатуры.
2. Инструкция call atoi используется для вызова из внешнего файла подпрограммы, преобразующей ASCII-код символа в целое число и записывающей результат в регистр eax.
3. За вычисление варианта отвечают строки листинга:

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

1. При выполнении инструкции div ebx остаток от деления записывается в регистр edx.
2. Инструкция inc edx увеличивает значение регистра edx на 1.
3. За вывод на экран результата вычислений отвечают строки листинга:

mov eax, edx call iprintLF

## 4.2 Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm

Figure 22: Создание файла

Figure 22: Создание файла

Записываю в созданный файл текст программы для вычиления значения выражения под вариантом 5 (9\*x-8)/8

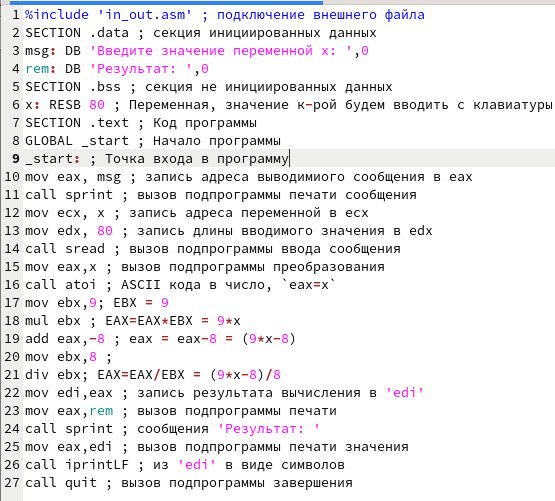


Figure 23: Редактирование файла

Создаю и запускаю исполняемый файл, ввожу значение 8

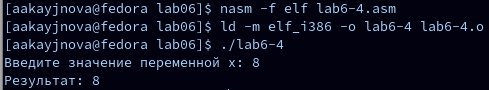


Figure 24: Запуск исполняемого файла

Ввожу другое значение x

Figure 25: Запуск исполняемого файла

Figure 25: Запуск исполняемого файла

# 5 Выводы

В ходе данной лабораторной работы мы научились работать с арифметическими инструкциями языка ассмеблера NASM.

# Список литературы

1. https://esystem.rudn.ru/pluginfile.php/2089086/mod\_resource/content/0/%D0%9B%D0%B0%D0%B1%D0%BE%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F%20%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%20%E2%84%966.%20%D0%90%D1%80%D0%B8%D1%84%D0%BC%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5%20%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%B8%20%D0%B2%20NASM..pdf