Отчёт по лабораторной работе №7

Дисциплина: архитектура компьютера

Тимофеева Екатерина Николаевна

Содержание

# 1 Цель работы

Цель работы заключается в освоении арифметических 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.

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. # Выполнение лабораторной работы

№1. Создаём директорию, в которой будем создавать файлы с программами для лабораторной работы №7 и переходим в созданный каталог. (рис. 1)

![Рис. 1: Cоздание директории и переход в каталог](data:application/octet-stream;base64,)

Рис. 1: Cоздание директории и переход в каталог

Создаём файл lab7-1.asm и открываем его. (рис. 2)

![Рис. 2: Создание файла](data:application/octet-stream;base64,)

Рис. 2: Создание файла

Вводим в файл текст программы из листинга. (рис. 3)

![Рис. 3: Ввод текста](data:application/octet-stream;base64,)

Рис. 3: Ввод текста

Создаём исполняемый файл программы и запускаем, программа вывела символ j. (рис. 4)

![Рис. 4: Создание файла](data:application/octet-stream;base64,)

Рис. 4: Создание файла

Далее изменяем текст программы, вместо стмволов “6” и “4” вводим 6 и 4. (рис. 5)

![Рис. 5: Замена текста программы](data:application/octet-stream;base64,)

Рис. 5: Замена текста программы

Создаём новый исполняемый файл программы и запускаем его. Вывелся символ с кодом 10, это символ перевода строки и он не отображается при выводе на экран. (рис. 6)

![Рис. 6: Запуск файла](data:application/octet-stream;base64,)

Рис. 6: Запуск файла

Создаём новый файл lab7-2.asm (рис. 7)

![Рис. 7: Создание файла](data:application/octet-stream;base64,)

Рис. 7: Создание файла

Вводим в файл текст другой программы, создаём и запускаем исполняемый файл, теперь вывело число 106. (рис. 8)

![Рис. 8: Создание и запуск файла](data:application/octet-stream;base64,)

Рис. 8: Создание и запуск файла

Заменяем в тексте программы символы, создаём и запускаем новый исполняемый файл, программа вывела 10. (рис.9)

![Рис. 9: Создание и запуск нового файла](data:application/octet-stream;base64,)

Рис. 9: Создание и запуск нового файла

Заменяем в тексте программы функцию iprintLF на iprint и запускаем новый файл, вывод не изменился.

№2. Создаём файл lab7-3.asm (рис. 10)

![Рис. 10: Создание файла](data:application/octet-stream;base64,)

Рис. 10: Создание файла

Вводим в созданный файл текст программы для вычисления значения выражения, создаём исполняемый файл и запускаем его. (рис. 11)

![Рис. 11: Создание и запуск файла](data:application/octet-stream;base64,)

Рис. 11: Создание и запуск файла

Изменяем программу, создаём и запускаем новый исполняемый файл. (рис. 12)

![Рис. 12: Создание и запуск нового файла](data:application/octet-stream;base64,)

Рис. 12: Создание и запуск нового файла

Создаём файл variant.asm (рис. 13)

![Рис. 13: Создание файла](data:application/octet-stream;base64,)

Рис. 13: Создание файла

Вводим в файл текст программы для вычисления варианта задания по номеру студенческого билета, создаём и запускаем исполняемый файл, вводим номер своего студенческого билета, вывело 7 вариант. (рис. 14)

![Рис. 14: Создние и запуск файла, ввод студенческого билета](data:application/octet-stream;base64,)

Рис. 14: Создние и запуск файла, ввод студенческого билета

# 4 Ответы на вопросы по программе

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

mov eax,rem call sprint

1. 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.
2. call atoi используется для вызова подпрограммы из внешнего файла, которая преобразует ascii-код символа в целое число и записывает результат в регистр eax.
3. 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

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 результатов вычислений отвечают строки:

mov eax,edx call iprintLF

# 5 Выполнение заданий для самостоятельной работы

Создаём файл для написания программы. (рис. 15)

![Рис. 15: Создание файла](data:application/octet-stream;base64,)

Рис. 15: Создание файла

Открываем созданный файл, вводим в него текст программы для вычисления выражения, создаём и запускаем исполняемый файл, выполняем проверки. (рис. 16)

![Рис. 16: Создание и запуск файла, проверка](data:application/octet-stream;base64,)

Рис. 16: Создание и запуск файла, проверка

Программа для вычисления выражения 5\*(x - 1)^2 (рис. 17)

![Рис. 17: Программа](data:application/octet-stream;base64,)

Рис. 17: Программа

# 6 Выводы

Мы освоили арифметические инструкции языка ассемблера NASM.

# Список литературы

1. Лабораторная работа №7.
2. Таблица ASCII