Отчёта по лабораторной работе №6

Архитектура компьютера

Еремина Оксана Андреевна НКАбд-02-23

Содержание

# Цель работы

Целью данной лабораторной работы является освоение арифметческих инструкций языка ассемблера NASM.

# Задание

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

# 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. 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 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.

# Выполнение лабораторной работы

1. Символьные и численные данные в NASM

Создаю каталог для программы данной лабораторной работы, и в нем создаю файл lab6-1.asm. И проверяю правильность выполнения команд.(рис.1)

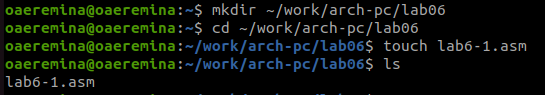


Figure 1: Создание папки и файла

Копирую в созданный каталог файл in\_out.asm, т.к. он будет использоваться далее. (рис.2)

Figure 2: Создание копии файла

Figure 2: Создание копии файла

Открываю созданный файл, вставляю в него программу вывода значения регистра eax. (рис.3)

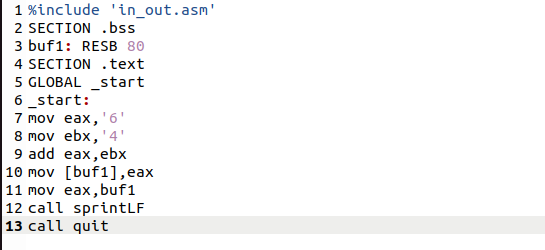


Figure 3: Содержимое файла lab6-1.asm

Создаю исполняемый файл и запускаю его (рис.4)

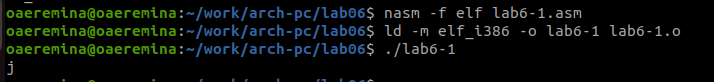


Figure 4: Запуск исполняемого файла

Изменяю в тексте программы символы ‘6’ и ‘4’ на 6 и 4 (рис.5)

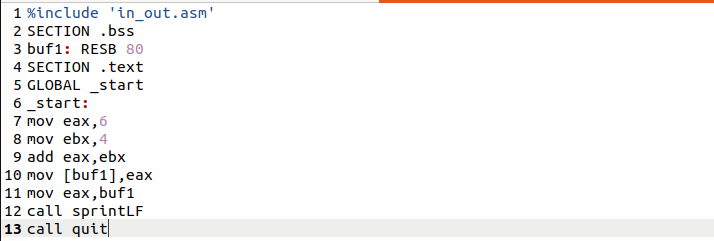


Figure 5: Редактирование файла lab6-1.asm

Создаю исполняемый файл программы и запускаю его. Вывелся символ с кодом 10 -символ перевода строки, он не отображается на экране при выводе (рис.6)

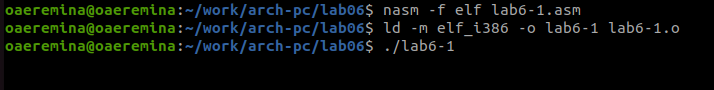


Figure 6: Запуск исполняемого файла

Создаю новый файл с помощью touch (рис.7)

Figure 7: Создание файла

Figure 7: Создание файла

Открываю созданный файл, вставляю в него текст другой программы для вывода значения регистра eax. (рис.8)

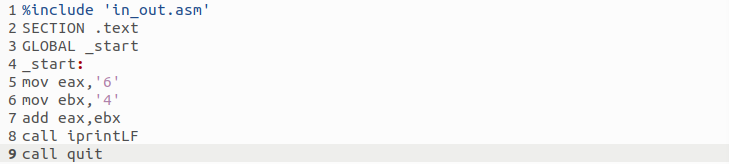


Figure 8: Содержимое файла lab6-2.asm

Создаю и запускаю исполняемый файл lab7-2 (рис.9). Теперь программа выводит число 106, потому что программа позволяет вывести именно число, а не символ, однако происходит именно сложение кодов символов “6” и “4”.

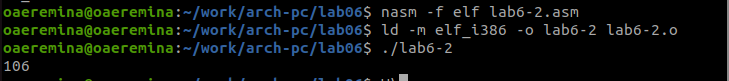


Figure 9: Запуск исполняемого файла

Изменяю в тексте программы символы ‘6’ и ‘4’ на 6 и 4 (рис.10)

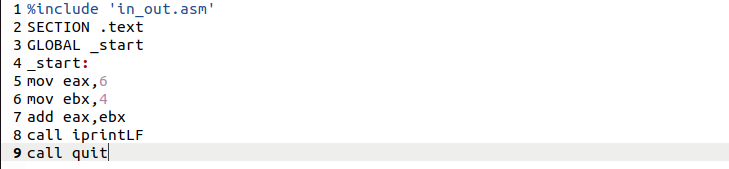


Figure 10: Редактирование файла lab6-2.asm

Создаю и запускаю исполняемый файл (рис.11)

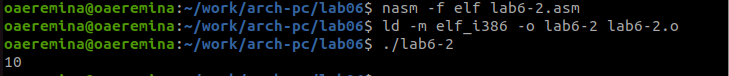


Figure 11: Запуск исполняемого файла

Заменяю в тексте программы функцию iprintLF на iprint (рис.12)

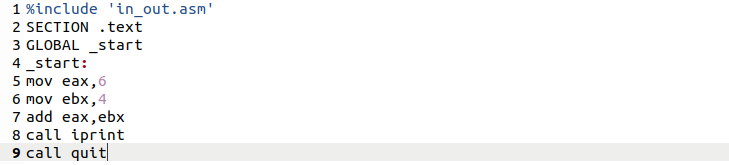


Figure 12: Редактирование файла lab6-2.asm

Создаю и запускаю новый исполняемый файл (рис.13).Вывод не изменится, 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.

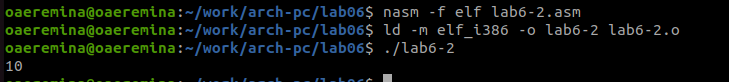


Figure 13: Запуск исполняемого файла

1. Выполнение арифметических операций в NASM

Создаю файл lab6-3.asm с помощью touch (рис.14)

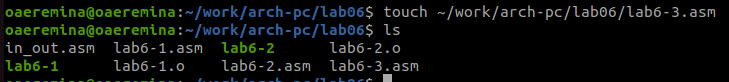


Figure 14: Создание файла

Ввожу в созданный файл текст программы для вычисления значения выражения f(x) = (5 \* 2 + 3)/3 (рис.15)

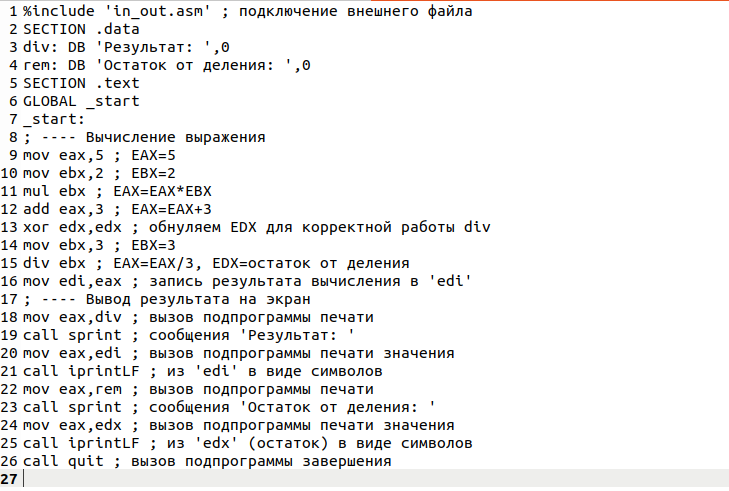


Figure 15: Содержимое файла lab6-3.asm

Создаю и запускаю новый исполняемый файл (рис.16)

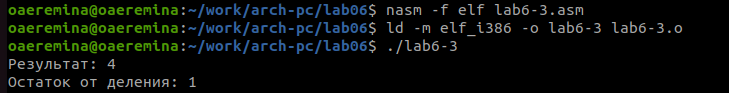


Figure 16: Запуск исполняемого файла

Изменяю текст программу для вычисления значения выражения f(x) = (4 \* 6 + 2)/5 (рис.17)

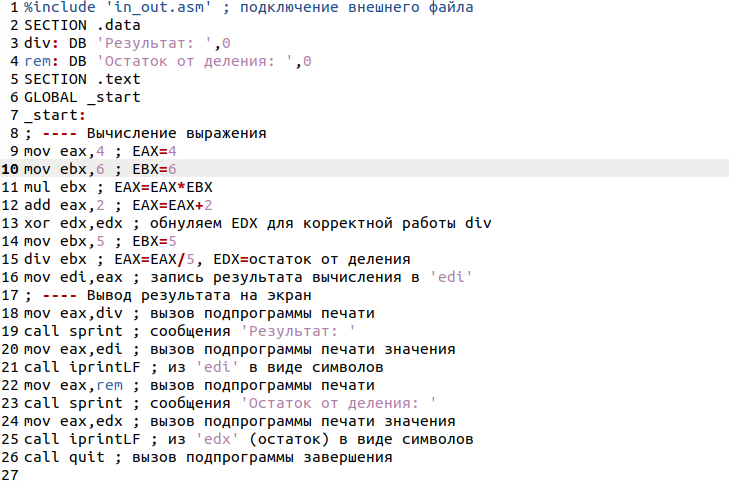


Figure 17: Содержимое файла lab6-3.asm

Создаю и запускаю новый исполняемый файл (рис.18)

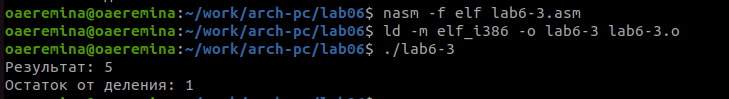


Figure 18: Запуск исполняемого файла

Создаю файл variant.asm с помощью touch (рис.19)

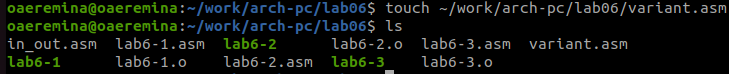


Figure 19: Создание файла

Ввожу в файл текст программы для вычисления варианта задания по номеру студенческого билета (рис.20)

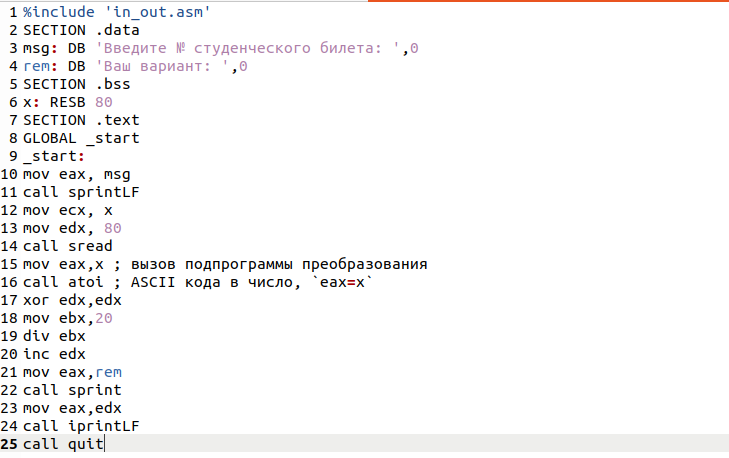


Figure 20: Редактирование файла variant.asm

Создаю и запускаю исполняемый файл (рис.21). Ввожу номер своего студенческого билета, программа вывела, что мой вариант - 17.

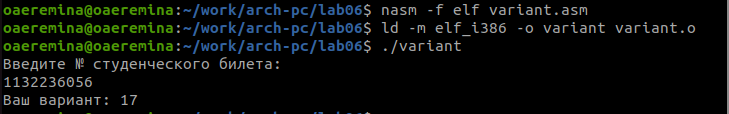


Figure 21: Запуск исполняемого файла

Ответы на вопросы:

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’?

За вывод сообщения “Ваш вариант” отвечают строки кода: mov eax,rem call sprint

1. Для чего используется следующие инструкции? mov ecx, x mov edx, 80 call sread

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

1. Для чего используется инструкция “call atoi”?

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

1. Какие строки листинга 6.4 отвечают за вычисления варианта?

За вычисления варианта отвечают строки: xor edx,edx mov ebx,20 div ebx inc edx

1. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”?

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

1. Для чего используется инструкция “inc edx”?

Инструкция inc edx увеличивает значение регистра edx на 1

1. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений?

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

1. Выполнение заданий для самостоятельной работы

Создаю файл lab6-4.asm с помощью touch. Открываю созданный файл, ввожу текст программы для вычисления значения выражения 18(𝑥 + 1)/6. x1 = 3, x2 = 1. Выражение под вариантном 17. (рис. 22)

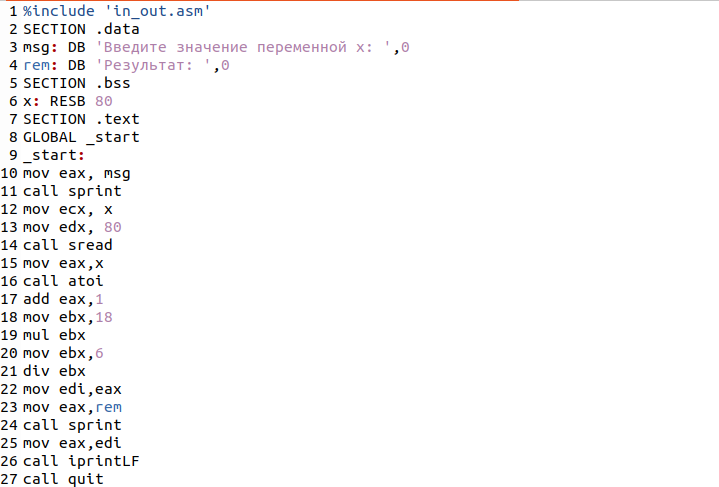


Figure 22: Программа

Создаю и запускаю исполняемый файл. x1 = 3, программа выводит 12. x2 = 1, вывод - 6, программа работает правильно. (рис. 23)

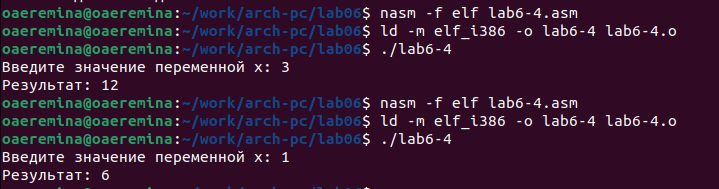


Figure 23: Выполнение программы

# Вывод

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

# Список литературы

Архитектура ЭВМ