Отчет по лабораторной работе №6

Дисциплина: архитектура компьютера

Ицков Андрей Станиславович

Содержание

# 1 Цель работы

Цель данной лабораторной работы - освоение арифметческих 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 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

## 4.1 Символьные и численные данные в NASM

С помощью утилиты mkdir создаю директорию, перемещаюсь туда и создаю там файл lab6-1.asm (рис. 1).

|  |
| --- |
| Создание директории и файла |

Рис. 1: Создание директории и файла

Вставляю в созданный файл данную мне программу (рис. 2).

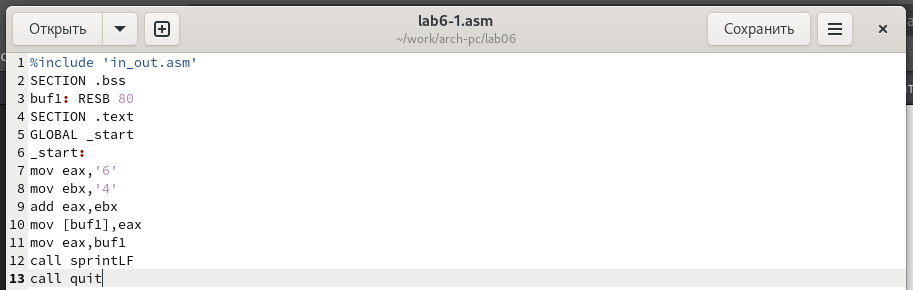


Рис. 2: Текст программы

Создаю исполняемый файл программы и запускаю его (рис. 3).

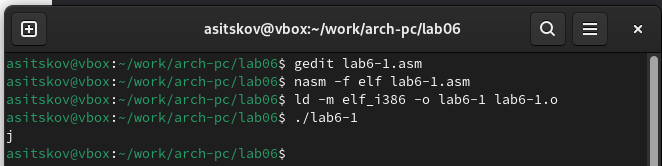


Рис. 3: Запуск исполняемого файла

Ввожу в текст программы запрашиваемые изменения (рис. 4).

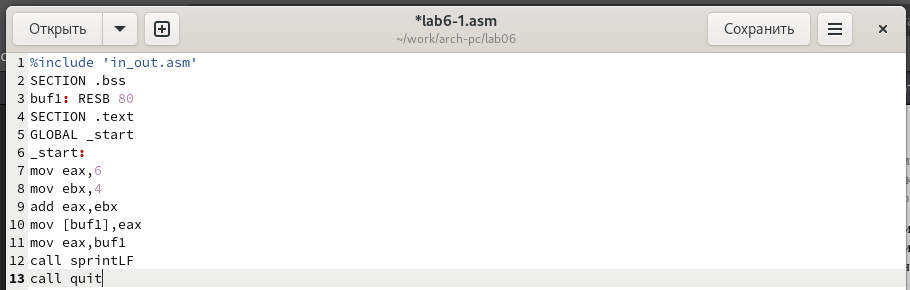


Рис. 4: Редактирование файла

Снова создаю исполняемый файл программы и запускаю его (рис. 5).

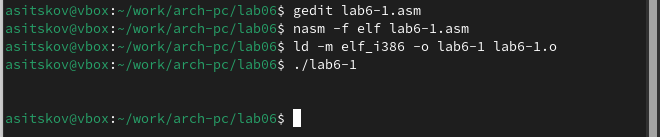


Рис. 5: Запуск исполняемого файла

Создаю новый файл lab6-2.asm и ввожу туда текст новой программы (рис. 6).

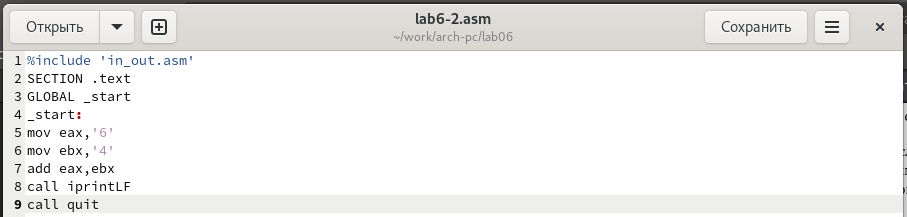


Рис. 6: Текст программы

Редактирую текст программы (рис. 7).

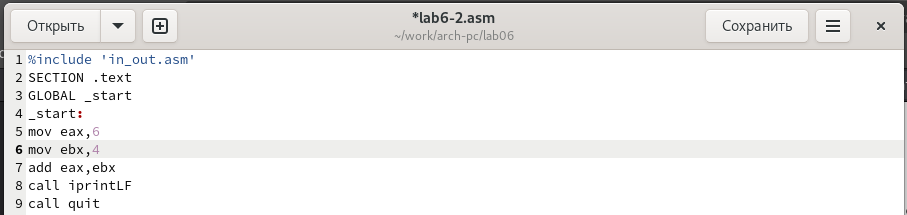


Рис. 7: Запуск исполняемого файла

Создаю исполняемый файл программы и запускаю его (рис. 8).

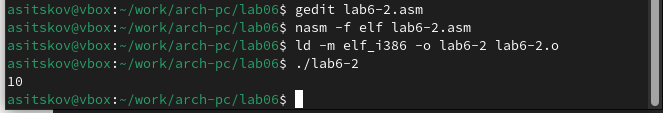


Рис. 8: Создание файла

Заменяю iprintLF на iprint в тексте программы (рис. 9).

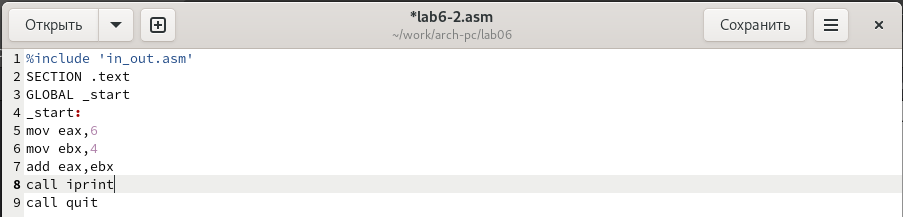


Рис. 9: Редактирование файла

Снова создаю и запускаю исполняемый файл lab6-2 (рис. 10).

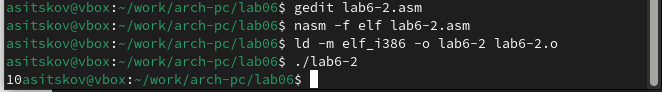


Рис. 10: Запуск исполняемого файла

## 4.2 Выполнение арифметических операций в NASM

Создаю пустой файл lab6-3.asm (рис. 11).

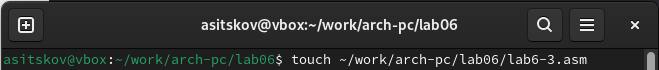


Рис. 11: Созданмие файла

Ввожу туда данную мне программу (рис. 12).

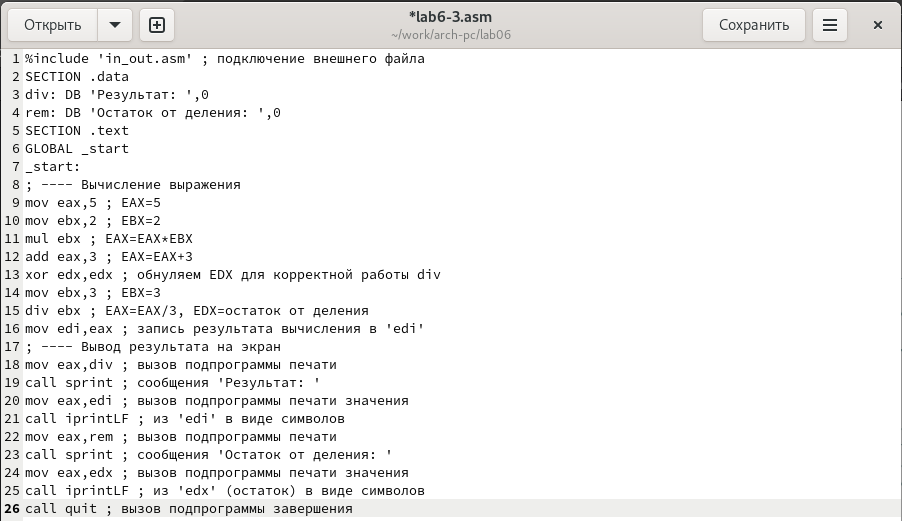


Рис. 12: Текст программы

Запускаю программы и получаю результат выражения (рис. 13).

|  |
| --- |
| Запуск исполняемого файла |

Рис. 13: Запуск исполняемого файла

Изменяю программу так, чтобы она вычисляла значение выражения f(x) = (4 \* 6 + 2)/5 (рис. 14).

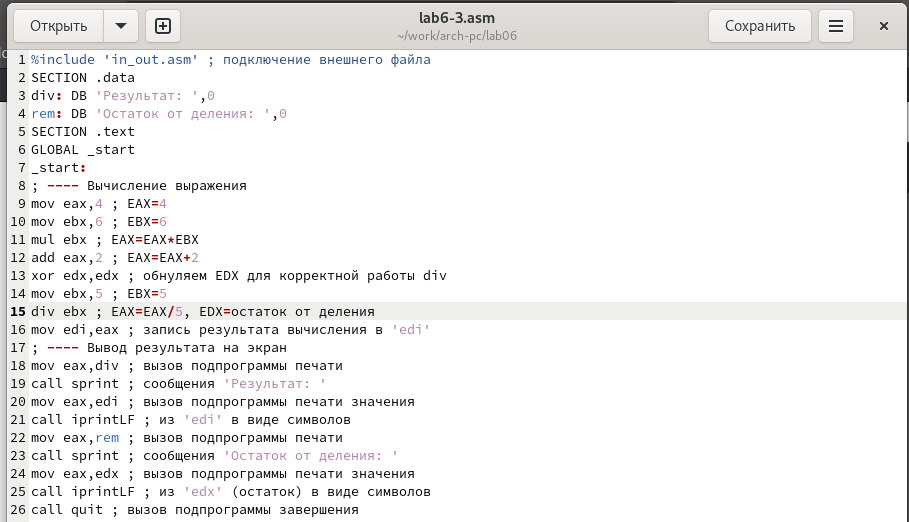


Рис. 14: Редактирование файла

Запускаю программы и получаю новый результат (рис. 15).



Рис. 15: Запуск файла

Создаю файл variant.asm и ввожу туда данную мне программу (рис. 16).



Рис. 16: Редактирование файла

Программа выдала мне номер моего варианта (рис. 17).

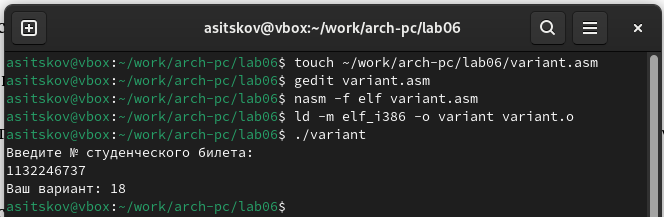


Рис. 17: Запуск исполняемого файла

## 4.3 Выполнение заданий для самостоятельной работы

Решаю задачу (рис. 18).



Рис. 18: Текст программы

Получаю ответы (рис. 19).

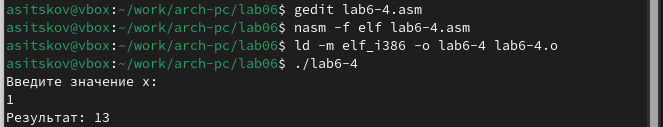


Рис. 19: Ответы

Создаю файл variant.asm с помощью утилиты touch (рис. 20).

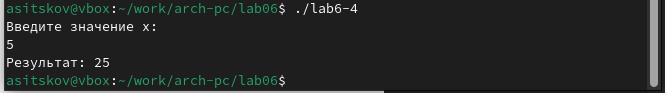


Рис. 20: Ответы

### 4.3.1 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 Выводы

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

# 6 Список литературы

1. [Лабораторная работа №7](https://esystem.rudn.ru/pluginfile.php/1584637/mod_resource/content/1/%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%967.pdf)
2. [Таблица ASCII](https://www.rapidtables.com/code/text/ascii-table.html)