Лабораторная работа №6

Арифметические операции в NASM

Замула Егор Сергеевич

Содержание

# Цель работы

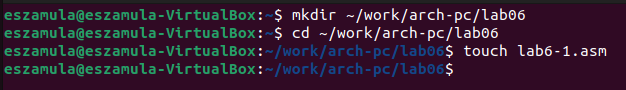
Освоение арифметических инструкций языка ассемблера NASM.

# Теоретическое введение

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.

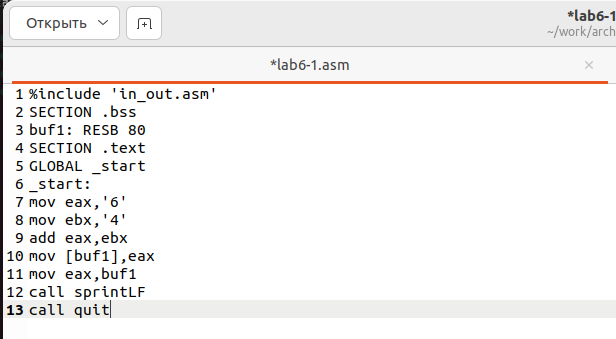
# Выполнение лабораторной работы

1. Создал каталог для программам лабораторной работы № 6, перешёл в него и создал файл lab6-1.asm



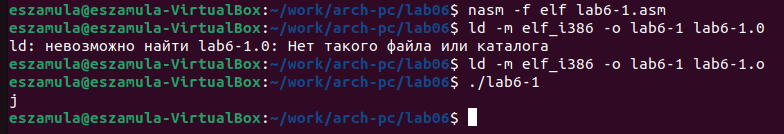
Создание каталога

1. Ввел в файл lab6-1.asm текст программы из листинга 6.1.



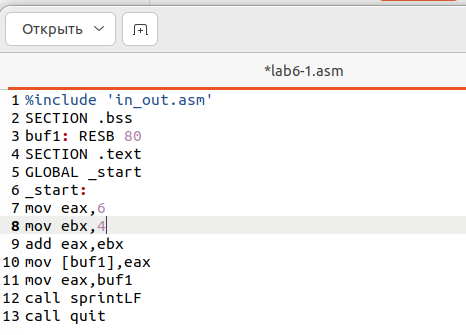
Ввод текста из листинга6.1

1. Создал исполняемый файл и запустил его.



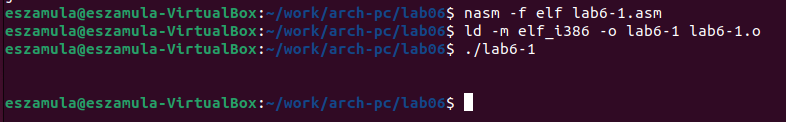
Запуск исполняемого файла

1. Далее изменил текст программы и вместо символов, записал в регистры числа.



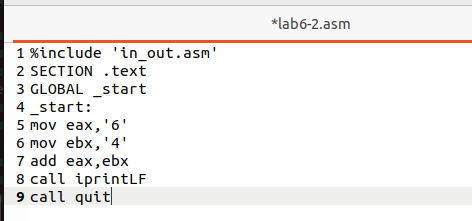
Изменение текста программы

1. Создал исполняемый файл и запустил его. Этот символ не будет отражаться



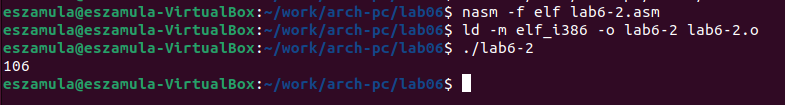
Запуск измененного исполняемого файла

1. Создал файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и ввел в него текст программы из листинга 6.2.



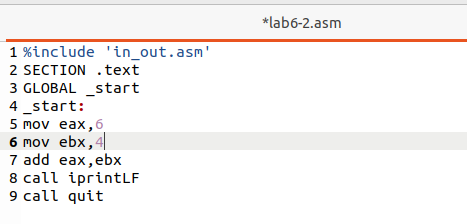
Создание файла lab6-2.asm

1. Создаю исполняемый файл и запускаю его. Получаем число 106.



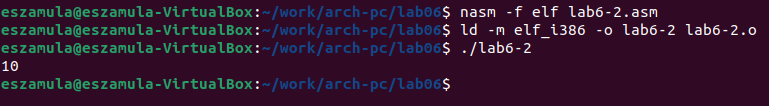
Запуск исполняемого файла

1. Аналогично предыдущему примеру изменим символы на числа.



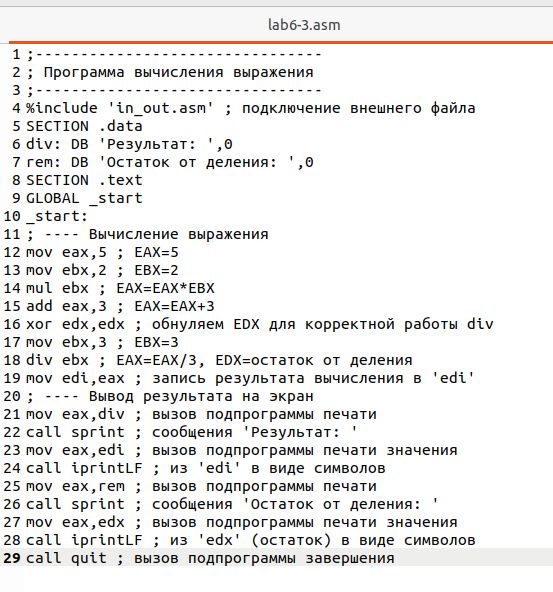
Изменяем файл lab6-2.asm

1. Создаю исполняемый файл и запускаю его. Получаем ответ 10.



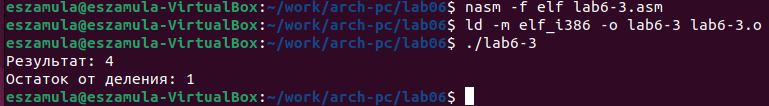
Запуск изменённого файла

1. Создал файл lab6-3.asm в каталоге ~/work/arch-pc/lab06, изучил текст программы из листинга 6.3 и ввел в lab6-3.asm.



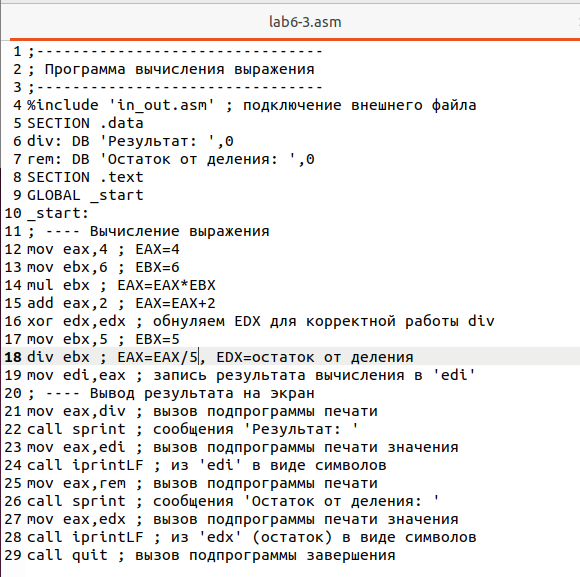
Создание файл lab6-3.asm

1. Создал исполняемый файл и запустил его. Результат должен получится 4, а остаток 1.



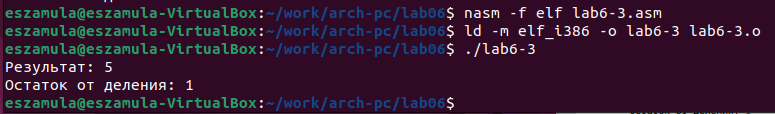
Запуск файла lab6-3.asm

1. Изменил текст программы для вычисления выражения.



Изменяю текст файла lab6-3.asm

1. Создаю исполняемый файл и проверяю его работу. Результат вычисления должен быть 5, и остаток 1.



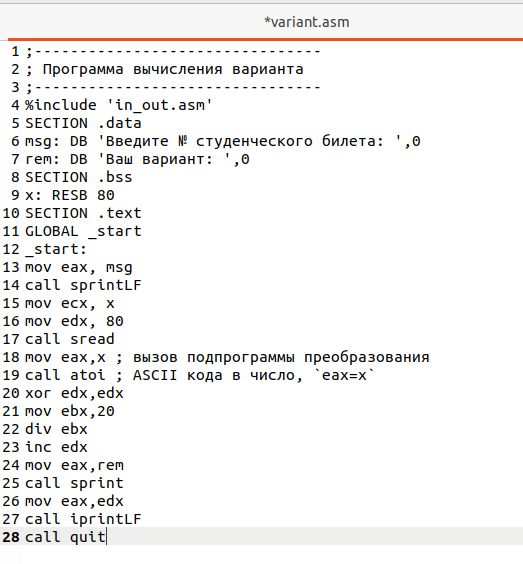
Запуск изменеённого файла

1. Создаю файл variant.asm в каталоге ~/work/arch-pc/lab06

Создаю файл variant.asm

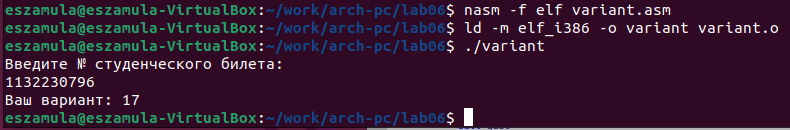
Создаю файл variant.asm

1. Внимательно изучил текст программы из листинга 6.4 и ввел в файл variant.asm.



Изменяю файл variant.asm

1. Создал исполняемый файл и запустил его. Проверил результат работы программы вычислив номер варианта аналитически.Мой вариант 17.



Создаю файл variant.asm

1. Ответы на вопросы:
2. mov eax, rem call sprint
3. mov eax, х - используется для записи входной переменной в регистр ecx; mov edx, 80 - запись размера переменной в регистр edx; call sread - вызов процедуры чтения данных
4. это функция преобразующая ascii-код символа в целое число и записывающий результат в регистр eax.
5. xor edx, edx mov, 20 div ebx inc edx
6. B ebx
7. Инструкция inc используется для увеличения операнда на еденицу
8. mov eax, rem call sprint mov eax, edx call iprintLF

# Выводы

Я освоил арифметические инструкции языка ассемблера NASM.

# Список литературы