Лабораторная работа №6

Архитектура компьютера

Кучмар София Игоревна

Содержание

# 1 Цель работы

Эта работа направлена на освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

Данная работа посвящена практическому освоению ассемблера NASM. Будут изучены основы работы адресацией в NASM, освоены арифметические операции в NASM, целочисленное сложение add, целочисленное вычитание sub, команды инкремента и декремента, команда изменения знака операнда neg, основными директивами ассемблера, команды умножения mul и imul и будет написана программу для вычисления выражений. Будет подключен внешний файл in\_out.asm с функциями ввода и вывода данных.

# 3 Выполнение лабораторной работы

Создадим каталог для программам лабораторной работы № 6, перейдём в него и создадим файл lab6-1.asm(рис. 1).

Рис. 1: Создание каталога и файла в нём

Рис. 1: Создание каталога и файла в нём

Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax (рис. 2).

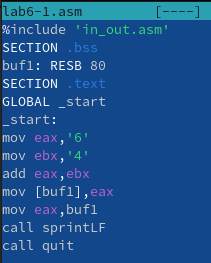


Рис. 2: Вводим программу

Создадим исполняемый файл и запустим его (рис. 3).

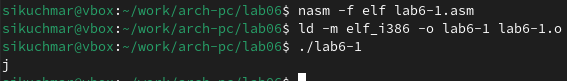


Рис. 3: Запуск файла

Далее изменим текст программы и вместо символов, запишем в регистры числа. Исправим текст программы (рис. 4).



Рис. 4: Изменение текст программы

Создадим исполняемый файл и запустим его (рис. 5)

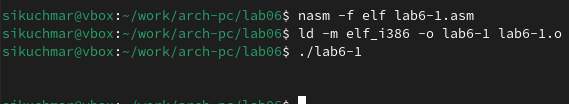


Рис. 5: Запуск файла

Создадим файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 и введём в него текст программы. Создадим исполняемый файл и запустим его (рис. 6).

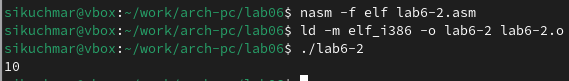


Рис. 6: Запуск файла

Аналогично предыдущему примеру изменим символы на числа. Создадим исполняемый файл и запустим его (рис. 7).

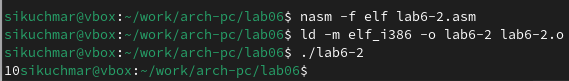


Рис. 7: Запуск файла

Создадим файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 и введём в него текст программы (рис. 8).

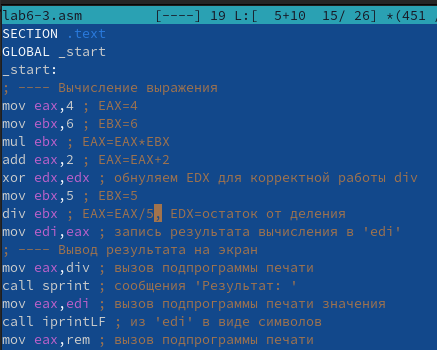


Рис. 8: Изменение файла lab5-3.asm

Создадим исполняемый файл и запустим его (рис. 9).

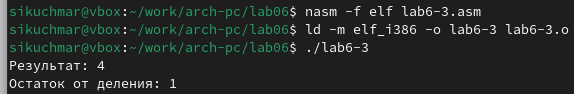


Рис. 9: Запуск файла

Изменим текст программы для вычисления выражения f(x)=(4∗6+2)/5. Создадим исполняемый файл и запустим его (рис. 10).

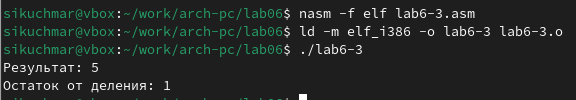


Рис. 10: Запуск файла

Создадим файл variant.asm в каталоге ~/work/arch-pc/lab06 и введём в него текст программы/ Создадим исполняемый файл и запустим его (рис. 11).

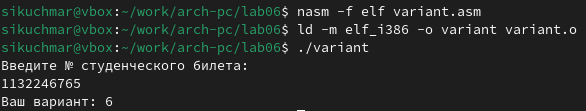


Рис. 11: Запуск файла

Заметим, что 1. Строки отвечающие за вывод на экран сообщения ’Ваш вариант: mov eax, rem call sprintLF 2. Инструкции mov ecx, x, mov edx, 80, call sread используются для чтения строки (в данном случае номера студенческого билета) из стандартного ввода. ecx указывает на буфер x, edx — на количество байтов для чтения, а sread считывает строку из стандартного ввода и сохраняет ее в буфере. 3. Инструкция “call atoi” используется для преобразования строки, хранящейся в x, в целое число. Преобразованное число сохраняется в регистре eax. 4. Строки, отвечающие за вычисления варианта: xor edx,edx mov ebx,20 div ebx inc edx 5. В регистр edx записывается остаток от деления при выполнении инструкции “div ebx” 6. Инструкция “inc edx” используется для увеличения остатка от деления на 1. Это необходимо, так как номера вариантов нумеруются с 1, а деление на 20 дает остаток от 0 до 19. 7. Строки, отвечающие за вывод на экран результата вычислений: mov eax,rem call sprint mov eax,edx call iprintLF

Создадим файл lab5-4.asm в каталоге ~/work/arch-pc/lab06 и введём в него текст программы (рис. 12).

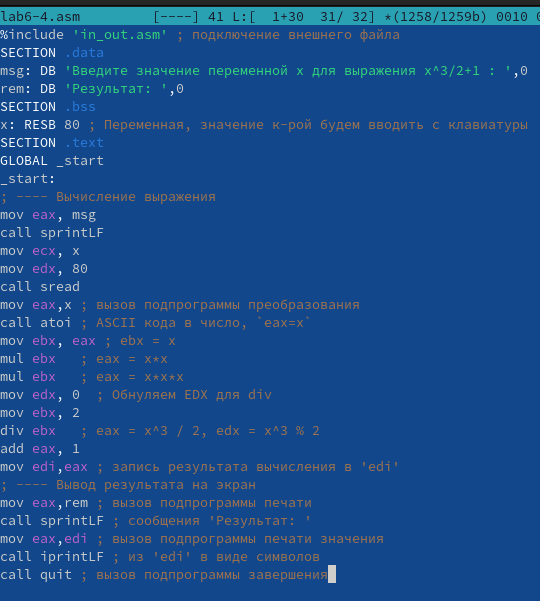


Рис. 12: Изменение файла lab5-4.asm

Создадим исполняемый файл и запустим его (рис. 13).

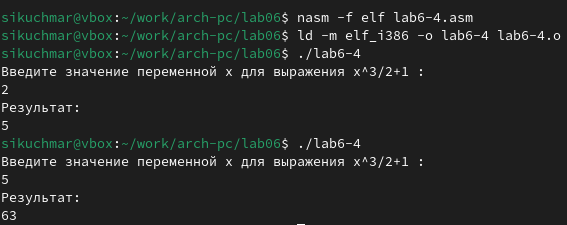


Рис. 13: Запуск файла

# 4 Выводы

В рамках данной работы были успешно освоены основы работы с ассемблером NASM. Были освоены арифметические операции в NASM и создание программ для вычисления выражений.