Лабораторная работа №6

Работа с NASM

Ромицына Анасасия Романовна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM # Задание

Написпть собственную прорамму.

# 2 Выполнение лабораторной работы

Создаем каталог для программам лабораторной работы No 6, переходим в него и создаем файл lab6-1.asm: (рис. [1](#fig:001)).

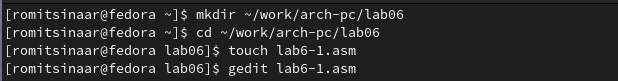


Figure 1: Создаем каталог и переходим в него

Рассмотрим примеры программ вывода символьных и численных значений. Программы будут выводить значения записанные в регистр eax. Вводим в файл lab6-1.asm текст программы из листинга 6.1.(рис. [2](#fig:002)).

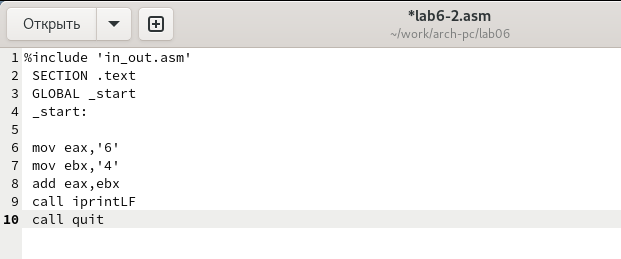


Figure 2: Программа из листнинга

Создайте исполняемый файл и запустите его.(рис. [3](#fig:003)).

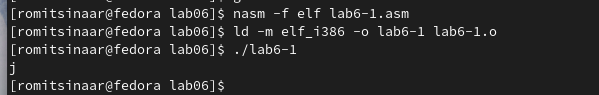


Figure 3: Запуск программы

Далее изменим текст программы и вместо символов, запишем в регистры числа. Ис- правим текст программы (Листинг 6.1) следующим образом: (рис. [4](#fig:004)).

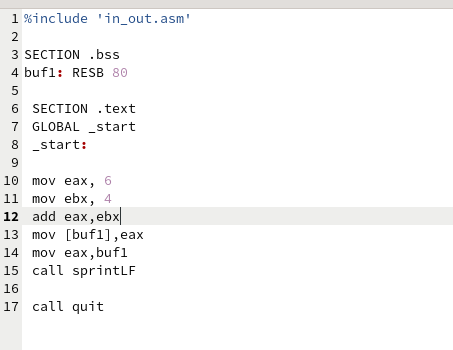


Figure 4: Программа из листнинга

Создайте исполняемый файл и запустите его (рис. [5](#fig:005)).

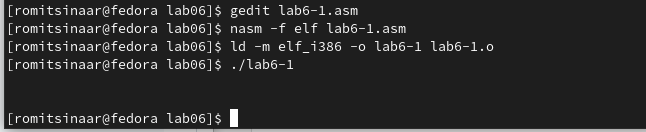


Figure 5: Запуск программы

Согласно таблице код 10 соответствует переносу на след строку, это программа и выводит.

Создадим файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. [6](#fig:006)).

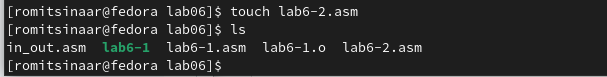


Figure 6: Создание файла

Введем в него текст программы из листинга 6.2 (рис. [7](#fig:007)).

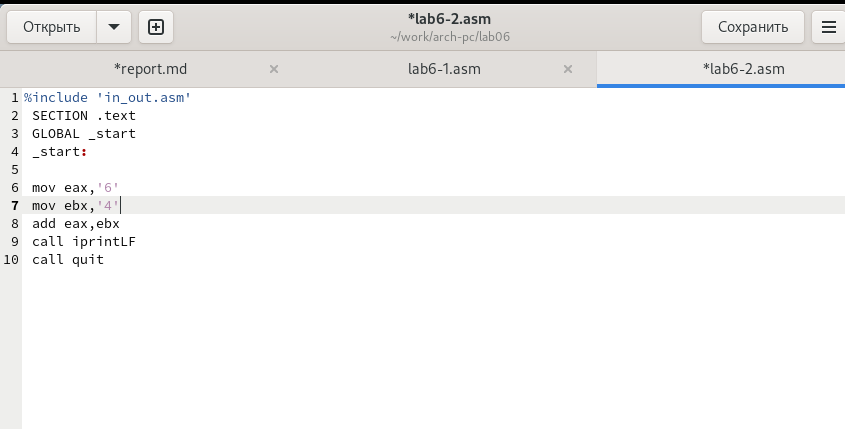


Figure 7: Программа из листнинга

Создадим исполняемый файл и запустим его.(рис. [8](#fig:008)).

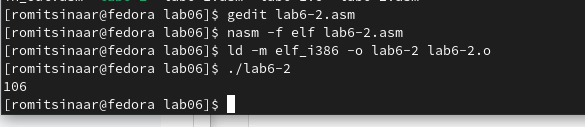


Figure 8: Запуск программы

Аналогично предыдущему примеру изменим символы на числа.(рис. [9](#fig:009)).

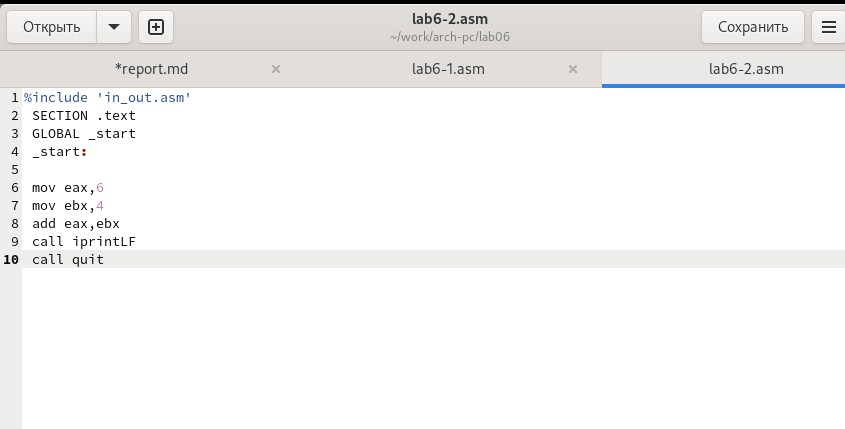


Figure 9: Изменние текста программы

Создадим исполняемый файл и запустим его.(рис. [10](#fig:010)).

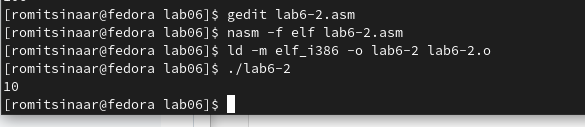


Figure 10: Запуск программы

Создадим файл lab6-3.asm в каталоге ~/work/arch-pc/lab06 (рис. [11](#fig:011)).

Figure 11: Создание нового файла

Figure 11: Создание нового файла

Внимательно изучаем текст программы из листинга 6.3 и вводим в lab6-3.asm.(рис. [12](#fig:012)).

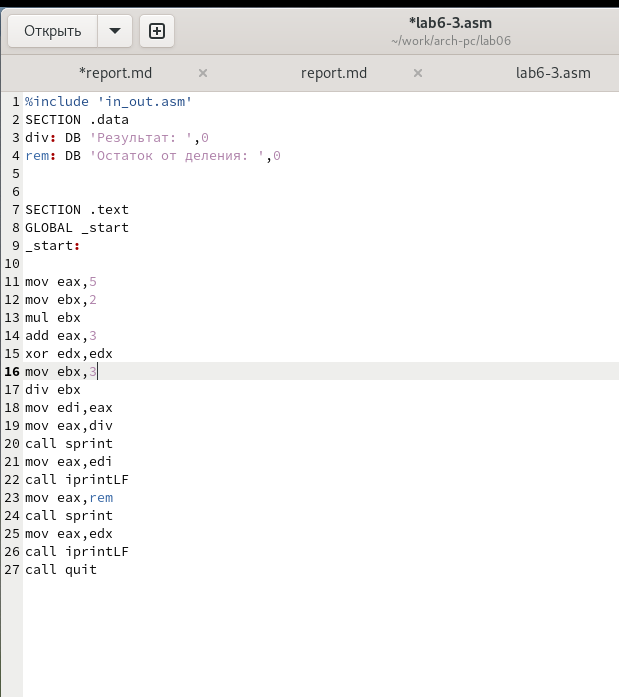


Figure 12: Изменние текста программы

Создаем исполняемый файл и запускаем его. Результат работы программы должен быть следующим:(рис. [13](#fig:013)).



Figure 13: Запуск программы

Изменяем текст программы для вычисления выражения 𝑓(𝑥) = (4 ∗ 6 + 2)/5. (рис. [14](#fig:014)).

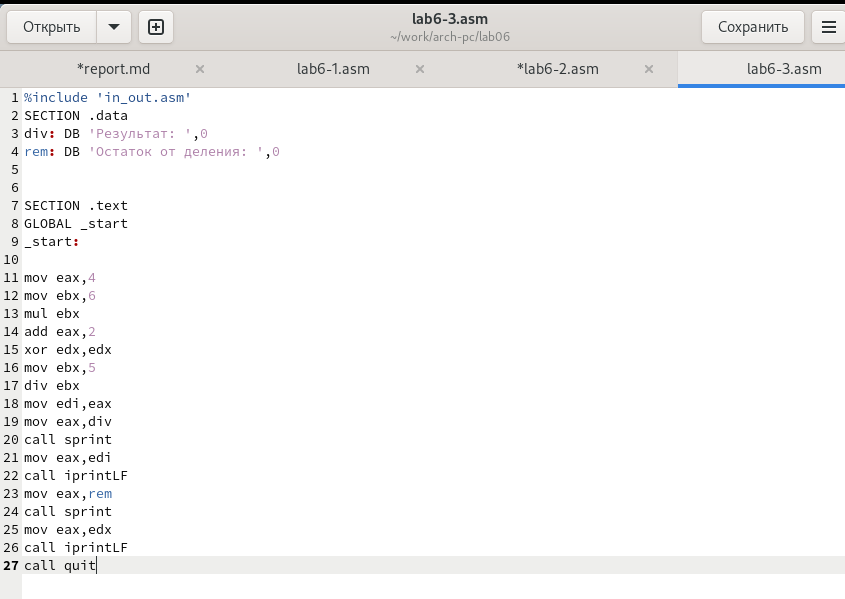


Figure 14: Редактируем файл

Создаем исполняемый файл и проверяем его работу. (рис. [15](#fig:015)).

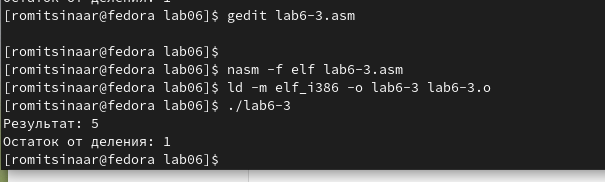


Figure 15: Запуск программы

Создаем файл variant.asm в каталоге ~/work/arch-pc/lab06: рис. [16](#fig:016)).

Figure 16: Создание файла

Figure 16: Создание файла

Внимательно изучаем текст программы из листинга 6.4 и вводим в файл variant.asm.(рис. [17](#fig:017)).

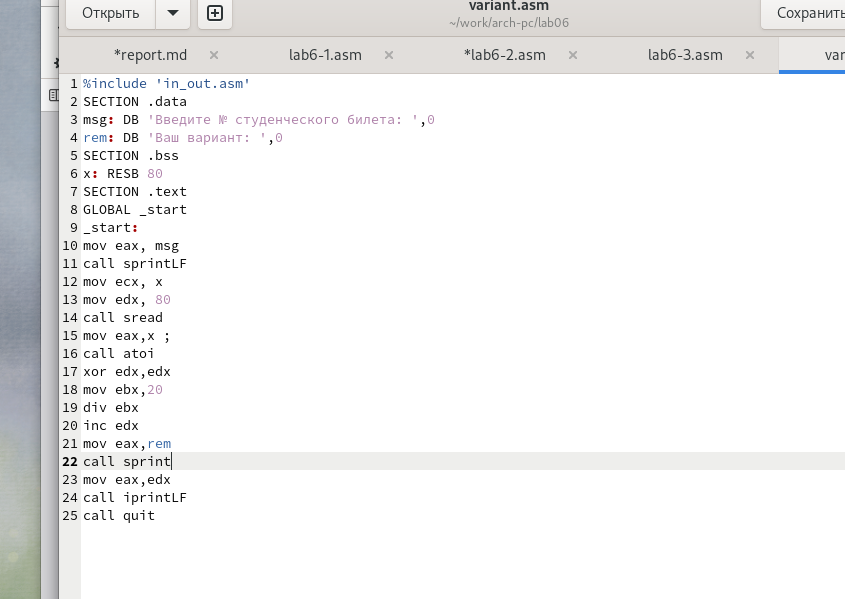


Figure 17: Заполнение файла

Создаем исполняемый файл и запускаем его. (рис. [18](#fig:018)).

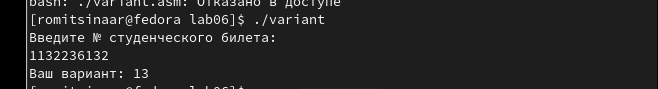


Figure 18: Запуск программы

Ответы на вопросы

1 Строка “mov eax,rem” и строка “call sprint” отвечают за вывод на экран сообщения ‘Ваш вариант:’. 2 Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре ecx, а количество символов в строке (максимальное количество символов, которое может быть считано) сохраняется в регистре edx. Затем вызывается процедура sread, которая выполняет чтение строки. 3 Инструкция “call atoi” используется для преобразования строки в целое число. Она принимает адрес строки в регистре eax и возвращает полученное число в регистре eax. 4 Строка “xor edx,edx” обнуляет регистр edx перед выполнением деления. Строка “mov ebx,20” загружает значение 20 в регистр ebx. Строка “div ebx” выполняет деление регистра eax на значение регистра ebx с сохранением частного в регистре eax и остатка в регистре edx. 5 Остаток от деления записывается в регистр edx. 6 Инструкция “inc edx” используется для увеличения значения в регистре edx на 1. В данном случае, она увеличивает остаток от деления на 1. 7 Строка “mov eax,edx” передает значение остатка от деления в регистр eax. Строка “call iprintLF” вызывает процедуру iprintLF для вывода значения на экран вместе с переводом строки.

# 3 Задания для самостоятельной работы

Создадим файл для выполнения самостоятельной работы . (рис. [19](#fig:019)).

Figure 19: Создание нового файла

Figure 19: Создание нового файла

Откроем новый файл в редакторе. (рис. [20](#fig:020)).

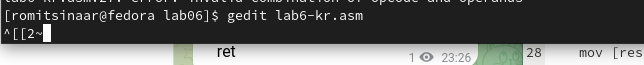


Figure 20: Открытие файла

Напишем программу для выполнения задания. (рис. [21](#fig:021)).

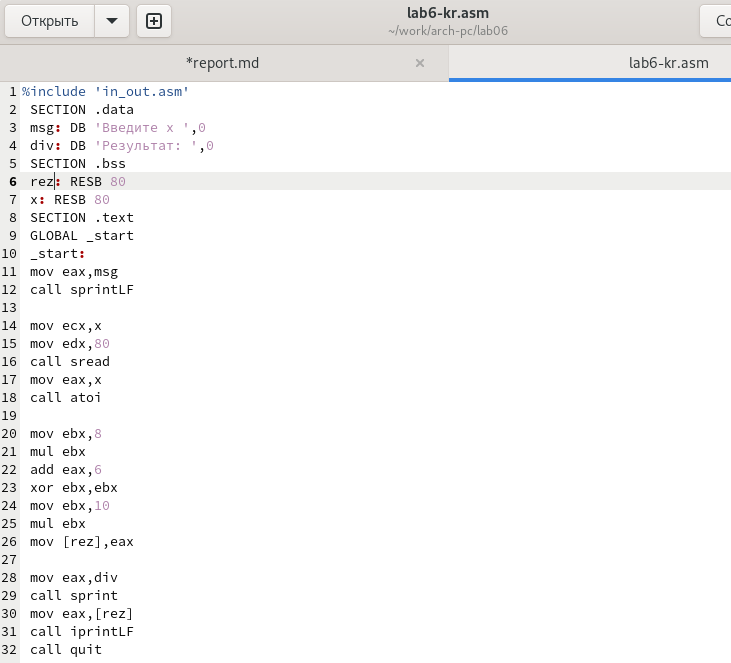


Figure 21: Содание новой программы

Запустим программу и проверим правильность её работы. Все работает верно. (рис. [22](#fig:022)).

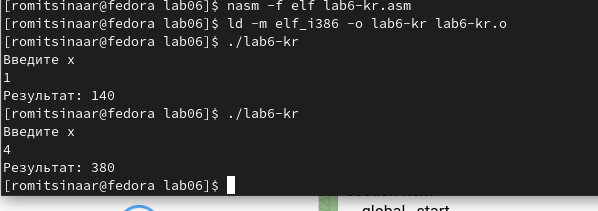


Figure 22: Запуск и проверка

# 4 Выводы

Мы освоили арифметические инструкции языка ассемблера NASM.