Лабораторная работа №6. Арифметические операции в NASM.

Аджигалиева Амина Руслановна

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Порядок выполнения лабораторной работы

## 2.1 Символьные и численные данные в NASM

Создадим каталог для программ лабораторной работы № 6, перейдите в него и создайте файл lab6-1.asm: (рис. 1).

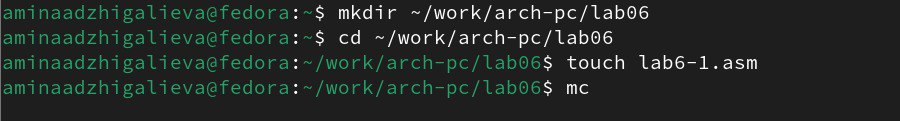


Рис. 1: Создание lab6-1.asm

Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1 (рис. 2).

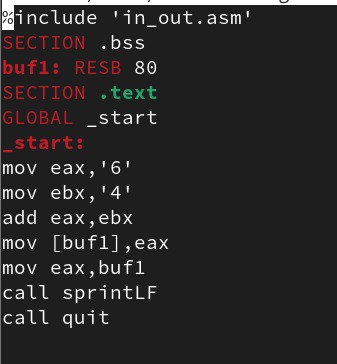


Рис. 2: Заполняем файл

Создаем исполняемый файл и запускаем его (рис. 3).

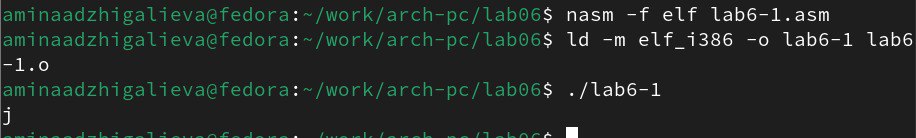


Рис. 3: Запуск программы

Изменим текст программы и вместо символов, запишем в регистры числа. Исправим текст программы (Листинг 6.1): заменим строки (рис. 4).

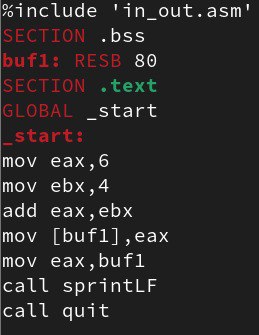


Рис. 4: Замена текста

Создаем исполняемый файл и запускаем его (рис. 5).

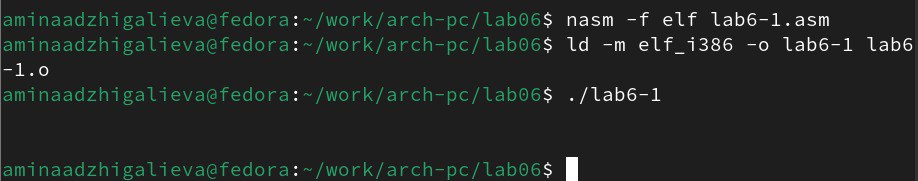


Рис. 5: Запуск программы

Создаем файл lab6-2.asm в каталоге ~/work/arch-pc/lab06 (рис. 6).



Рис. 6: Файл lab6-2.asm

Введем в него текст программы из листинга 6.2 (рис. 7).

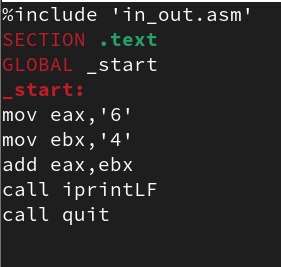


Рис. 7: Ввод текста

Создаем исполняемый файл и запускаем его (рис. 8).

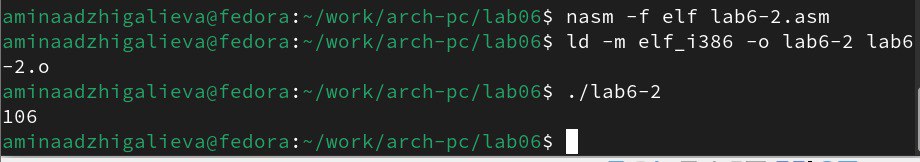


Рис. 8: Запуск программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений (рис. 9).

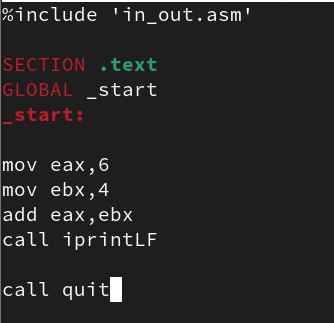


Рис. 9: Замена текста

Создаем исполняемый файл и запускаем его (рис. 10).

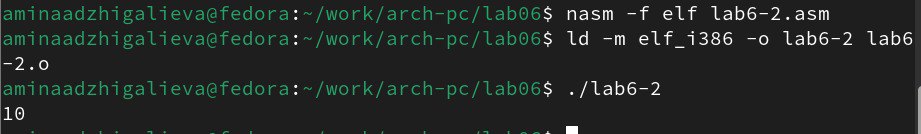


Рис. 10: Запуск программы

Заменим функцию iprintLF на iprint (рис. 11).

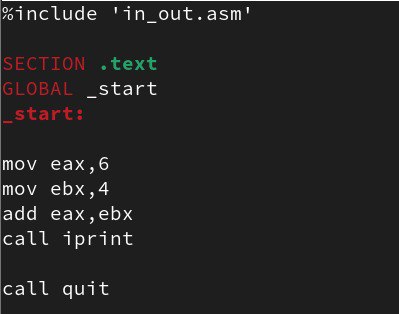


Рис. 11: Функция iprint

Создаем исполняемый файл и запускаем его (рис. 12).

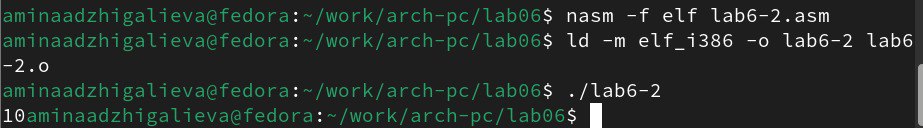


Рис. 12: Запуск программы

Функции iprintLF и iprint отличаются только тем, что LF переносит на новую строку.

## 2.2 Выполнение арифметических операций в NASM

Создаем файл lab6-3.asm в каталоге ~/work/arch-pc/lab06: (рис. 13).

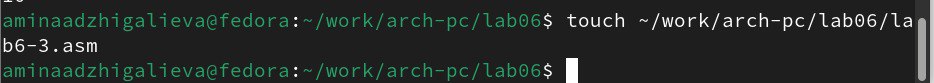


Рис. 13: Создание файла

Изучим текст программы из листинга 6.3 и введем в lab6-3.asm (рис. 14).

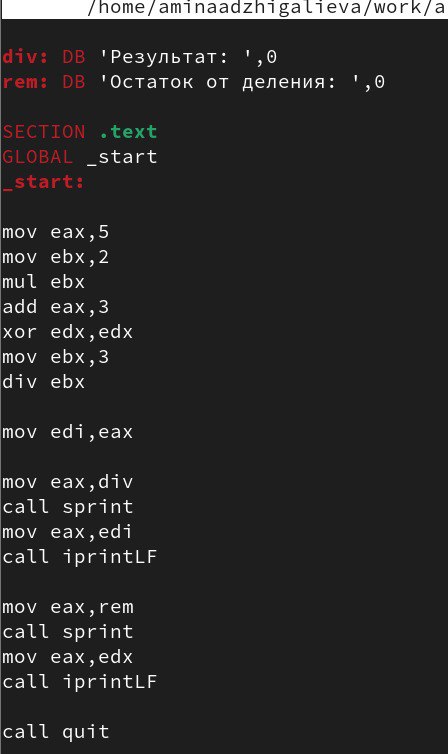


Рис. 14: Ввод программы

Создаем исполняемый файл и запускаем его (рис. 15).

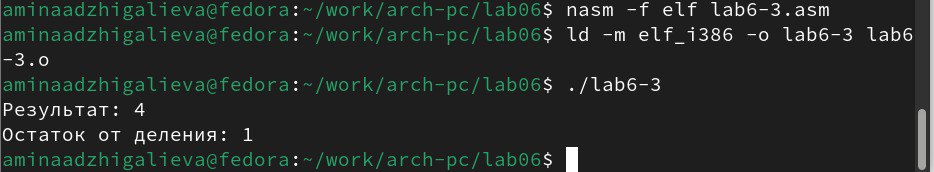


Рис. 15: Запуск программы

Изменим текст программы для вычисления выражения у = (4 \* 6 + 2)/5 (рис. 16).

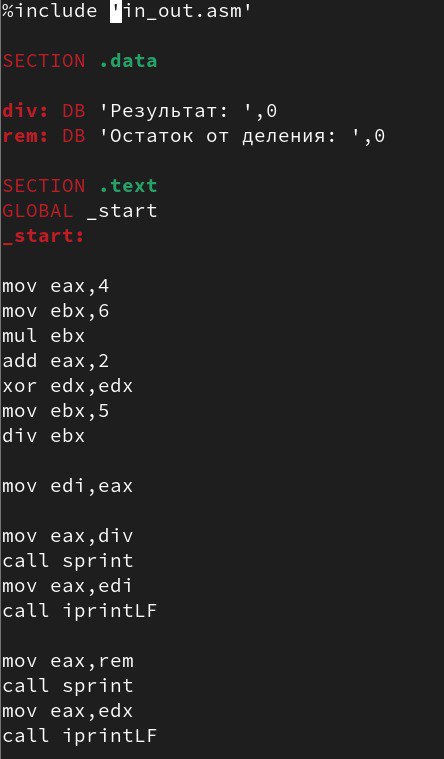


Рис. 16: Замена текста

Создаем исполняемый файл и запускаем его (рис. 17).

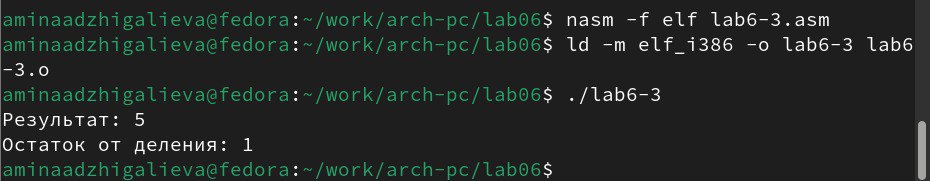


Рис. 17: Запуск программы

Создаем файл variant.asm в каталоге ~/work/arch-pc/lab06 (рис. 18).

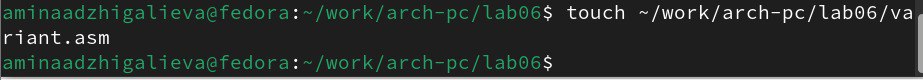


Рис. 18: Файл variant.asm

Изучим текст программы из листинга 6.4 и введем в файл variant.asm (рис. 19).

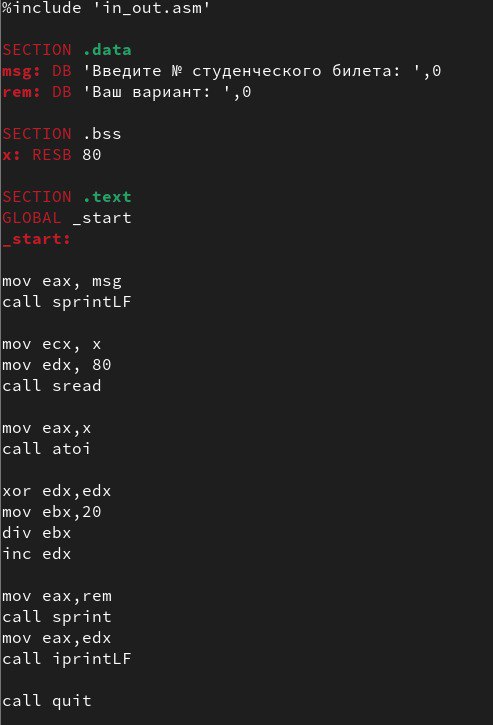


Рис. 19: Программа со студ.билетом

Создаем исполняемый файл и запускаем его (рис. 20).

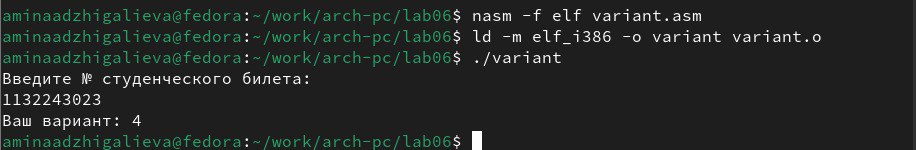


Рис. 20: Запуск программы

## 2.3 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.

## 2.4 Задание для самостоятельной работы

Открываем файл и заполняем, чтобы решалось выражение у = 4/3(х − 1)+5 (рис. 21).

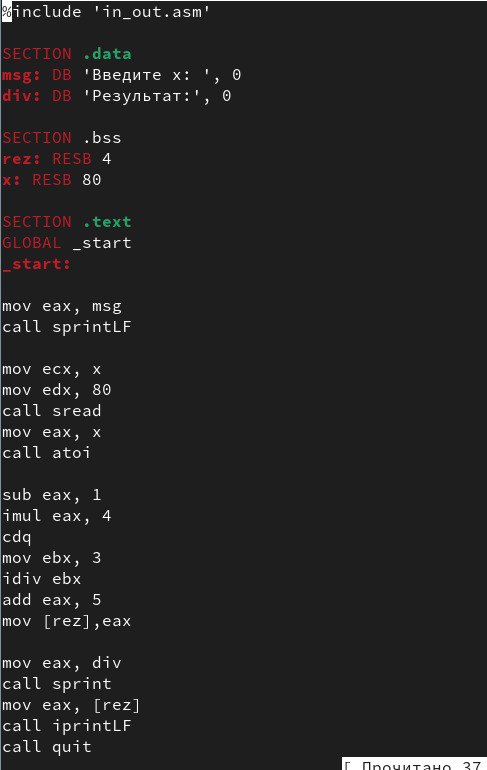


Рис. 21: Решение уравнения на ассамблере

Создаем исполняемый файл и запускаем его для х = 4 (рис. 22).

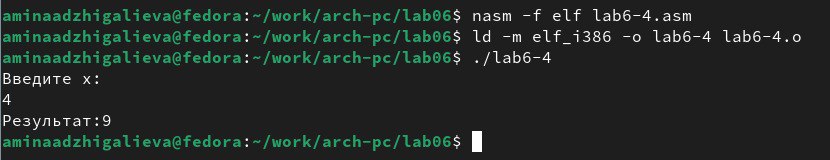


Рис. 22: Запуск программы

Теперь для х = 10 (рис. 23).

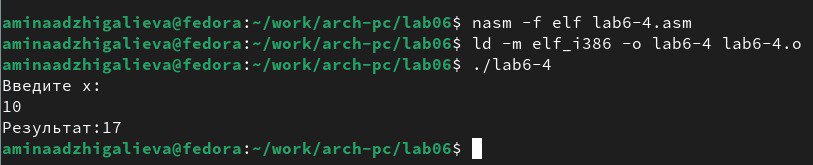


Рис. 23: Запуск программы

# 3 Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.