Лабораторная работа №6

Дисциплина: Архитектура компьютера

Комягин Андрей Николаевич

Содержание

# 1 Цель работы

Изучить арифметические инструкции языка ассемблера NASM и применить полученные знания на практике.

# 2 Выполнение лабораторной работы

## 2.1 Программа 6.1

Создадим каталог для программ лабораторных работ, в нем создадим файл **lab6-1.asm**. (рис. [1](#fig:001)).

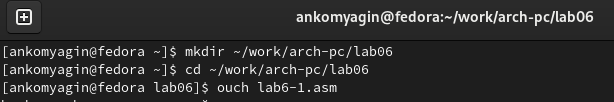


Figure 1: Создание каталога и файла ассемблера

Введем в созданный файл текст из листинга **6.1** (рис. [2](#fig:002)).

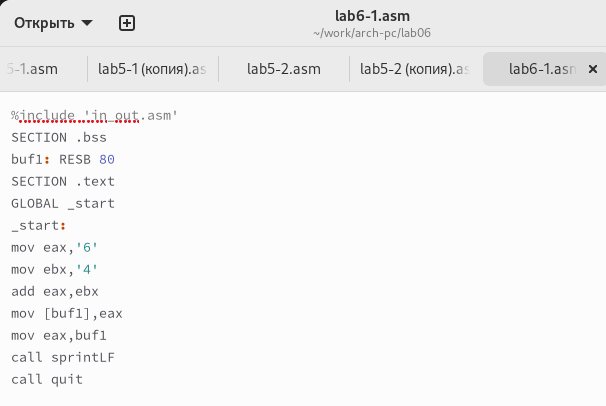


Figure 2: Ввод команлы

Создадим исполняемый файл и запустим его (рис. [3](#fig:003)). Результат его работы - вывод символа **j**.

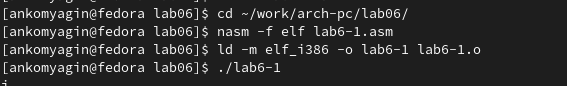


Figure 3: Запуск файла

Изменим текст программы (вместо символов запишем в регистры числа) и запустим её (рис. [4](#fig:004)). Программа выводит пустой символ (переноса строки).

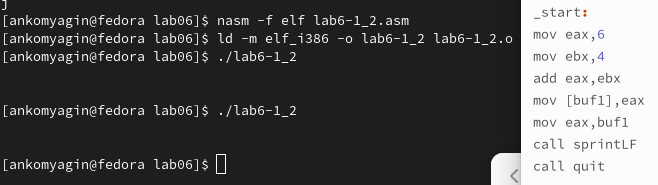


Figure 4: Запуск измененной программы

## 2.2 Программа 6.2

Создадим файл **lab6-2.asm** и введем в него текст программы из листинга **6.2** (рис. [5](#fig:005)).

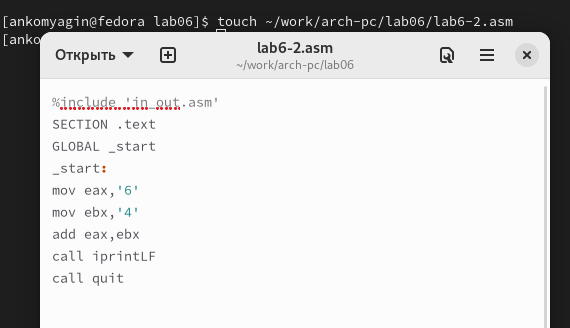


Figure 5: Создание программы 6.2

Создадим исполняемый файл и запустим его (рис. [6](#fig:006)). Результат работы - число 106

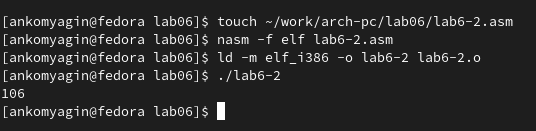


Figure 6: Запуск команды со строками

Заменим строки на числа, создадим исполняемый файл и запустим его (рис. [7](#fig:007)). Результат работы программы - 10.

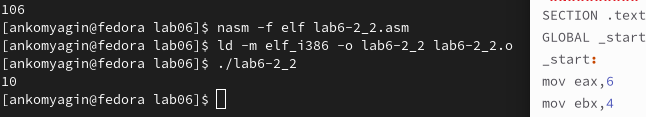


Figure 7: Запуск команды с числами

Заменим функцию **iprintLF** на **iprint**. Запустим файл. Разница работы программ и наличии и отсутствии переноса строки (рис. [8](#fig:008)).

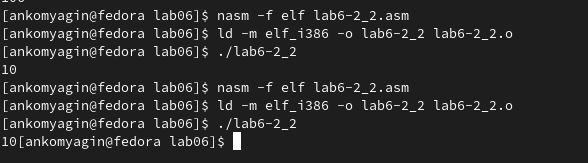


Figure 8: Сравнение команд вывода

## 2.3 Программа 6.3

Создадим файл **lab6-3.asm** и заполним его в соответствии с листингом **6.3**. Запустим файл, результат работы изображен на (рис. [9](#fig:009)).

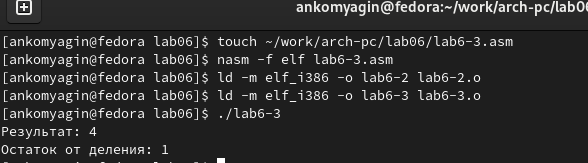


Figure 9: Создание и запуск файла

Изменим текст программы для вычисления выражения **𝑓(𝑥) = (4 ∗ 6 + 2)/5** и проверим работоспособность (рис. [10](#fig:010)).

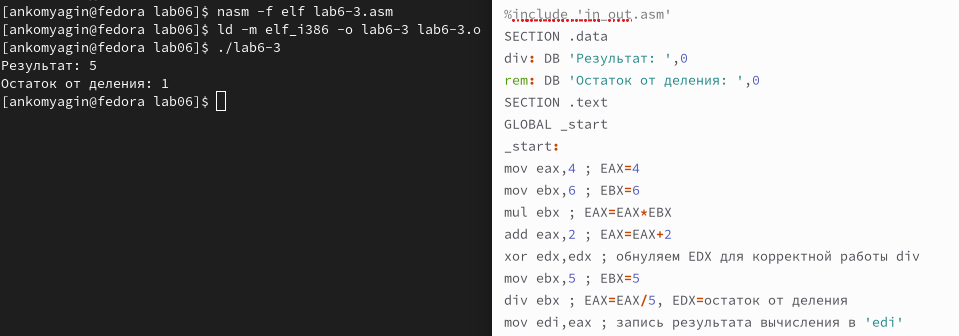


Figure 10: Изменение и запуск файла

## 2.4 Программа 6.4

Создадим файл **variant.asm**, заполним его в соответствии с листингом **6.4** (рис. [11](#fig:011)).

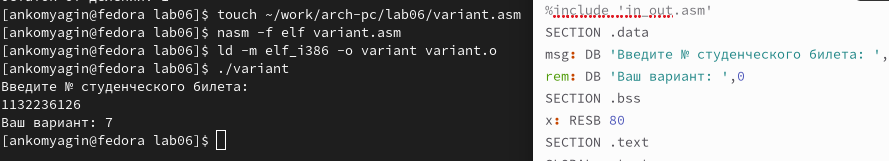


Figure 11: Запуск программы

Результат программы - 7. Проверим его аналитически. Для определения остатка деления на 20 достаточно смотреть на 2 последние цифры номера студенческого билета. В моем случае это 26. Остаток деления 26 на 20 = 6, добавляем единицу, которая нужна, чтобы не получился вариант 0, 6+1 = 7.

1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? **Строки mov eax,rem call sprint**
2. Для чего используется следующие инструкции? mov ecx, x mov edx, 80 call sread. **Для создания переменной х**
3. Для чего используется инструкция “call atoi”?. **Для преобразования кода символа в число**
4. Какие строки листинга 6.4 отвечают за вычисления варианта? **xor edx,edx mov ebx,20 div ebx inc edx**
5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? **в регистр ax**
6. Для чего используется инструкция “inc edx”? **Для добавления единицы**
7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений? **mov eax,edx call iprintLF**

## 2.5 Самостоятельная работа

Написать программу для вычисления значения функции.

**Вариант 7**. **f(x) = 5(x − 1)^2**. **x1 = 3, x2 = 5**

Программа (рис. [12](#fig:012)).

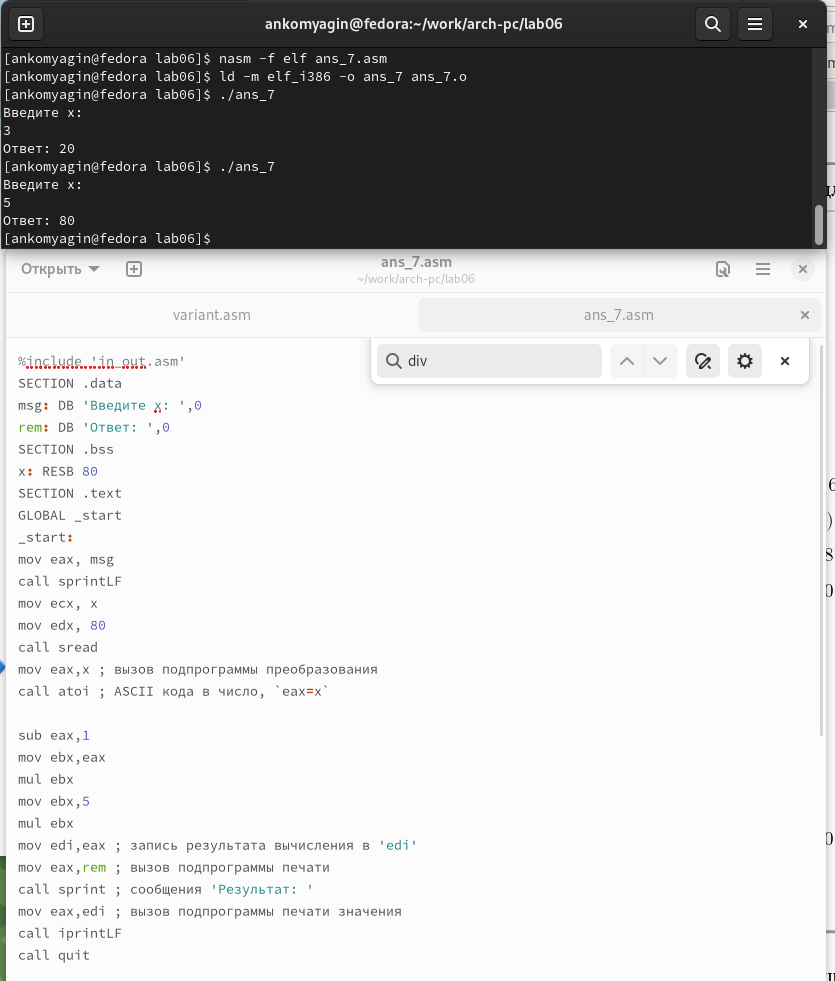


Figure 12: Запуск программы

# 3 Выводы

Я ходе работы я освоил арифметические конструкции языка ассемблера NASM. Изучил несколько программ и написал собственную.