Лабораторная работа №6

Арифметические операции в NASM

Бунин Арсений Викторович

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM

# 2 Задание

1. Создать файл на языке Ассемблер, выводящий значения регистра
2. Создать файл на языке Ассемблер, выполняющий арифметические операции
3. Выполнить индивидуальное задание по написанию программы на Ассемблере
4. Загрузить файлы на github

# 3 Теоретическое введение

Схема команды целочисленного сложения add (от англ. addition - добавление) выполняет сложение двух операндов и записывает результат по адресу первого операнда. Команда add работает как с числами со знаком, так и без знака. Команда целочисленного вычитания sub (от англ. subtraction – вычитание) работает аналогично команде add Довольно часто при написании программ встречается операция прибавления или вычитания единицы. Прибавление единицы называется инкрементом, а вычитание — декрементом. Для этих операций существуют специальные команды: inc (от англ. increment) и dec (от англ. decrement), которые увеличивают и уменьшают на 1 свой операнд. Умножение и деление, в отличии от сложения и вычитания, для знаковых и беззнаковых чисел производиться по-разному, поэтому существуют различные команды. Для беззнакового умножения используется команда mul. Для знакового умножения используется команда imul. Для деления, как и для умножения, существует 2 команды div и idiv.

# 4 Выполнение лабораторной работы

Создаем исполняемый файл(рис. [1](#fig:fig1) и рис. [2](#fig:fig2))

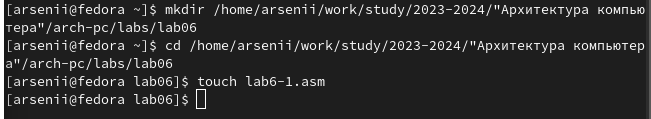


Figure 1: Вызов Midnight Commander

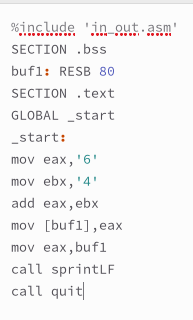


Figure 2: Каталог курса в Midnight Commander

Результат работы программы (рис. [3](#fig:fig3)).

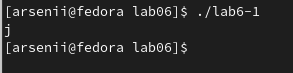


Figure 3: Окно создания папки

Заменяем символы на цифры в программе (рис. [4](#fig:fig4)).

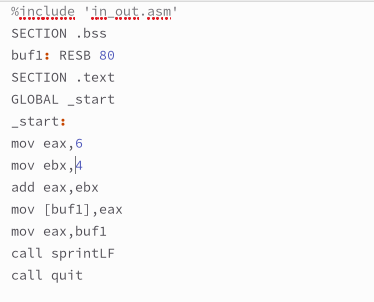


Figure 4: Окно создания файла

Программа с подключенными внешними функциями (рис. [5](#fig:fig5))

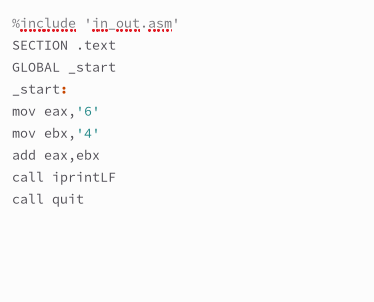


Figure 5: Файл в редакторе mcedit

Результат работы программы (рис. [6](#fig:fig6))

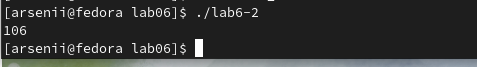


Figure 6: Файл в режиме просмотра

Заменяем символы на цифры в программе (рис. [7](#fig:fig7))

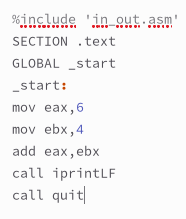


Figure 7: Линковка и компоновка

Результат работы программы(рис. [8](#fig:fig8))

Figure 8: Работа первой программы

Figure 8: Работа первой программы

Код программы, считающей значение выражения (5\*2 + 3)/3(рис. [9](#fig:fig9))

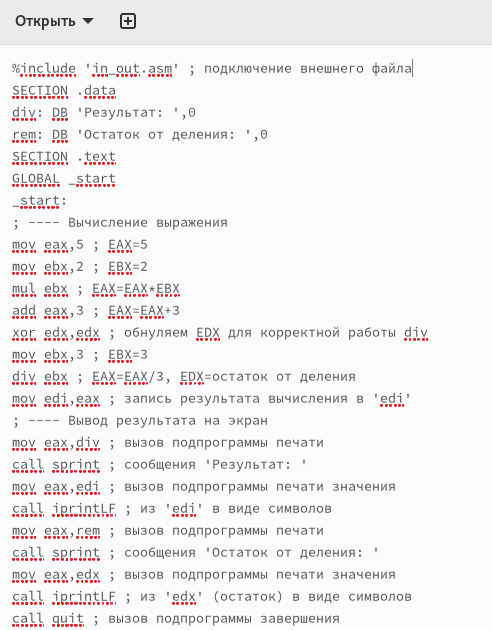


Figure 9: Окно замены файла

Результат работы программы(рис. [10](#fig:fig10))

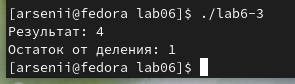


Figure 10: Код второй программы

Код программы, считающей значение выражения (4\*6 + 2)/5(рис. [11](#fig:fig11))

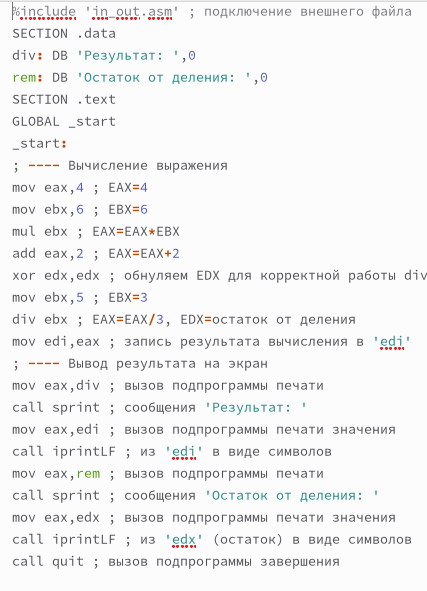


Figure 11: Вторая программа

Результат работы программы (рис. [12](#fig:fig12))

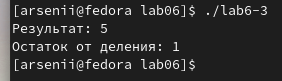


Figure 12: Измененная вторая программа

Результат работы программы, выводящей номер варианта по номеру студенческого билета (рис. [13](#fig:fig13))

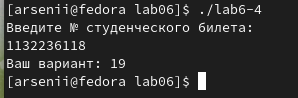


Figure 13: Измененная вторая программа

# 5 Самостоятельная работа

Написать программу вычисления выражения y=f(x) Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x, вычислять заданное выражение в зависимости от введенного x, выводить результат вычислений. Вид функции f(x)=(x/3+5)\*7 Создайте исполняемый файл и проверьте его работу для значений 3 и 9

Код программы(рис. [14](#fig:fig14))

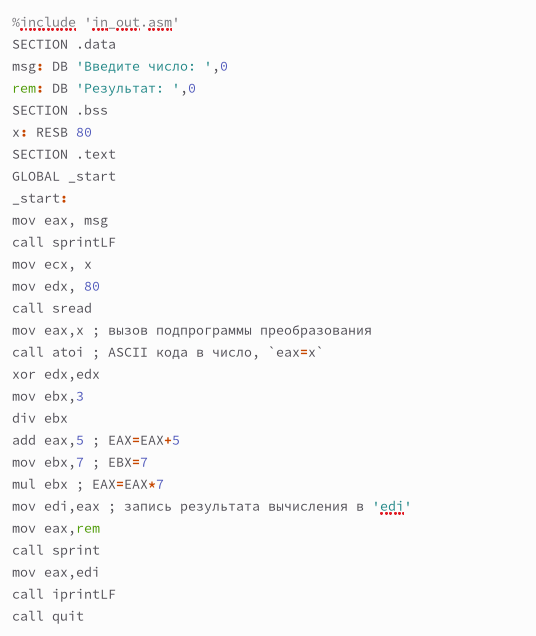


Figure 14: Код программы

Результат работы программы (рис. [15](#fig:fig15))

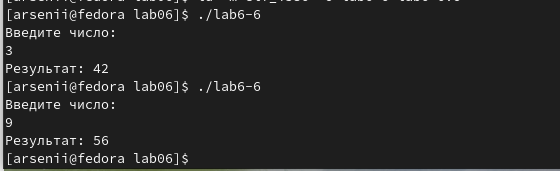


Figure 15: Работа программы

# 6 Выводы

Освоили арифметические инструкции языка ассемблера NASM

# Список литературы