Лабораторная работа №6

Архитектура вычислительных систем

Дмитрий Владимирович Орлюк

Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

1. 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, выво- дить результат вычислений. Вид функции 𝑓(𝑥) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполне- нии лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений 𝑥1 и 𝑥2 из 6.3

# 3 Выполнение лабораторной работы

1. Создадим каталог для программ л.р №6, перейдем в него и создадим файл



Рис. 1: Создание каталога для программ

1. Введем в файл текст программы из листинга, с помощью MC



Рис. 2: Ввод текст программы

1. Создаем исполняемый файл и запускаем

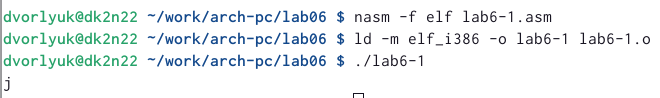


Рис. 3: Создание и запуск исполняемого файла

1. Изменим текст программы (уберем кавычки)

Рис. 4: Изменения в тексте программы

Рис. 4: Изменения в тексте программы

1. Создаем исполняемый файл и запускаем его.

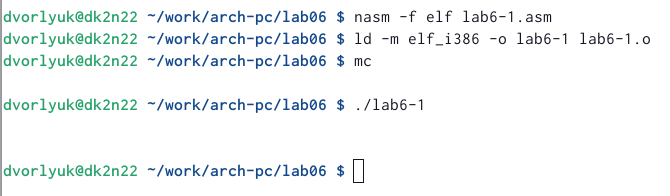


Рис. 5: Создание и запуск исполняемого файла

1. Просмотрим по таблице ASCII определим какому символу соответствует код 10. Символ не отображается.

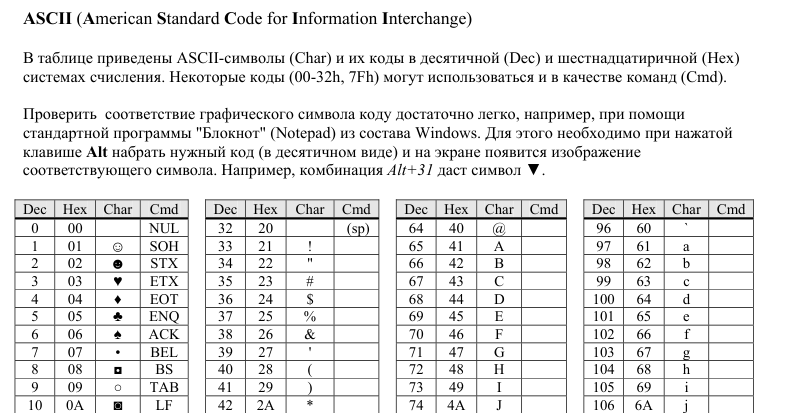


Рис. 6: Таблица ASCII

1. Создадим файл ассемблера с помощью команды touch

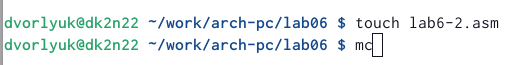


Рис. 7: Создание файла lab6-2

1. Введем команду из листинга 6.2

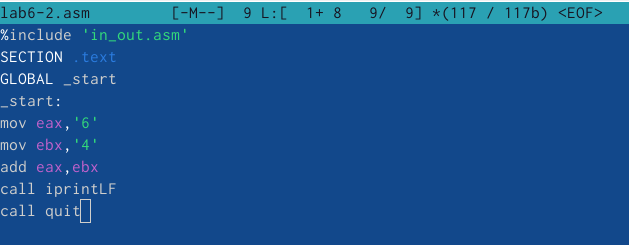


Рис. 8: Ввод текста программы из листинга

1. Создаем исполняемый файл и запускаем его.

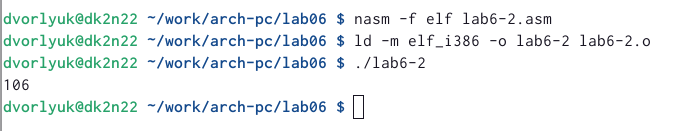


Рис. 9: Создание и запуск исполняемого файла

1. Вводим нужные изменения в текст программы

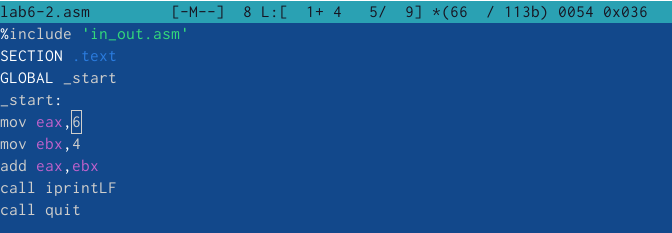


Рис. 10: Изменение программы

1. Создадим исполняемый файл и проверим его работоспособность (Команда выводит просто число 10. Команды iprint и iprintLF отличаются тем, что LF - это перенос на новую строку)

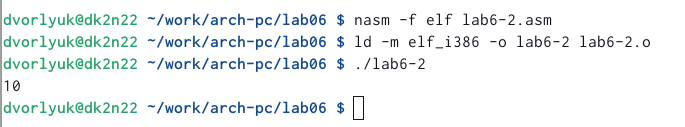


Рис. 11: Проверка работоспособности

1. Создадим файл 6-3 с помощью команды touch

Рис. 12: Создание файла 6-3

Рис. 12: Создание файла 6-3

1. Вставим текст из листинга.

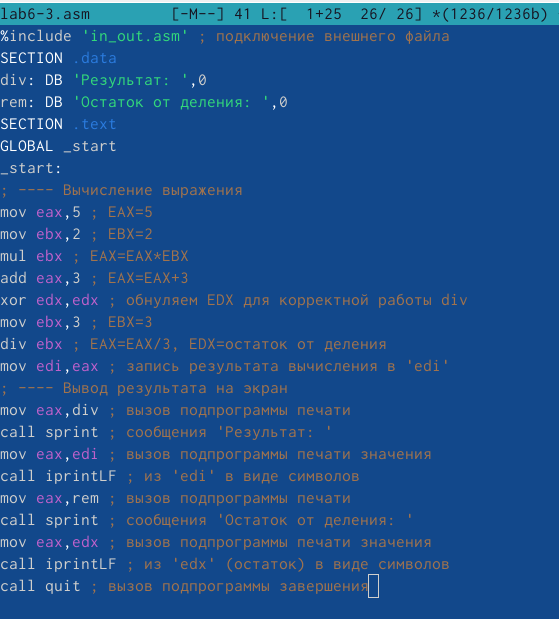


Рис. 13: Программа для вычисления выражения

1. Создадим исполняемый файл и запустим.

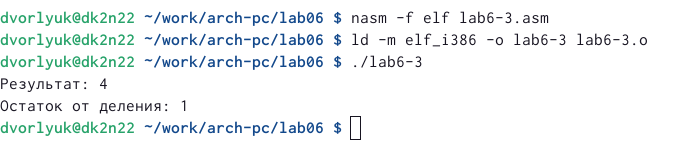


Рис. 14: Проверка работоспособности

1. Меняем текст программы для вычисления другого выражения.

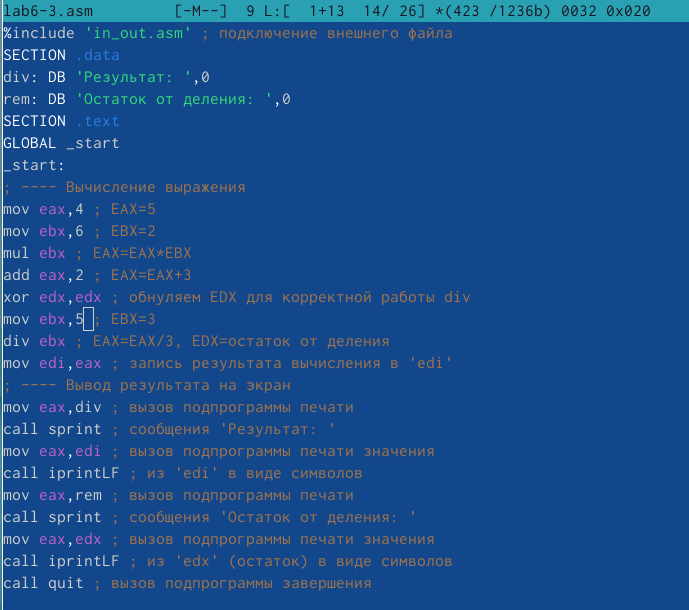


Рис. 15: Программа для вычисления измененного выражения

1. Проверяем работоспособность

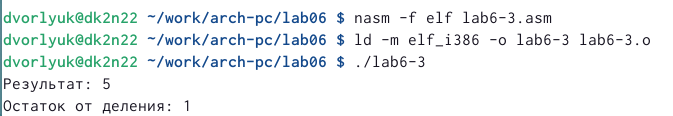


Рис. 16: Проверка работоспособности

1. Создадим файл под названием vаriant

Рис. 17: Создание файла variant

Рис. 17: Создание файла variant

1. Вставка нужной программы

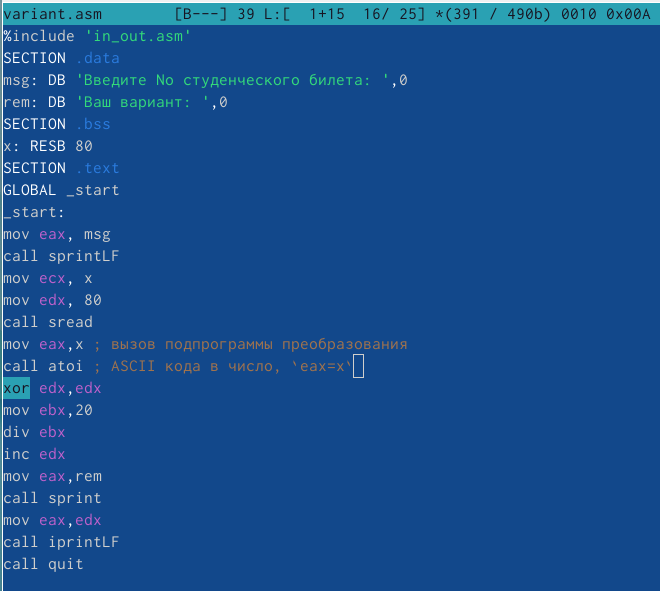


Рис. 18: Программа для вычисления варианта по номеру студ. билета

1. Проверка работоспособности

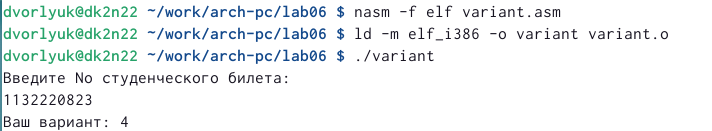


Рис. 19: Проверка работоспособности

#Ответы на вопросы

1. Какие строки листинга 7.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’ Строчка rem: DB “Ваш вариант: ’,0 отвечает за вывод сообщения.
2. Для чего используется следующие инструкции? nasm mov ecx, x mov edx, 80 call sread
3. Для чего используется инструкция “call atoi”? Функция call atoi преобразует ascii-код символа в целое число и записывает результат в регистр еах.
4. Какие строки листинга 7.4 отвечают за вычисления варианта? mov eax, msg call sprintLF
5. В какой регистр записывается остаток от деления при выполнении инструкции “div ebx”? Остаток записывается в регистр ah.
6. Для чего используется инструкция “inc edx”? Команда inc edx увеличивает значение регистра edx на
7. Какие строки листинга 7.4 отвечают за вывод на экран результата вычислений? mov eax,rem call sprint mov eax,edx call iprintLF call quit

# 4 Выводы

Я освоил арифметические инструкции языка ассемблера NASM, а также узнал как записывать в МС все действия с числами