Отчёт по лабораторной работе №6

Арифметические операции в NASM.

Попутников Егор Сергеевич

Содержание

# 1 Цель работы

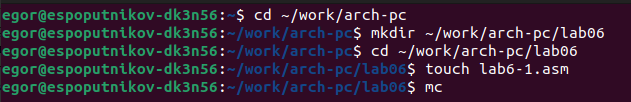
Освоение арифметических инструкций языка ассемблера NASM.

# 2 Задание

Написать программу вычисления выражения y=f(x). Программа должна выводить выражение для вычисления, выводить запрос на ввод значения x, вычислять задан- ное выражение в зависимости от введенного x, выводить результат вычислений. Вид функции f(x) выбрать из таблицы 6.3 вариантов заданий в соответствии с номером полученным при выполнении лабораторной работы. Создайте исполняемый файл и проверьте его работу для значений x1 и x2 из 6.3.

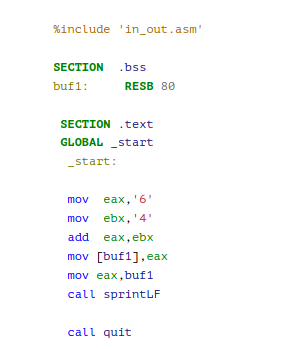
# 3 Выполнение лабораторной работы

1. Создадим каталог для программ лабораторной работы № 6, перейдем в него и создадим файл lab6-1.asm:(??)



Создание файла

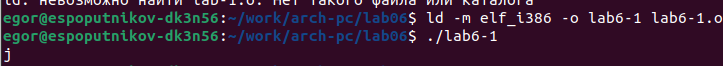
1. Введём в файл lab6-1.asm текст программы из листинга 6.1.(??).



Текст из листинга

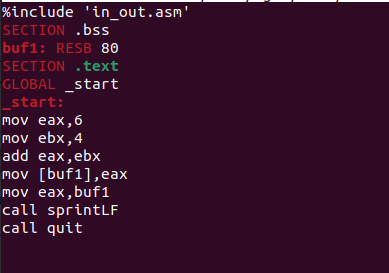
1. Создадим исполняемый файл и запустим его.(??).

Figure 1: 



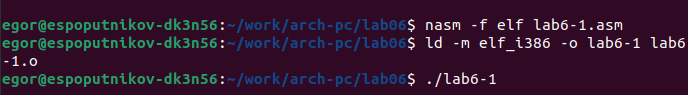
Создание исполняемого файла

1. Далее изменим текст программы и вместо символов, запишем в регистры числа. Ис- правим текст программы (Листинг 6.1) следующим образом: заменим строки: mov eax,‘6’ mov ebx,‘4’ на строки mov eax,6 mov ebx,4 (??).



Изменение текста листинга

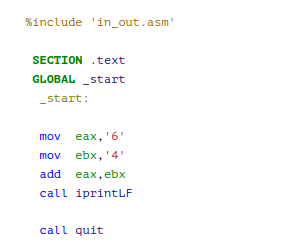
1. Создадим исполняемый файл и запустим его.(??).

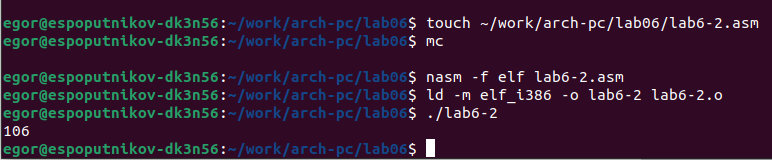


Создание исполняемого файла

Как и в предыдущем случае при исполнении программы мы не получим число 10. В данном случае выводится символ с кодом 10.Число 10 в таблице ASCII соответствует символу переноса строки,следовательно, на экран ничего не выводится.

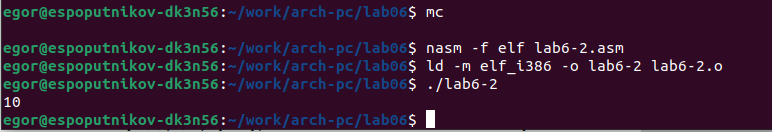
1. Преобразуем текст программы из Листинга 6.1 с использованием функций из файла in\_out.asm.Создадим файл lab6-2.asm исправим в нём текст из листинга и создадим исполняемый файл.(??)





Преобразование текста из листинга

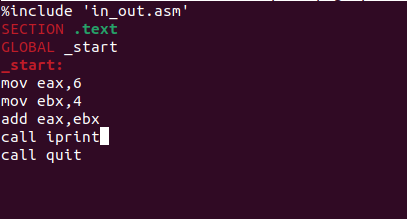
1. Аналогично предыдущему примеру изменим символы на числа. Заменим строки: mov eax,‘6’ mov ebx,‘4’ на строки mov eax,6 mov ebx,4 Создадим исполняемый файл и запустим его. (??)

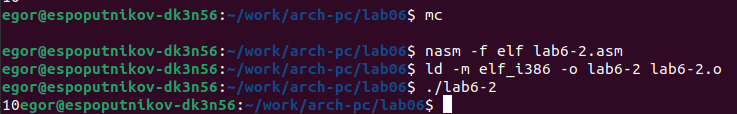


Создание исполняемого файла

В результате выполнения программы был получен результат 10.

1. Заменим функцию iprintLF на iprint. Создадим исполняемый файл и запустим его.(??)





Преобразование текста из листинга

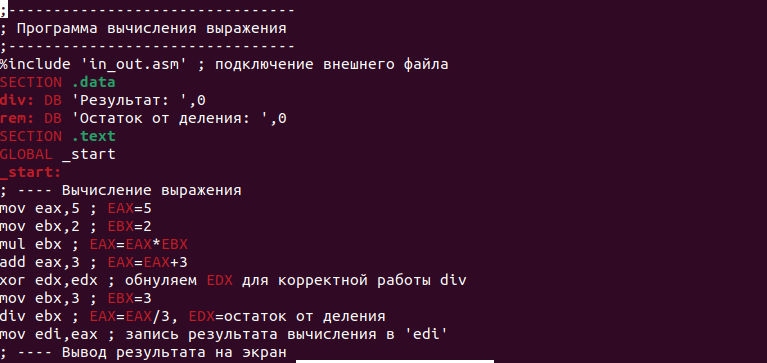
Функции отличаются тем, что iprintLF переводит написанный текст на следующую строку, а iprint нет.

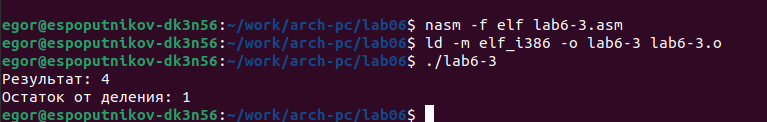
1. Создадим файл lab6-3.asm в каталоге ~/work/arch-pc/lab06(??)

Создание файла

Создание файла

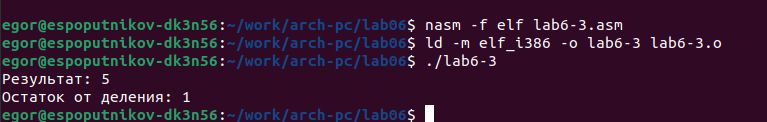
1. Запишем текст из листинга 6.3, создадим исполняемый файл и запустим его.(??)





Создание исполняемого файла

1. Изменим текст программы для вычисления выражения f(x) = (4 \* 6 + 2)/5. Создадим исполняемый файл и проверим его работу.(??)

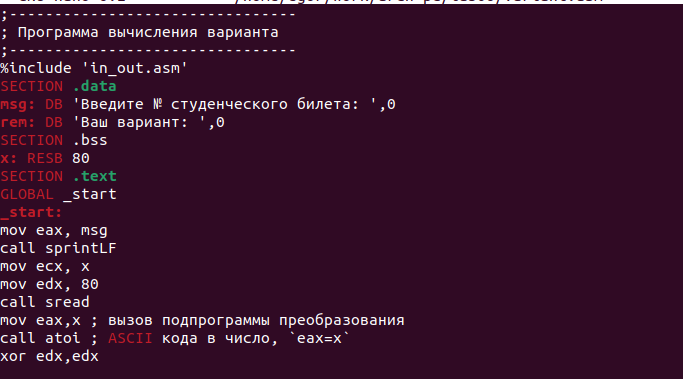


Изменение текста из листинга 6.3

1. В качестве другого примера рассмотрим программу вычисления варианта задания по номеру студенческого билета, работающую по следующему алгоритму: • вывести запрос на введение № студенческого билета • вычислить номер варианта по формуле: (Sn mod 20) + 1, где Sn – номер студен- ческого билета (В данном случае a mod b – это остаток от деления a на b). • вывести на экран номер варианта. В данном случае число, над которым необходимо проводить арифметические операции, вводится с клавиатуры. Как отмечалось выше ввод с клавиатуры осуществляется в символь- ном виде и для корректной работы арифметических операций в NASM символы необхо- димо преобразовать в числа. Для этого может быть использована функция atoi из файла in\_out.asm. Создадим файл variant.asm в каталоге ~/work/arch-pc/lab06:

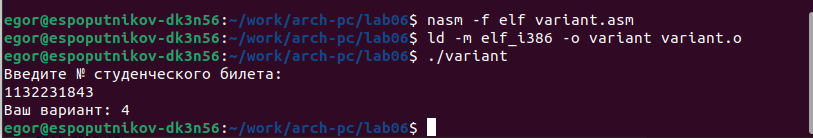
touch ~/work/arch-pc/lab06/variant.asm

Запишем в него текст из листинга 6.4: (??)



Запись текста из листинга 6.4

Создадим исполняемый файл и проверим работу программы.(??)

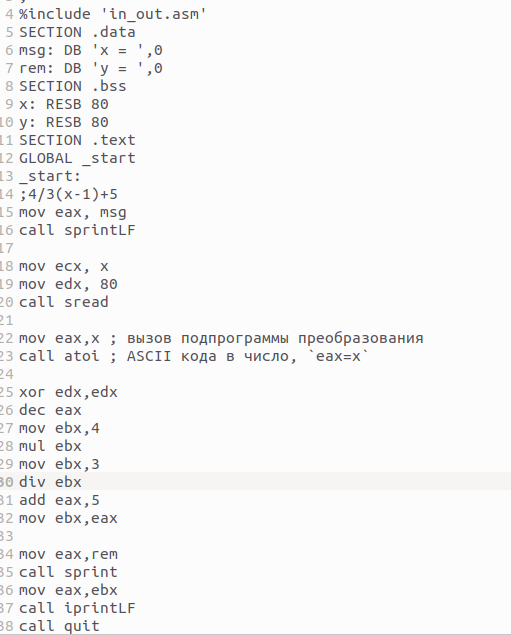


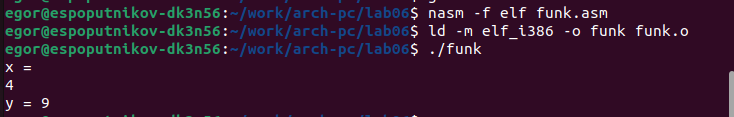
Создание исполняемого файла

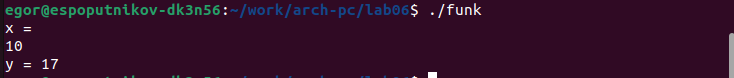
1. Какие строки листинга 6.4 отвечают за вывод на экран сообщения ‘Ваш вариант:’? За вывод сообщения на экран отвечают строки: rem: DB ‘Ваш вариант:’,0 mov eax,rem call sprint
2. Для чего используется следующие инструкции? mov ecx, x mov edx, 80 call sread Добавление в регистр ecx x, добавление в регистр edx 80, чтение с консоли
3. Для чего используется инструкция “call atoi”? Преобразование 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 edx.
6. Для чего используется инструкция “inc edx”? Прибавление 1 к регистру edx.
7. Какие строки листинга 6.4 отвечают за вывод на экран результата вычислений? call iprintLF

# 4 Выполнение самостоятельной работы

Напишем программу вычисления выражения y = 4/3\*(x − 1)+5.Создадим исполняемый файл и проверим его работу для значений x1 и x2(4,10)(??)







Выполнение самостоятельной работы

# 5 Выводы

Я освоил арифметические инструкции языка ассемблера NASM.