ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

дисциплина: Архитектура компьютера

Холопов Илья Алексеевич

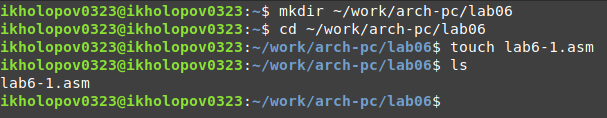
Содержание

# 1 Цель работы

Освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

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



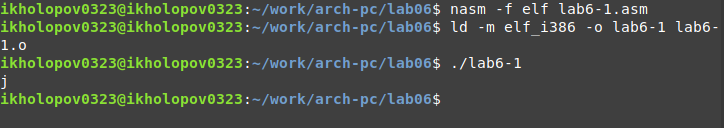
Создание каталога и файла в нем lab6-1.asm

Введем в файл lab6-1.asm текст программы. В данной программе в регистр eax записывается символ 6 (mov eax,‘6’), в регистр ebx символ 4 (mov ebx,‘4’). Далее к значению в регистре eax прибавляем значение регистра ebx (add eax,ebx, результат сложения запишется в регистр eax). Далее выводим результат. Так как для работы функции sprintLF в регистр eax должен быть записан адрес, необходимо использовать дополнительную переменную. Для этого запишем значение регистра eax в переменную buf1 (mov [buf1],eax), а затем запишем адрес переменной buf1 в регистр eax (mov eax,buf1) и вызовем функцию sprintLF (рис. ??).



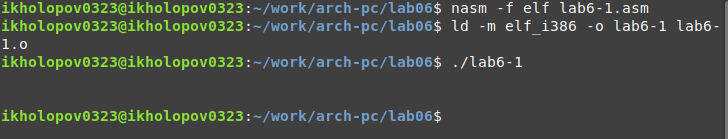
Содержимое файла lab6-1.asm

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



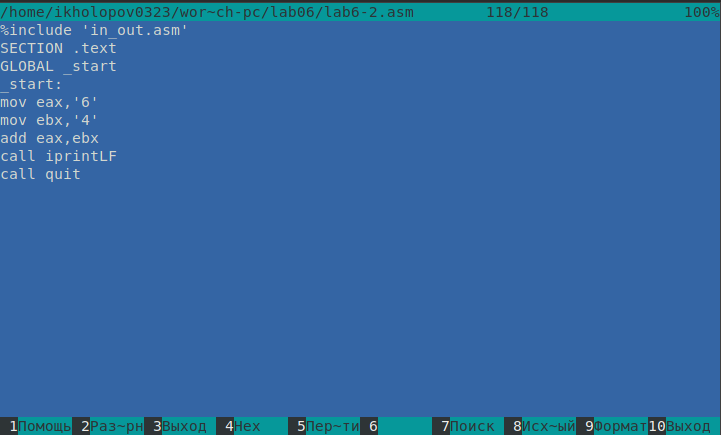
Создание и запуск исполняемого файла

Далее изменим текст программы, вместо символов запишем в регистры числа. Создадим и запустим исполняемый файл. В результате программа выводит символ с кодом 10, т.е символ перевода строки (рис. ??).



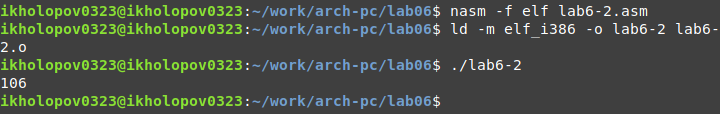
Результат работы измененной программы

Для работы с числами в файле in\_out.asm реализованы подпрограммы для преобразования ASCII символов в числа и обратно. Преобразуем текст программы из lab6-1.asm с использованием этих функций. (рис. ??).



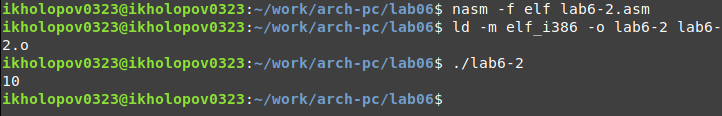
Использование подпрограмм из in\_out.asm

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



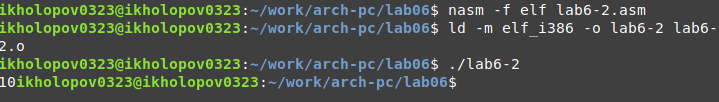
Создание и запуск исполняемого файла

Теперь изменим символы на числа. Создадим и запустим исполняемый файл. На рис. ?? видно, что в результате выводится на экран число 10.



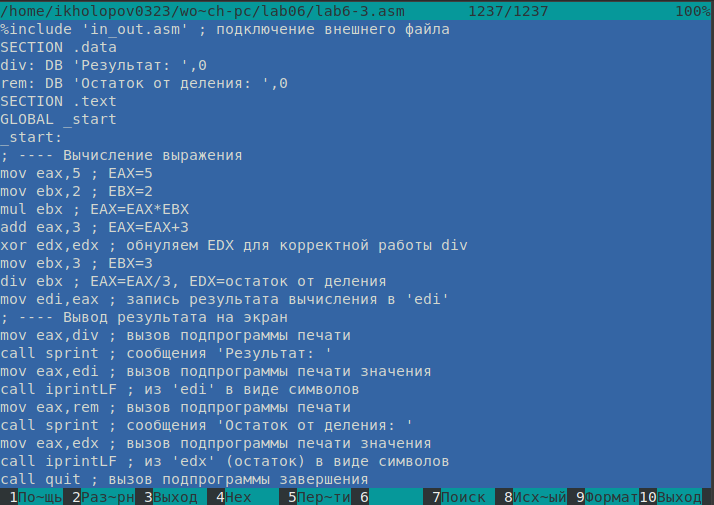
Результат работы программы с числами вместо символов

Заменим функцию iprintLF на iprint. Создадим исполняемый файл и запустим его. На рис. ?? видно, что iprint в отличии от iprintLF не выводит символ перевода строки.



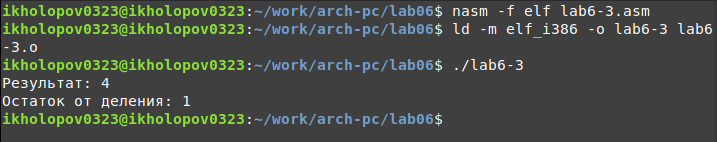
Работа программы с iprint вместо iprintLF

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



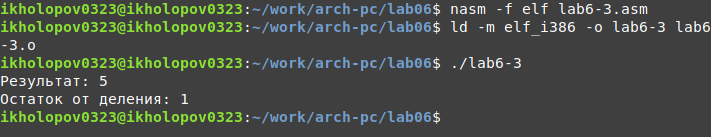
Содержимое файла lab6-3.asm

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



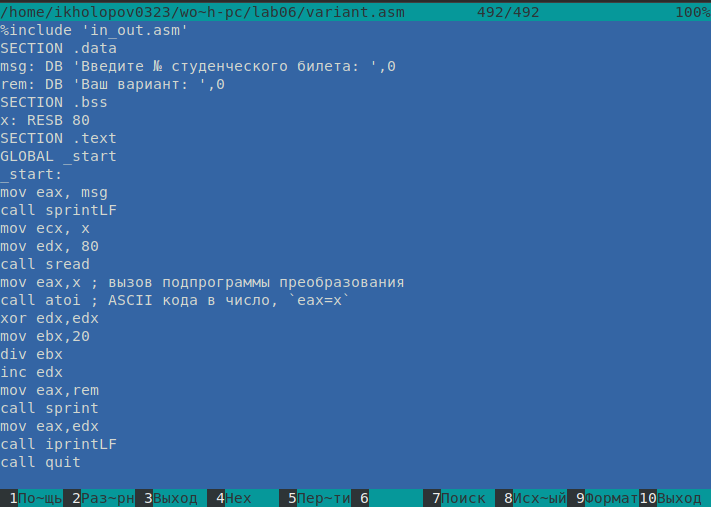
Создание и запуск исполняемого файла

Изменим текст программы для вычисления выражения . Создадим исполняемый файл и проверим его работу (рис. ??).



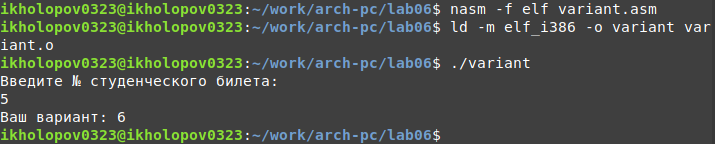
Создание и запуск исполняемого файла для вычисления выражения

Создадим файл variant.asm в каталоге ~/work/arch-pc/lab06. Напишем в файле variant.asm код программы, вычисляющей вариант по номеру введенного с клавиатуры (рис. ??).



Содержимое файла variant.asm

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



Создание и запуск исполняемого файла, вычисляющего номер варианта

Следующие строки отвечают за вывод на экран сообщения ‘Ваш вариант:’:

mov eax,rem  
call sprint

Данные введенные с клавиатуры записываются в x:

mov ecx, x  
mov edx, 80  
call sread

Инструкция “call atoi” используется для перевода ASCII кода в число.

Следующие строки отвечают за вычисление варианта:

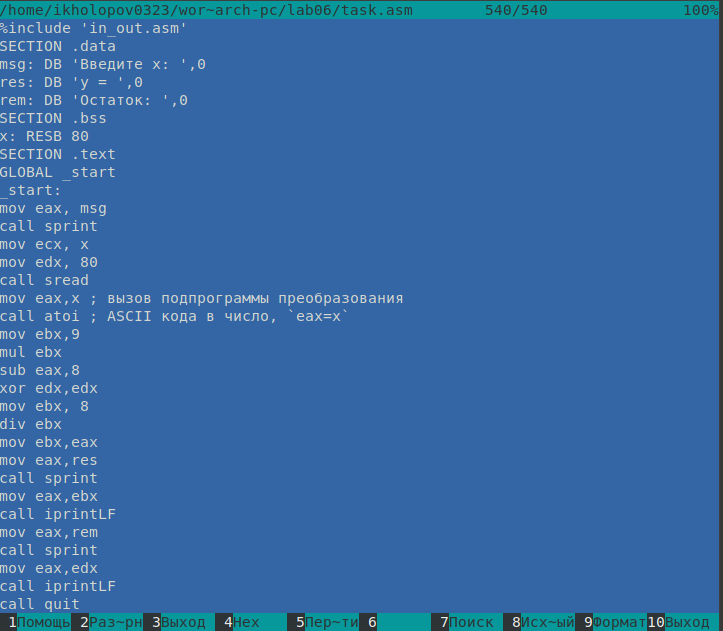
xor edx,edx  
mov ebx,20  
div ebx  
inc edx

В регистр edx записывается остаток от деления при выполнении инструкции “div ebx”. Инструкция “inc edx” используется, так как необходимо вычислить остаток от от деления на 20 и прибавить 1.

Следующие строки отвечают за вывод на экран результата вычислений:

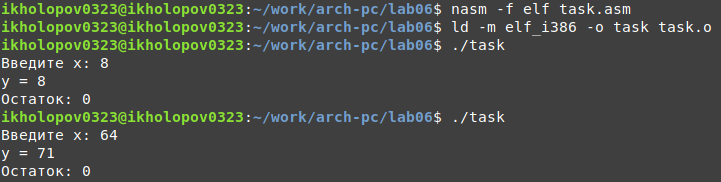
mov eax,edx  
call iprintLF

Напишем программу для вычисления выражения (рис. ??).



Содержимое файла программы для вычисления значения y

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



Создание и запуск исполняемого файла, вычисляющего значение y

# 3 Выводы

В результате выполнения лабораторной работы были освоены арифметические инструкции языка ассемблера NASM, созданы программы для вычисления выражений.