Лабораторная работа №7

Дисциплина: Архитектура компьютера

Александрова Ульяна Вадимовна

Содержание

# 1 Цель работы

Целью лабораторной работы является освоение арифметических инструкций языка ассемблера NASM.

# 2 Выполнение лабораторной работы

Создаю каталог lab07 и перехожу туда. В каталоге создаю файл *lab7-1.asm* и редактирую в текстовом редакторе так, чтобы текст соответствовал предложенному листингу. Создаю исполняемый файл и запускаю программу (рис. 1).

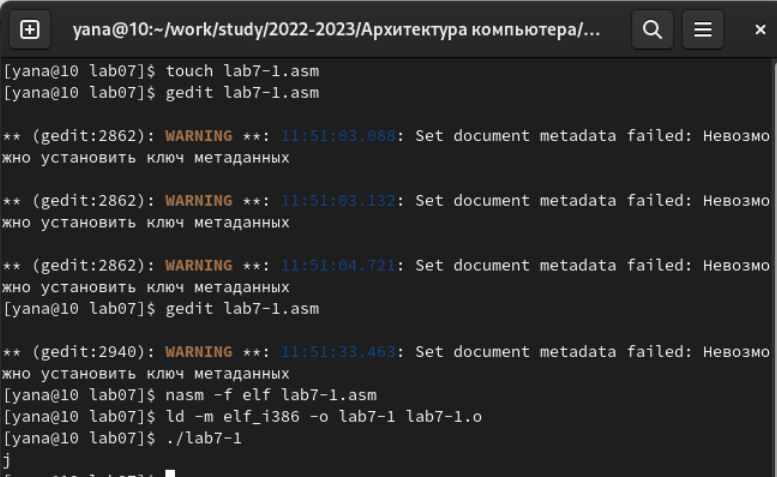


Рис. 1: Запуск программы lab7-1

Далее заменяю в программе символы ‘6’ и ‘4’ на 6 и 4, а затем запускаю программу. На экран был выведен символ перевода строка с кодом 10 (рис. 2).

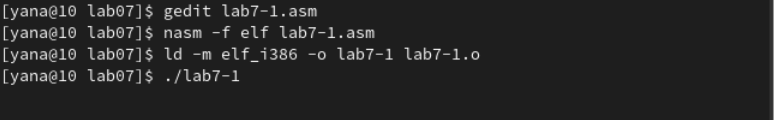


Рис. 2: Запуск новой программы

Создаю новый файл *lab7-2.asm* и ввожу в него текст программы из листинга 7.2. В результате работы программы получаю число 106 (рис. 3).

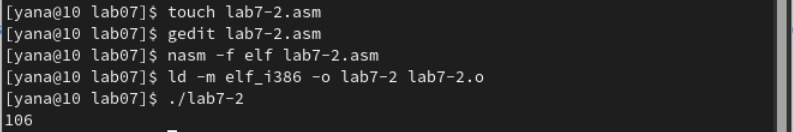


Рис. 3: Запуск программы lab7-2

Заменяю символы на числа и при запуске программы получаю число 10, тк система складывает числа, а не коды, соответствующие данным символам (рис. 4).

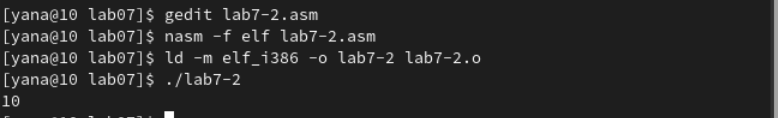


Рис. 4: Запуск измененной программы

Меняю функцию iprintLF на iprint. В результате запуска программы видно, что система не перешла на новую строку (рис. 5).

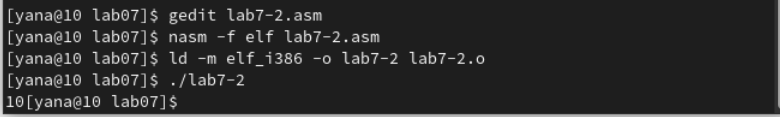


Рис. 5: Запуск измененной программы

Создаю файл *lab7-3.asm* и ввожу в него текст из листинга 7.3 (рис. 6).

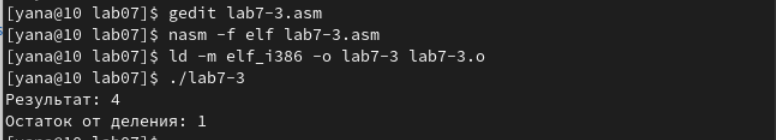


Рис. 6: Запуск программы lab7-3

Изменяю текст файла для соответствия заданию (рис. 7).

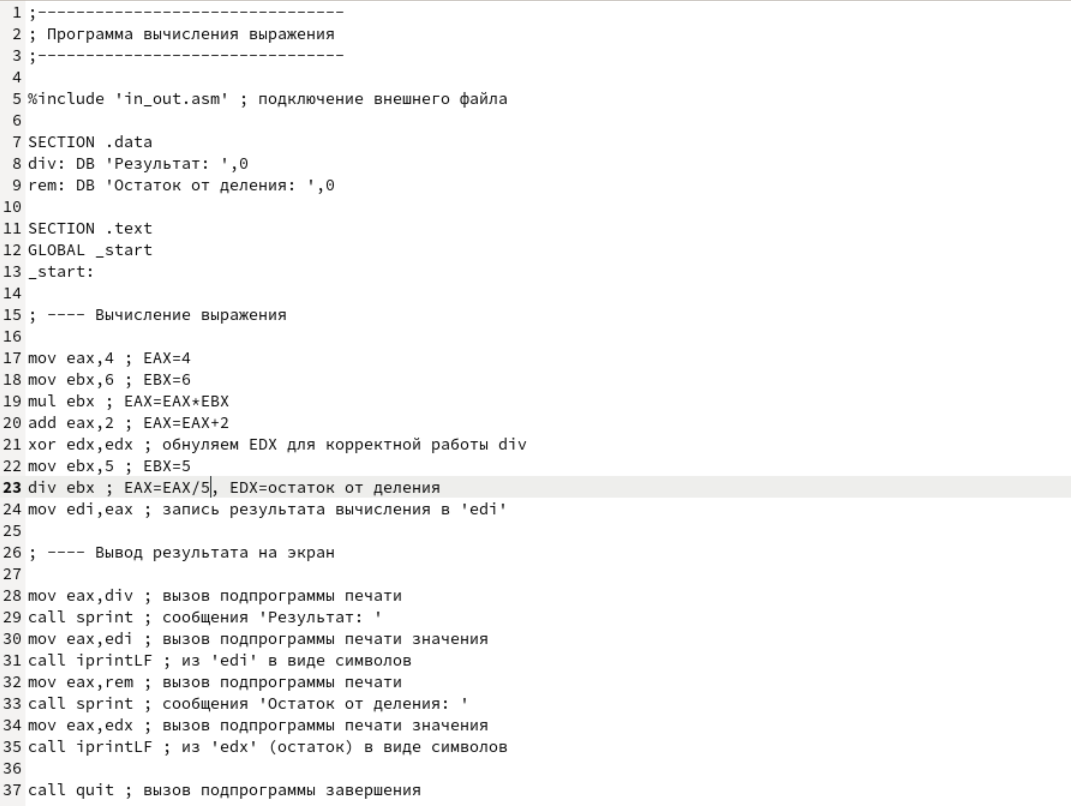


Рис. 7: Редактирование текста файла

Запускаю файл и получаю результат (рис. 8).

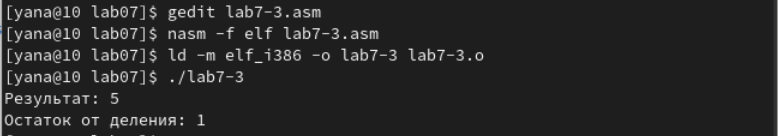


Рис. 8: Запуск измененной программы

Создаю файл *variant.asm* и вношу текст из листинга 7.4 в него. В результате запуска программы получаю свой вариант (рис. 9).

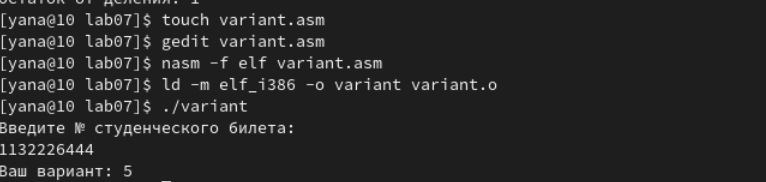


Рис. 9: Запуск программы variant

1. За вывод сообщения “Ваш вариант” отвечают данные строки кода

mov eax,rem  
call sprint

1. *mov ecx, x* используется для того, чтобы положить адрес вводимой пользователем строки в регистр ecx. *mov edx,80* – запись длины видимой строки. *call sread* – вызов подпрограммы из файла.
2. Инструкция используется для вызова подпрограммы, преобразующей код символа (ASCII) в целое число, а также записывает результат в регистр eax.
3. За вариант отвечают строки

xor edx,edx ; обнуление edx для корректной работы div  
mov ebx,20 ; ebx = 20  
div ebx ; eax = eax/20, edx - остаток от деления  
inc edx ; edx = edx + 1

1. Остаток от деления записывается в регистр edx.
2. инструкция *inc edx* используется для увеличения значения в регистре edx на один.
3. За вывод на экран результатов отвечают строки

mov eax,edx  
call iprintLF

# 3 Задание для самостоятельной работы

Я создала файл *zadanie.asm* и заполнила программу в соответствии с моим вариантом (рис. 10).

%include 'in\_out.asm'  
  
SECTION .data  
msg: DB 'Введите x: ',0  
rem: DB 'Ответ: ',0  
  
SECTION .bss  
x: RESB 80  
SECTION .text  
GLOBAL \_start  
  
\_start:  
mov eax, msg  
call sprintLF  
mov ecx, x  
mov edx, 80  
call sread  
mov eax,x ; вызов подпрограммы преобразования  
call atoi ; ASCII кода в число, `eax=x`  
mov ebx,9  
mul ebx  
sub eax,8  
mov ebx,8  
div ebx  
xor ebx, ebx  
mov edi, eax  
  
mov eax, rem  
call sprint ; сообщения 'Результат: '  
mov eax,edi ; вызов подпрограммы печати значения  
call iprintLF ; из 'edi' в виде символов  
  
call quit ; вызов подпрограммы завершения

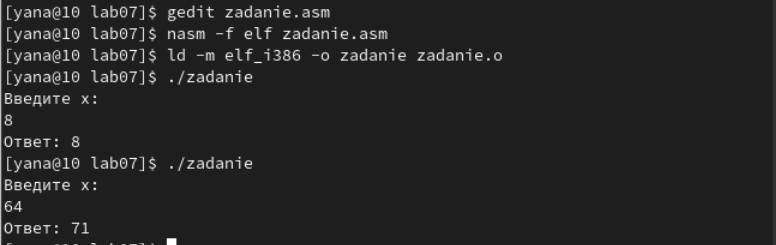


Рис. 10: Запуск программы

# 4 Выводы

Я освоила арифметические инструкции языка ассемблер NASM.