Государственный Университет Молдовы  
Факультет Математики и Информатики  
Департамент Информатики

**Лабораторная работа №7**

По курсу “Архитектура компьютера и язык ассемблера”

Тема: Команды целочисленой арифметики

Выполнил студент группы I1902:  
Чобану Артём

Кишинэу, 2019

**Лабораторная работа №7**

**Цель работы:** изучение арифметических команд Ассемблера, принципов их действия и применения в программе.

**Задание**

**Вариант 3**: Вычислить значение выражения (значения переменных вводятся с клавиатуры): (10\*a – 2\*b)\*c – d

a, b – define byte,c, d – define word

При этом, a,b,c,d – **полноценные числа, а не цифры.**

**Алгоритм:**

1. Перевод из строки в число:

1) Цифры выбираются, начиная со старшего разряда и прибавляются к числу(Предварительно вычтев 30h для преобразования в число).

2) Если следующий разряд существует, число умножается на 10.

3) Повторяется до того, как закончатся разряды

1. Перевод из числа в строку:

1) Число делится на 10.

2) Выбирается остаток и отправляется в стек.

3) Повторяется до того как целая часть будет равна нулю.

4) Цифры вытаскиваются из стека и выводятся на экран как символы путём прибавления к ним 30h

1. Для сокращения кода я использовал макросы, описанные в лабораторной работе No.10
2. В после подсчёта, происходит проверка знака числа. Если число отрицательное – выводится знак “-

**Код программы:**

|  |  |
| --- | --- |
| ;(10\*a-2\*b)\*c-d; a, b – define byte  ;c, d – define word  .model small  .stack  .data      a db 0      b db 0      c dw 0      d dw 0        n dw ? ;Результат будет  ;записываться в эту переменную        mlen db 4 ;Максимальная длина      len db ? ;Текущая длина      s db 4 dup (?) ;Буфер для строки        dn dw 10      newLine db 10, 13, '$' ;Сообщения  ;и переменная для перехода  ;на новую строку      m1 db "Enter A:", 10, 13, '$'      m2 db "Enter B:", 10, 13, '$'      m3 db "Enter C:", 10, 13, '$'      m4 db "Enter D:", 10, 13, '$'      m5 db "Result:", 10, 13, '$'  .code  ReadNumber macro msg, var      EnterNum msg ;Ввод      StringToNum var ;Преобразование в  число      ;Очистка буфера      mov s[0], ?      mov s[1], ?      mov s[2], ?      mov s[3], ?  endm  ReadByte macro msg, var ;Макрос ввода  ;для типа Byte      mov n, 0  mov ax, dx ;Берём результат      mul dn  ;Умножаем его на 10      mov dx, ax ;Возвращаем в dx      jmp strToNum    ex:      mov var, dx ;Возвращаем результат      mov ax, 0      mov dx, 0      mov bx, 0  endm  PrintByte macro var ;Вывод для типа  ;Byte      mov al, var      cbw      mov n, ax      PrintNum n      mov n, 0      mov ax, 0  endm  PrintNum macro var ;Макрос вывода    ;на экран      local toStack      local output      mov ax, 0 ;Очистка регистров      mov dx, 0      mov bx, 0      mov len, 0;Сброс длины. Длину    ;будем считать      mov ax, var      toStack:      inc len ;Увеличиваем длину  mov dx, 0 ;Делим ax на 10,    ;обнулив перед этим dx      div dn      add dx, 30h;Добавляем к остатку  ;30h чтобы получить символ цифры      cmp ax, 0 ;Сравниваем целую часть    ;с нулём      mov al, b ;То же самое с b      cbw      mov bx, ax      mov ax, 2 ;Умножаем на 2      mul bx      mov bx, ax      mov dx, 0      sub n, bx ;Вычитаем      mov ax, n      mul c   ;Умножаем на c      sub ax, d ;Вычитаем d      mov n, ax      mov ah, 9      lea dx, m5      int 21h | ReadNumber msg n      mov ax, n   cbw      mov var, al      mov ax, 0      mov n, 0  endm  EnterNum macro msg ;Макрос ввода      mov ah, 9      lea dx, msg      int 21h      mov ah, 0Ah      lea dx, mlen      int 21h        mov ah, 9      lea dx, newLine      int 21h  endm  StringToNum macro var ;Макрос перевода  ;в число      local strToNum ;Обьявление  ;локальных меток      local ex      mov ax, 0 ;Обнуляем регистры      mov dx, 0      mov bx, 0      strToNum:      mov al, s[bx] ;Перенос цифры в al      cbw ;Переводи в ax      sub ax, 30h ;Вычитаем 30h      add dx, ax ;Складываем с dx      inc bl ;Увеличиваем bl      cmp bl, len ;Пока не  ;дойдём до длины  je ex  mov ax, 0        push dx ;Отправляем остаток    ;в стек      jne toStack ;Повторяем если    ;целая часть > 0      output:      mov ah, 2 ;Вывод символа      pop dx  ;Сам символ берётся    ;из стека      int 21h      dec len ;Длина уменьшается      cmp len, 0 ;Проверка, закончились    ;ли разряды      jne output ;Повторяем пока    ;не кончатся цифры      NL  endm  NL macro ;Макрос для перехода   ;на новую строку      mov ah, 9      lea dx, newLine      int 21h  endm  mov ax, @data  mov ds, ax      ;Ввод всех 4 чисел:      ReadByte m1 a      ReadByte m2, b      ReadNumber m3, c      ReadNumber m4, d      mov al, a ;а - byte, поэтому      cbw       ;переводим его в ax      mov n, ax ;чтобы перевести в word      mov ax, 10      mul n ;Умножаем n на 10      mov n, ax      mov dx, 0      cmp n, 0      jg p      neg n      mov ah, 2      mov dl, "-"      int 21h        mov ax, 0      mov dx, 0  p:  PrintNum n  mov ah, 4Ch  int 21h  end |

Вывод программы:



Для отрицательного результата:

