МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №7
по дисциплине «Организация ЭВМ и систем»
Тема: Использование арифметических операций над целыми числами и процедур в Ассемблере.

Студент гр. 9383 ___ Поплавский И.

Преподаватель ___ Ефремов М.А.

Санкт-Петербург
2020

Цель работы.

Научиться выполнять арифметические операции над целыми числами в Ассемблере.

Задание.

Разработать на языке Ассемблер процессора IntelX86 две процедуры:

- одна – выполняет прямое преобразование целого числа, заданного в регистре AX (или в паре регистров DX:AX) в строку, представляющую его символьное изображение в заданной системе счисления (с учетом или без учета знака в зависимости от варианта задания);
- другая - обратное преобразование строки, представляющей символьное изображение числа в заданной системе счисления в целое число, помещаемое в регистр AX (или в пару регистров DX:AX)

Строка должна храниться в памяти, а также выводиться на экран для индикации.

Ход работы.

Для 10 варианта предлагается реализовать преобразования по шифру 2.2.3 и 1C2A, где

2 – 32 бита,

2 – без учета знака,

3 – система счисления десятичная.

1C – near, через кадр стека,

2A – far, только через РОНы (регистры общего назначения).

Программа выполняет преобразование числа в строку в функции NUMBERLINE и строки в число в функции LINENUMBER, далее выполняется сравнение двух чисел введенных пользователем. Результатом

программы служит, вывод преобразованного числа и ответ на вопрос, равны числа или нет.

Тестирование.

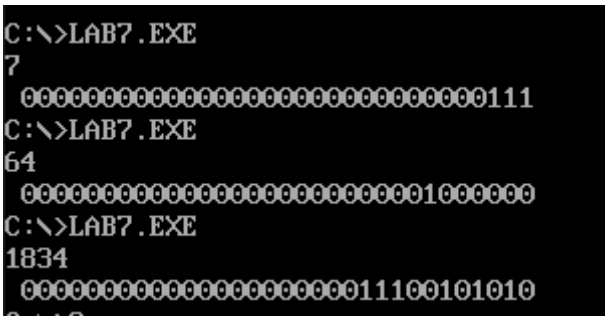

Рис 1 – Тестирование программы

Вывод.

В ходе выполнения программы была составлена программа выполняющая сравнение двух чисел, преобразования из строки в число и из числа в строку.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
stack segment stack
   dw 64 dup(0)
stack ends

data segment
   origin db 33, ?, 33
dup(0)
   result db 33 dup(0)
data ends
code segment
   assume ds:data,
cs:code, ss:stack

   strToInt proc far

            push ax
            mov ax, data
            mov ds, ax
            pop ax

            xor cx, cx
            mov ah,0ah
            mov dx,offset
origin              ;
Считывание строки и
запись её в буфер,
перевод на новую
строку
            int 21h
            mov dl,0ah
            mov ah,02
            int 21h
```

```
        mov si,offset
origin+2

        xor ax,ax
                ;
Готовим регистры для
записи: ax = 0, dx = 0,
bx = 2 - основание СС
        xor dx, dx
        mov bx,10

        mov cl,
origin[1]

        transformdx:
        cmp cx, 17
                ; Расчёт
двух старших байтов
        jl sec_word

        push cx

        mov cl, [si]
        cmp cl,'0'
                ;
Проверка на
соответствие цифре
        jb err
        cmp cl,'9'
        ja err

        sub cl,'0'
                ;
Перевод из кода
символа в цифру,
домножение на 10,
прибавление в конец
```

```
        mul bx
        add ax,cx
        inc si

        pop cx

        loop
transformdx

        sec_word:
                ; Расчёт
двух младших байтов
        push ax
        xor ax, ax

        transformax:
        mov cl,[si]
                ;
Проверка на
последний символ
        cmp cl,0dh
        jz fin

        cmp cl,'0'
                ;
Проверка на
соответствие цифре
        jb err
        cmp cl,'9'
        ja err

        sub cl,'0'
                ;
Перевод из кода
символа в цифру,
домножение на 2,
прибавление в конец
```

```
        mul bx
        add ax,cx
        inc si
        jmp
transformax

        err:
        mov dx, offset
error   ; Ошибка
(если не цифра), выход
        mov ah,09
        int 21h
        int 20h

        fin:
        pop dx
        pop cx
        pop bx

        push ax
                        ;
Помещение числа в
стек
        push dx

        push bx
        push cx
        ret

  error db "incorrect
number$"
  strToInt endp

  intToStr proc near
        push ax
        push bx
```

```
        push cx
        push dx
        push di

        lea di, result
        ; Переход в
конец строки, запись
символа конца строки
        add di, 33
        mov cl, '$'
        mov [di], cl
        dec di

        mov cx, 16

        shiftax:
                shr ax, 1
                jc setax
                mov ch,
48
                jmp
recax

                setax:

   mov ch, 49           ;
Сдвиг вправо,
заполнение первых 16
знаков

                recax:
                mov [di],
ch
                dec di
                and cx,
00FFh
```

```
                loop
shiftax

        mov cx, 16

        shiftdx:
                shr dx, 1
                jc setdx
                mov ch,
48
                jmp
recdx

                setdx:

   mov ch, 49           ;
Сдвиг вправо,
заполнение последних
16 знаков

                recdx:
                mov [di],
ch
                dec di
                and cx,
00FFh
                loop
shiftdx

        pop di
        pop dx
        pop cx
        pop bx
        pop ax
        ret
   intToStr endp
```

```
   main proc far
            xor ax, ax
            push ds
            push ax

            mov ax, offset
data
            mov ds, ax

            call es:strToInt
            ; Ввод числа

            pop dx
                        ;
Получение числа из
стека
            pop ax

            call intToStr
            ; Запись числа
в виде строки

            mov dx, offset
result
            mov ah, 9
                  ; Вывод
результата на экран
            int 21h

            ret
   main endp
code ends
end main
```