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

ОТЧЕТ

по лабораторной работе №7

по дисциплине «Организация ЭВМ и систем»

Тема: Преобразование целых чисел. Использование процедур в Ассемблере.

Студент гр. 1381 _____ Мелькумянц Д.А.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Составить программу для преобразования чисел их одной заданной системы счисления в другую.

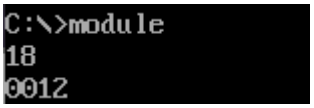
Задание.

Разработать на языке Ассемблер IntelX86 две процедуры: одна — прямого и другая — обратного преобразования целого числа, заданного в регистре ax. С учетом знака. Система счисления для изображения числа — десятичная. Связь данных между основной программой и подпрограммами осуществляется через сегмент стека.

Выполнение работы.

В процедуре Main в регистр ax вносится число, которое через стек передается в процедуру Ints. В процедуре Ints происходит проверка на отрицательное число, если отрицательное, то в строку DEC_STR добавляется «-», иначе происходит деление числа на 10 и сохранение остатков в стеке. После чего извлекаются остатки и записываются в строку DEC_STR, также добавляется символ конца строки и выход из процедуры. Далее строка DEC_STR печатается на экран. После чего выводится символ переноса строки и DEC_STR передаем в процедуру ToReg с помощью стека. В процедуре ToReg сперва идет проверка на знак минус, если он есть, то мы в di помещаем флаг 1. Дальше в bx передаем систему счисления, после чего идет проверка на конец строки. После чего символ преобразуем в 16-ю систему и записываем в ax. Если в di был флаг 1, то берется противоположное со знаком число в ax. После окончание процедуры переход в процедуру Hex. В процедуре Hex в cl заносится количество бит для считывания из строки, то есть вывод по 4 бита. Далее в al заносится цифра в соответствие с 16-ой системой, после чего в al получаем символ цифры и происходит проверка на цифру, если это цифра, то переходим к Digit, где записываем в строку и если cl еще больше или равно 0, то повторяем действия. После окончание процедуры выводим строку HEX_STR на экран.

Рис. 1 — Проверка работы программы.



Вывод.

Составлена программа преобразующая 16-ое число из регистра ах в 10-ое в строковом виде и из строкового 10-ого вида в строковое 16-ое.

Приложение А

Исходный код программы

Название файла: module.asm

```
AStack  SEGMENT STACK
   DB 1024 DUP(?)
AStack  ENDS

DATA    SEGMENT
   DEC_STR DB ' ','$'
   HEX_STR DB ' ', '$'
DATA    ENDS

CODE    SEGMENT
   ASSUME CS:CODE, DS:DATA, SS:AStack

WriteMsg  PROC  NEAR
        mov AH, 9
        int 21h
        ret
WriteMsg  ENDP

Ints   proc
       pop cx
       pop di
       pop ax

       push    bx
       push    cx
       push    dx
```

```
    mov     bx,     10
    xor     cx,     cx
    or      ax,     ax
    jns     div1
            neg     ax
            push    ax
            mov     dl,     '-'
                                    mov [di], dl
                                    inc di
            pop     ax
    div1:
            xor     dx,     dx
            div     bx
            push    dx
            inc     cx
            or      ax,     ax
    jnz     div1
    sto:
            pop     dx
            add     dl,     '0'
                                    mov [di], dl
                                    inc di
                loop    sto

mov dl, '$'
mov [di], dl
inc di

pop     dx
pop     cx
```

```
    pop     bx

    push cx
        ret
Ints        endp

ToReg proc
    pop cx
    pop dx

    push cx
    push si
    push di

    mov si, OFFSET DEC_STR
    cmp byte ptr [si], "-"
    jnz l1
    mov di, 1
    inc si
    l1:
        xor ax, ax
        mov bx, 10
    l2:
        mov cl, [si]
        cmp cl, '$'
        jz endin

        sub cl, '0'
        mul bx
        add ax, cx
```

```
        inc si
        jmp l2
    endin:
        cmp di, 1
        jnz l3
        neg ax
    l3:

        pop di
        pop si
        pop cx

        push ax
        push cx

        ret
ToReg endp

Hex proc
        pop cx
        pop di
        pop ax

        push    cx
        push    dx

        mov    cl,      ((16-1)/4)*4
        xchg   dx,      ax

Repeat:
```

```
        mov    ax,     dx
        shr    ax,     cl
        and    al,     0Fh
        add    al,     '0'
        cmp    al,     '9'
        jbe    Digit
        add    al,     'A'-('9'+1)

Digit:
                    push dx
                    mov dl , al
                    mov [di], dl
                    inc di
                    pop dx
        sub    cl,     4
        jnc    Repeat

        mov dl, '$'
        mov [di], dl
        inc di

        pop     dx

        ret
Hex endp

Main PROC FAR
      push ds
    sub ax,ax
```

```
push ax
mov ax, DATA
mov ds, ax

mov ax, 12h
push ax
mov di, OFFSET DEC_STR
push di
  call Ints

  mov dx, OFFSET DEC_STR
  call WriteMsg

  push dx
  push ax
  mov ah,2
mov dl, 10
int 21h
  pop ax
  pop dx

mov dx, OFFSET DEC_STR
push dx
  call ToReg

mov di, OFFSET HEX_STR
push di
  call Hex

  mov dx, OFFSET HEX_STR
```

```
    call WriteMsg

    ret
Main ENDP
CODE ENDS
END Main
```