

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

ОТЧЕТ
по лабораторной работе №7
по дисциплине «Организация ЭВМ и систем»
Тема: Преобразование целых чисел. Использование процедур в
Ассемблере.
Вариант 15

Студент гр. 1303

Кузнецов Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Составить программу для преобразования чисел из 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 символьного изображения числа - десятичная. Способ вызова процедуры - дальний. Связь данных между основной программой и подпрограммами осуществляется через сегмент стека.

Выполнение работы.

Main

В процедуре происходит запись в регистр ax и дальнейшие преобразования, требующиеся по заданию. Числа выводятся в строковом виде. Изначально в регистр ax вносится значение, которое проверяется на отрицательность, в результате этого на стек заносится знак плюса или минуса, он будет использоваться для печати. Так же, если число отрицательное, то к нему применяется neg. Далее это число приводится десятичный вид с помощью процедуры FROM_AX_TO_STR_10 и записывается в строку DEC_STR. Полученная строка «переворачивается» с помощью процедуры REVERSE и выводится. Далее строка преобразуется в 16-ый вид функцией FROM_STR_10_TO_REGISTER. Со стека снимается ранее полученный знак числа, и если это минус, то к числу применяется neg. Полученное число преобразуется в строку функцией FROM_AX_TO_STR_16. Полученная строка «отражается» и выводится.

FROM_AX_TO_STR_10

Число постоянно целочисленно делится на A, пока оно не станет нулем, а остаток от деления записывается в строку DEC

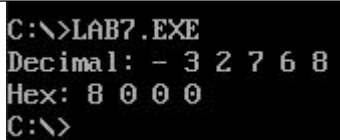
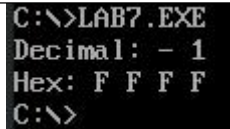
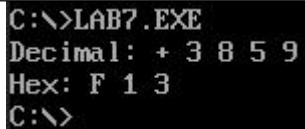
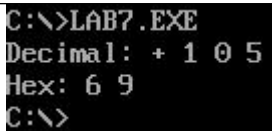
FROM_STR_10_TO_REGISTER

Каждая цифра числа умножается на основания 16-ой системы счисления, возведенной в степень позиции цифры. Для возведения используется функция POWER.

FROM_AX_TO_STR_16

Целочисленным делением, «отсекается» последняя цифра, которая приводится к символу и прибавляется к строке HEX.

Тестирование.

№ Теста	Ввод	Вывод	Результат
1	8000h	 <pre>C:\>LAB7.EXE Decimal: - 3 2 7 6 8 Hex: 8 0 0 0 C:\></pre>	Верно
2	0FFFFh	 <pre>C:\>LAB7.EXE Decimal: - 1 Hex: F F F F C:\></pre>	
3	0F13h	 <pre>C:\>LAB7.EXE Decimal: + 3 8 5 9 Hex: F 1 3 C:\></pre>	
4	69h	 <pre>C:\>LAB7.EXE Decimal: + 1 0 5 Hex: 6 9 C:\></pre>	

Вывод.

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

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab7.asm

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

DATA SEGMENT
    DECIM DB 'Decimal: ', '$'
    HEX DB 'Hex: ', '$'
    N DB ' ', 0ah, '$'
    DEC_STR DB ' ', '$'
    HEX_STR DB ' ', '$'
    SIGN DB ' ', '$'
DATA ENDS

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

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

POWER PROC NEAR
    push bx
    push dx
    push cx
    cmp cx, 0
    je power_end

    mov bx, 0Ah
power_begin:
    sub dx, dx
    mul bx
    sub cx, 1
    cmp cx, 0
    jne power_begin

power_end:
    pop cx
    pop dx
    pop bx
    ret
POWER ENDP

REVERSE PROC FAR
    pop ax
```

```

    pop cx

    pop di ;взятие строки переданной в функцию
    pop bx

    push cx
    push ax

    sub ax,ax
reverse_begin:
    push bx
    mov bx,ax
    mov cx,[di+bx]
    pop bx
    mov dx,[di+bx]

    push bx
    mov bx,ax
    mov [di+bx],dx
    pop bx
    mov [di+bx],cx

    add ax,2
    sub bx,2
    cmp ax,bx
    jnl reverse_begin
    ret
REVERSE ENDP

FROM_AX_TO_STR_10 PROC FAR
    pop bx
    pop cx

    pop di ;взятие значений переданных в функцию
    pop ax

    push cx
    push bx

    mov cx,0Ah
    mov bx,0
begin:
    sub dx,dx
    div cx
    add dx,'0'
    mov [di+bx],dx
    add bx,2
    cmp ax,0
    jne begin
    mov dx,'$'
    mov [di+bx],dx
    sub bx,2

```

```

    pop ax
    pop cx

    push bx ;возвращаю длину строки

    push cx
    push ax

    ret
FROM_AX_TO_STR_10 ENDP

FROM_STR_10_TO_REGISTER PROC FAR
    pop ax
    pop cx

    pop di ;взятие значений переданных в функцию
    pop bx

    push cx
    push ax

    sub dx,dx
    mov cx,0
    new_begin:
        mov ax,[di+bx]
        sub ax,'0'
        call power
        add dx,ax
        inc cx
        sub bx,2
        cmp bx,0
        jnl new_begin

    pop ax
    pop cx

    push dx ;возвращаю сумму

    push cx
    push ax

    ret
FROM_STR_10_TO_REGISTER ENDP

FROM_AX_TO_STR_16 PROC FAR
    pop ax
    pop cx

    pop di ;взятие значений переданных в функцию
    pop dx

    push cx
    push ax

```

```

    sub bx,bx
    mov ax,dx
    mov cx, 10h
    hex_begin:
        sub dx,dx
        div cx
        add dx,'0'
        cmp dx,'9'
        jle end_hex
        add dx,7

        end_hex:
        mov [di+bx],dx
        add bx,2
        cmp ax,0
        jne hex_begin
    mov cx,'$'
    mov [di+bx],cx
    sub bx,2

    pop ax
    pop cx

    push bx ;возвращаю длину строки

    push cx
    push ax

    ret
FROM_AX_TO_STR_16 ENDP

MAIN PROC FAR
    push ds
    sub ax,ax
    push ax
    mov ax, DATA
    mov ds, ax

    mov dx,offset decim
    call writemsg

    mov ax,0f13h ;!!! запись в AX
    mov di,offset sign
    mov bx,'+'
    cmp ax,0
    jnl skip
    mov bx,'-'
    neg ax

    skip:
    push bx
    mov [di],bx

```

```

push ax
mov dx,offset sign
call writemsg
pop ax

mov di,offset dec_str
push ax ;передача ax в функцию
push di
call from_ax_to_str_10
pop bx

push bx ;сохранил длину строки

mov di,offset dec_str
push bx ;передаю в функцию длинну строки
push di ;строку
call reverse

mov dx,offset dec_str
call writemsg
mov dx,offset n
call writemsg

pop bx;достал длину строки

mov di,offset dec_str
push bx
push di
call from_str_10_to_register
pop dx ;получил регистр

pop bx
cmp bx,'-'
jne second_skip
neg dx
second_skip:

mov di,offset hex_str
push dx
push di
call from_ax_to_str_16
pop bx

push bx ;передаю в функцию длинну строки
push di ;строку
call reverse

mov dx,offset hex
call writemsg
mov dx,offset hex_str
call writemsg

```



```
        ret
MAIN ENDP
CODE ENDS
      END MAIN
```