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

ОТЧЕТ

по лабораторной работе №6
по дисциплине «Организация ЭВМ и систем»
Тема: Использование арифметических операций над целыми числами и процедур в Ассемблере.

Студентка гр. 1303 _____ Королева П.А

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2022

Цель работы.

Разработать программу, переводящую число из одной системы счисления в другую.

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.

Отрицательные числа при представлении с учетом знака должны в памяти

храниться в дополнительном коде, а на экране изображаться в прямом коде с явным указанием знака или в символьном виде со знаком.

Вариант 13

16-битное число, с учетом знака, в восьмиричной системе счисления, способ вызова процедур – far, связь данных между основной процедурой и дополнительными через РОНы.

Выполнение работы.

Были реализованы четыре процедуры:

WriteMsg для вывода строки на экран.

Int_to_oct_str для преобразования числа в восьмиричную систему. Основная идея состоит в том, что в bx заносится делитель 8 и в цикле происходит деление числа на bx, остатки деления сохраняются в стеке. После цикла остатки вытаскиваются из стека, к ним прибавляется код ‘0’ для того чтобы получить код нужной цифры и заносятся в строку Oct_str. Благодаря этому остатки

записываются в обратном порядке. В конце добавляется символ окончания строки $.

Oct_str_to_int для преобразования числа из восьмиричной системы в десятичную. Подсчитывается длина строки Oct_str и вызывается цикл (количество итераций определяется длиной строки) где из каждой цифры строки вычитается код символа ‘0’, она умножается на 8 и добавляется к результату.

MAIN — основная процедура, где происходит вызов остальных процедур. Здесь проверяется знак числа, если оно положительно, в SIGN кладется символ плюса, иначе — минуса и число меняет знак с помощью команды neg.

Разработанный программный код см. в приложении А.

Вывод.

Разработать программа, переводящая число из одной системы счисления в другую.

ПРИЛОЖЕНИЕ А

Название файла: lr7.asm

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

DATA    SEGMENT
     N DW 0
     Oct_str DB 10, 13, ' ', '$'
     SIGN DB ' ', '$'
     NUMBER DW -36
DATA    ENDS

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


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



Int_to_oct_str proc FAR
    push ax
    push cx
    push dx
    push bx
    xor cx,cx              ;CX = 0
    mov bx,8               ;В BX делитель (8 для восьмиричной системы)
    mov di, offset Oct_str

lp1:                  ;Цикл получения остатков от деления
    xor dx,dx               ;Обнуление старшей части двойного слова
    div bx                  ;Деление AX=(DX:AX)/BX, остаток в DX
    add dl,'0'             ;Преобразование остатка в код символа
    push dx                 ;Сохранение в стеке
    inc cx                  ;Увеличение счетчика символов
    test ax,ax              ;Проверка AX
    jnz lp1           ;Переход к началу цикла, если частное не 0.

lp2:                  ;Цикл извлечения символов из стека
    pop dx                  ;Восстановление символа из стека
    mov [di],dl             ;Сохранение символа в Oct_str
    inc di                  ;Смещаем адрес Oct_str на единичку вправо
    loop lp2          ;Команда цикла

    mov bx, '$'
    mov [di], bx      ;положили символ конца строки

    pop bx
    pop dx
    pop cx
    pop ax
```

```
    ret
Int_to_oct_str ENDP


Oct_str_to_int proc FAR
    push di
    push cx
    push bx
    push dx

    mov di, offset Oct_str
    mov dx, '$'

    xor bx,bx
    len:
    cmp [di+bx], dx
    je en             ;Если встреченный символ - конец строки, выходим из
цикла
    inc bx
    jmp len
    en:
    mov cx, bx

    mov bx, 8
    mov dx, 0

    lp_1:
        mul bx
        mov dl, [di]
        sub dl, '0'
        add al, dl
        inc di
    loop lp_1

     mov di, offset N
    mov dx, [di]
    cmp dx, 0
    je pos_num

    neg ax

    pos_num:

    pop dx
    pop bx
    pop cx
    pop di
    ret
Oct_str_to_int endp


MAIN PROC FAR
    push DS
    xor ax,ax
    push ax
    mov ax, DATA
    mov ds, ax
```

```
    mov dx, offset Oct_str
    call WriteMsg

    mov ax, NUMBER
    mov di, offset SIGN
    mov bx, "+"
    cmp ax, 0
    jge set_sign
    mov bx, "-"
    neg ax
    push bx
    mov bx, 1
    mov N, bx
    pop bx

    set_sign:
        mov [di], bx
        inc di
        mov bx, '$'
        mov [di], bx

    push ax
    mov dx, offset SIGN
    call WriteMsg
    pop ax

    call Int_to_oct_str

    push ax
    mov dx, offset Oct_str
    call WriteMsg
    pop ax

    xor ax, ax
    call Oct_str_to_int
    ret
MAIN ENDP
CODE ENDS
    END MAIN
```