

```

DGROUP  group  _TEXT
_TEXT   segment use16 para public 'CODE'
        assume cs:DGROUP, ds:DGROUP, es:nothing, ss:nothing
        org     7C00h                ;offset used to find oemid

vbr     proc    far                    ;assume good unknown stack
        jmp     near ptr code         ;CAVEAT: fix jump for FAT1x
oemid    db      "REXXFAT "           ;"TA" > ".3" for "3.0" bug

        db      55 dup (?)           ;test worst case FAT32 BPB
        db      29h                  ;CAVEAT: code designed for
        db      12h,34h,56h,78h      ; offset -26 from code +3
        db      "VolumeLabel"       ; offset -22 from code +3
        db      "VolumeFS"          ; offset -11 from code +3

code:    call    pout                  ;"push IP", does not return
        db      "boot code placeholder for ",0
vbr     endp

show     proc    near                  ;clobbers AX and BX
        push    si

next:    lodsb                        ;DS:SI message ASCIIZ
        or      al,al                 ;AL = ASCII or NUL (Z)
        jz      done                 ;AL = 0 terminator (Z)
        mov     bx,7                  ;BL = 7 white on black
        mov     ah,0Eh               ;AH = 0Eh TTY output
        int     10h                  ;BH = 0 display page
        jmp     next

done:    pop     si
        retn                          ;AL = 0
show     endp

x2c      proc    near                  ;CX to ASCII (result buffer DS:DX) +++++
        push    es
        push    di                   ;keeps ES:DI and CX
        push    ax                   ;keeps DS:DX and AX

        push    ds
        pop     es
        mov     di,dx                ;ES:DI result DS:DX
        mov     ax,4                 ;AX = 4 hex. digits
        xchg    cx,ax                ;AX = CX, CX = 4

nibble:  rol     ax,4
        push    ax
        and     al,0Fh               ;+++++++ X2C credits to Tim Lopez +++++
        cmp     al,0Ah               ;0..9 C; -6A 96..9F C A; -66 30..39
        sbb     al,69h               ;A..F NC; -69 A1..A6 C NA; -60 41..46
        das                                ;read NC = No Carry, C NA = C + No Aux.
        stosb                          ;ASCII result
        pop     ax
        loop    nibble

        xchg    cx,ax                ;restore CX
        pop     ax                    ;restore AX

```

```

        pop     di                ;restore DI
        pop     es                ;restore ES (result pointer DS:DX) +++++
        retn
x2c     endp

pout    proc    near
        push    cs
        pop     ds
        mov     di,20h           ;DI = SPace + NUL
        mov     si,offset oemid
        mov     [si+8],di        ;oemid to ASCIIIZ
        call    show
        pop     si               ;SI = offset code + 3
        call    show
        sub     si,11            ;SI = offset code - 8
        mov     [si+8],di        ;FATxx to ASCIIIZ
        call    show
        mov     [si],di          ;label to ASCIIIZ
        sub     si,11            ;SI = offset code - 19
        call    show
        mov     word ptr [si],"-"
        sub     si,4             ;SI = offset code - 23
        mov     dx,si            ;DX = X2C output buffer
        mov     cx,[si+2]        ;high word volume ID
        mov     ax,[si+0]        ; low word volume ID
        call    x2c              ;high X2C, input: CX
        xchg    ax,cx
        call    show
        mov     [si+4],di        ;hyphen "-" to SPace
        call    x2c              ; low X2C, input: CX
        call    show            ;AL = 0 returned by show
        cbw                     ;AX = 0
        int     16h              ;AH = 0 (press any key)
        int     18h              ;no boot partition
        int     19h              ;no ROM BASIC (joke)
pout    endp

_TEXT   ends
        end     vbr

```