

```

; SERIAL.ASM

; Retro UNIX 8086 v1 Terminal Program (DOS version)
; (Standalone DOS program)
; by Erdogan TAN
; 08/07/2014, 23/07/2013, 27/07/2014
; (06/07/2014, 05/07/2014, 04/07/2014, 03/07/2014)

.8086

CODE_SEG      segment para public
               assume  CS:CODE_SEG, DS:CODE_SEG, SS:CODE_SEG, ES:CODE_SEG

               org 100h

start:
    mov     ax, 0600h ; Scroll up, clear (AL=0)
    mov     bh, 07h   ; Black background (0),
                       ; Light gray foreground (7)
    sub     cx, cx     ; Left-Upper column, row
    mov     dx, 184Fh ; Right-Lower column, row
    int     10h
    ;
    mov     ah, 2      ; Set cursor position
    xor     dx, dx     ; Row 0 (DH), Column 0 (DL)
    xor     bh, bh ; 0
    int     10h
    ;
    mov     si, offset StartMsg
    mov     bl, 7
    call    proc_printmsg
    ;
    xor     ax, ax
    mov     ds, ax
    mov     si, offset 27*4
                       ; INT 1Bh vector
    mov     di, offset old_ctrlbrk
    movsw   ; Save the old ctrl+brk interrupt
    movsw
    push    cs
    pop     ds
    mov     es, ax
    mov     ax, offset ctrlbrk
    mov     di, 27*4 ; INT 1Bh vector - offset
    stosw
    mov     ax, cs
    stosw   ; INT 1Bh vector - segment
    mov     es, ax

@@:
    xor     ah, ah
    int     16h
    ;
    cmp     al, '1'
    je      short @@f
    cmp     al, '2'
    je      short _x
    ;
    mov     al, 07h ; BEEP !
    mov     ah, 0Eh
    int     10h
    jmp     short @b

_x:
    mov     si, offset _3F8h + 1
    dec     byte ptr [SI] ; 2F8h
    add     si, 2
    dec     byte ptr [SI] ; 2F9h
    add     si, 2
    dec     byte ptr [SI] ; 2FAh
    add     si, 2
    dec     byte ptr [SI] ; 2FCh
    mov     si, offset _EFh
    mov     byte ptr [SI], 0F7h

@@:
    sub     al, '1'
    mov     byte ptr [port], al
    ;
    mov     si, offset ComSMsg
    add     byte ptr [SI]+4, al
    ;mov     bx, 7
    call    proc_printmsg

```

```

;
xor     ah, ah
mov     al, 0E3h ; Communication parameters
; 9600 baud, parity none, one stop bit
;xor    dh, dh
mov     dl, byte ptr [port]
int     14h
;
mov     cx, 65535

@@:
nop
nop
nop
loop    @b
;
mov     si, offset AnyKeyMsg
call    proc_printmsg
;
xor     ah, ah
int     16h
;
cmp     al, 1Bh ; ESC key
je      _return
;
mov     ax, 0600h ; Scroll up, clear (AL=0)
mov     bh, 17h ; Blue background (1),
; Light gray foreground (7)
sub     cx, cx ; Left-Upper column, row
mov     dx, 184Fh ; Right-Lower column, row
int     10h
;
mov     ah, 2 ; Set cursor position
sub     dx, dx ; Row 0 (DH), Column 0 (DL)
mov     bx, 7
int     10h
;
mov     dl, byte ptr [port]
; hook serial port interrupt
xor     ax, ax ; 0
mov     ds, ax ; IVT base

mov     si, 0Bh*4 ; Port 1 (COM2)
and     dl, dl ; Port 0 (COM1) ?
jnz     short @f
add     si, 4 ; 0Ch*4 (COM1)

@@:
push     si
mov     di, offset old_serial
movsw
movsw
;
push     cs
pop      ds
mov     es, ax ; 0
pop     di

@@:
mov     ax, offset serial ; serial port interrupt handler
stosw ; INT 0Ch (0Bh) vector - offset
mov     ax, cs
stosw ; INT 0Ch (0Bh) vector - segment
;mov    es, ax
;
mov     dx, word ptr [_3FCh]
;modem control register
in      al, dx ;read register
or      al, 8 ;enable bit 3 (OUT2)
out     dx, al ;write back to register
mov     dx, word ptr [_3F9h]
;interrupt enable register
in      al, dx ;read register
or      al, 1 ;receiver data interrupt enable
out     dx, al ;write back to register
in      al, 21h ;read interrupt mask register
and     al, byte ptr [_EFh]
;enable IRQ 4 (0EFh) (COM1)
; or IRQ 3 (0F7h) (COM2)
out     21h, al ;write back to register
;
sub     al, al ; null

```

```

sendchr:      jmp     short _1    ; (initialization, wakeup signal)

              cmp     byte ptr [cbrk], ah ; ctrl + break
              ja      short _exit

@@:           mov     ah, 1
              int     16h
              jnz     short _0
              hlt
              nop
              nop
              nop
              jmp     short @b

_0:           xor     ah, ah      ; 0
              int     16h        ; Read character

_1:           push    ax

_2:           xor     dh, dh
              mov     dl, byte ptr [port]
              mov     ah, 3
              int     14h
              and     ah, 32      ;transmitter holding register empty
              jnz     short @f     ;yes, ready to send
              hlt                ;no, check status again
              nop
              nop
              jmp     short _2

@@:           pop     ax
              mov     dx, word ptr [_3F8h] ;data port
              out     dx, al       ;send on serial port
              jmp     short sendchr

_exit:        ; Restore old interrupt vectors
              ;xor     ax, ax
              ;mov     es, ax      ; 0
              mov     si, offset old_serial
              mov     di, offset 0Bh*4 ; (COM2)
              dec     byte ptr [port]
              jz      short @f
              add     di, 4        ; 0Ch*4 (COM1)

@@:           movsw                ; Restore
              movsw

_return:      mov     si, offset old_ctrlbrk
              mov     di, offset 27*4
                  ; INT 1Bh vector
              movsw                ; Restore
              movsw
              ;
              int     20h

here:         hlt
              jmp     short here

serial:       ;
              ; INT 0Ch (0Bh) serial port interrupt handler
              ;
              push    ds
              push    ax
              push    bx
              push    dx
              ;
              mov     ax, cs
              mov     ds, ax
              ;
              mov     dx, word ptr [_3FAh]
                  ;interrupt identification register
              in      al, dx        ;read register
              and     al, 0Fh      ;leave lowernibble only
              xor     ah, ah       ; 0
              cmp     al, 4        ;is receiver data available
              jne     short @f     ;no, leave interrupt handler
              mov     dx, word ptr [_3F8h]
                  ;data register

```

```

        in      al, dx      ;read character
        ;
        mov     ah, al
@@:
        mov     al, 20h
        out     20h, al     ;end of interrupt
        ;
        mov     al, ah
        mov     bx, 7
        mov     ah, 0Eh
        int     10h        ; Write character on TTY display
@@:
        pop     dx
        pop     bx
        pop     ax
        pop     ds
        iret

ctrlbrk:
        ;
        ; INT 1Bh (control+break) handler
        ;
        inc     byte ptr CS:[cbrk]
        iret

proc_printmsg:
        mov     ah, 0Eh
        ;mov     bx, 7
@@:
        lodsb
        and     al, al
        jz      short @f
        int     10h
        jmp     short @b
@@:
        retn

StartMsg:
        db 0Dh,0Ah
        db 'Terminal program for Retro UNIX 8086 v1... (27/7/2014)'
        db 0Dh,0Ah
        db 'Press 1 for COM 1 or press 2 for COM2 serial port...'
        db 0Dh,0Ah,0h

ComSMsg:
        db 07h
        db 'COM1 selected...'
        db 0Dh, 0Ah, 0

AnyKeyMsg:
        db 'Press a key to continue.'
        db 0Dh,0Ah,0h

_EFh:   db 0EFh
;
_3F8h:  dw 3F8h
_3F9h:  dw 3F9h
_3FAh:  dw 3FAh
_3FCh:  dw 3FCh

port:   db 0
cbrk:   db 0

old_ctrlbrk: dd 0
old_serial:  dd 0

CODE_SEG ends

        end      start

```