

```

; SCANCODE.ASM

; Modification for Retro UNIX 8086 v1
; 20/01/2014
; 08/12/2013

; (c) Erdogan TAN 1998
; Scancode, prints character codes
; 1/2/1998

.8086

; UNIX v1 system calls
_rele equ 0
_exit equ 1
_fork equ 2
_read equ 3
_write equ 4
_open equ 5
_close equ 6
_wait equ 7
_creat equ 8
_link equ 9
_unlink equ 10
_exec equ 11
_chdir equ 12
_time equ 13
_mkdir equ 14
_chmod equ 15
_chown equ 16
_break equ 17
_stat equ 18
_seek equ 19
_tell equ 20
_mount equ 21
_umount equ 22
_setuid equ 23
_getuid equ 24
_stime equ 25
_quit equ 26
_intr equ 27
_fstat equ 28
_emt equ 29
_mdate equ 30
_stty equ 31
_gtty equ 32
_ilgins equ 33

sys macro syscallnumber, arg1, arg2, arg3
; Retro UNIX 8086 v1 system call.
ifnb <arg1>
    mov bx, arg1
endif
ifnb <arg2>
    mov cx, arg2
endif
ifnb <arg3>
    mov dx, arg3
endif
mov ax, syscallnumber
int 20h
endm

; Retro UNIX 8086 v1 system call format:
; sys syscall (ax) <arg1 (bx)>, <arg2 (cx)>, <arg3 (dx)>

```

```

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

proc_start    proc      far
start:
               mov     si,offset BossMsg
               call    proc_printmsg
               xor     bx, bx

again:
               mov     byte ptr [Character], 20h
@@:
               sys     _gtty, 0, 0
               and     bx, bx
               jz      short @b
               cmp     bl, 0Dh
               je      short pass_enter
               mov     byte ptr [Character], bl

pass_enter:
               mov     al, bh
               call    proc_hex
               mov     word ptr [Reg_ScanCode], ax
               mov     al, bl
               call    proc_hex
               mov     word ptr [Reg_AsciiCode], ax
               mov     si, offset ScancodeMsg
               call    proc_printmsg
               sys     _read, 0, character, 1
               ; bx = 0
               cmp     byte ptr [character], 0Dh
               jne     short again

               sys     _exit

proc_start    endp

;.....;
; From binary (byte) to hexadecimal (character) converter ;
; ;
; input -> AL = byte (binary number) to be converted ;
; output -> AL = First character of hexadecimal number ;
; output -> AH = Second character of hexadecimal number ;
; ;
; (c) Erdogan TAN 1998 ;
;.....;

proc_hex      proc      near

               mov     ah,al
               and     ah,0Fh
               add     ah,30h
               cmp     ah,39h
               jna     short pass1
               add     ah,07h

pass1:
               shr     al,1
               shr     al,1
               shr     al,1
               shr     al,1
               add     al,30h
               cmp     al,39h
               jna     short pass2
               add     al,07h

pass2:
               retn

proc_hex      endp

proc_printmsg proc near
               push    si
               xor     dx, dx

@@:
               lodsb
               and     al, al
               jz      short @f
               inc     dx
               jmp     short @b

```

```

@@:
    pop     cx
    mov     bx, 1
    sys     _write
    retn

proc_printmsg    endp

BossMsg:
    db 0Dh,0Ah
    db '[ (c) Erdogan TAN 1998-2013 ] Press a key to scan code...'
    db 0Dh,0Ah
    db 0Dh,0Ah,0h

ScancodeMsg:
    db 'Character : '
Character:
    db ?
    db '      Scan Code : '
Reg_ScanCode:
    dw ?
    db 'h'
    db '      ASCII Code : '
Reg_AsciiCode:
    dw ?
    db 'h'
    db 0Dh,0Ah,0h

CODE_SEG        ends

                end    start

```