

```
-----  
;  
; OS Interface Specification  
;  
; All OS interaction is done by programs setting up the  
; registers appropriately and invoking RST 08, This file  
; defines the register setups for different calls.  
;  
; Register:  
;  
; C - OS function, all other registers depend on the function  
;  
-----  
  
OSGetRAM = 0 ;get usable RAM boundaries  
; returns DE = start address, HL = end address  
OSGetKeyPress = 1 ;check keyboard for keypress  
; returns A = key press (0x00 if no key available)  
OSGetKybdStatus = 2 ;get keyboard status  
; returns A = status  
OSClearScreen = 3 ;clear the screen  
OSHomeScreen = 4 ;move the screen cursor to home position  
OSCursorFwd = 5 ;move cursor forward  
; input A = #of characters to move forward  
OSCursorBack = 6 ;move cursor backward  
; input A = #of characters to move backward  
OSEnableCursor = 7 ;enable/disable cursor  
; input A = 0->disable, 1->enable  
OSDispChar = 8 ;display a single byte as an ascii character  
; input A = char  
OSDisp2Hex = 9 ;display a single byte as a 2 byte hex value  
; input A = char  
OSDisp4Hex = 10 ;display a word as a 4 byte hex value  
; input DE = word  
OSNewLine = 11 ;pad out the current line and move the cursor  
OSSetVMode = 12 ;set video mode raw/terminal  
; input A = 0->terminal (default) / 1->raw  
  
OSReadSector = 13 ;read a disk sector  
OSWriteSector = 14 ;write a disk sector  
; input A = drive (0-A:, 1-B:, 2-C:, ...) - CURRENTLY ONLY HARD DRIVE  
; HL = pointer to buffer  
; DE = D0:D16 of sector number  
; output A = 0 on success, !0 = error code  
OSDiskInfo = 15 ;get the supported disk info
```

```
; input A = drive (0-A:, 1-B:, 2-C:, ...)  
; output A = 0 on success, !0 = error code  
;         BC = last usable sector number  
;         DE = #of bytes per sector
```