

Errata: FYSOS: The System Core  
1st Edition

Dated: 16 Aug 2016

Page 4-3: Table 4-1 and the first check mark below it

The signature is: 0x5452415020494645

not: 0x5252415020494645 (highest byte in error)

Page 4-4: Note wrongly states that the specification does not require you to backup the Partition Entries. Version 2.5 of the specification states that you must backup the entries as well.

Page 6-13: Listing 6-4 (end of)

```
; -----  
; let's make sure we made it here  
; if still in 16bit mode, the below code  
; will look like this -----> ; xor  ax,ax  
xor eax,eax          ; xor  ax,0000h  
xor eax,0C08B0000h  ; mov  ax,ax  
jz  short still_in_16bit ; jz  short still_in_16bit
```

The reference of 0C08B00h should be 0C08B000h (and 0xC08B000) in this listing and all references below it. Sorry for the missing zero.

The last part of the paragraph below the listing should now read:

Therefore, the two XOR instructions will mean something different in 16-bit code because of the length of the immediate 0xC08B0000. In 16-bit code, the XOR instruction will only use the 0x0000 portion of the immediate and the 0xC08B portion will now be a new instruction, MOV AX,AX.

If you have the CDROM, please update the loader.asm source file in the main\syscore\source\loader folder.

Change line 893:

```
xor eax,0C08B000h ; mov  ax,ax
```

to:

```
xor eax,0C08B0000h ; mov  ax,ax
```

Page 7-4: Fifth paragraph:

"After setting bit 2 of the byte read,"  
should be

"After setting bit 1 of the byte read,"  
(meaning: the second bit, or a value of 02.)

Page 7-9: Listing 7-4: Cont...

```

...
cmp al,0FAh ; was it the ACKnowledge byte?
mov ah,al
jne short method_thr_done ; return if not
...

```

The "mov ah,al" is a remnant of a different function. It is not needed, though has no effect either way.

Page B-4:

D:\FYSOS\MAIN\SYSCORE\UTILS\UTILS\A20\>A20<enter>  
should be

D:\FYSOS\MAIN\SYSCORE\UTILS\A20\>A20<enter>

D:\fysos\main\syscore\source\embr

embr.asm needs the following additions:

```

----- line 517: -----
                ; down key
                cmp  ax,5000h
                jne  short not_down
                mov  ax,cur_selected
                sub  ax,cur_start
+               ; check to see if at last one
+               mov  bx,tot_entries
+               dec  bx
+               cmp  ax,bx
+               je   do_again
+               ; else see if we can scroll the screen
                cmp  ax,(TOTAL_DISPLAY-1)
                jae  short @f

----- line 1004: -----
disp_it1:      ; di -> entry
                ; clear the display box
                mov  cx,(TOTAL_DISPLAY<<1)
                mov  word row,3
+               mov  byte color,07h
@@:           mov  word col,7
                mov  si,offset blank_line

```

CDROM: Utilities: MFYSFS

The MFYSFS utility on the CDROM incorrectly calculates the amount of needed sectors for the bitmap(s). There is nothing wrong with it, since it compensates correctly. However, it wastes a few sectors with larger than necessary bitmaps. You simply need to divide the number of sectors it calculates by the number of sectors per cluster. I missed adding that simple divide :-)

CDROM: Source: eMBR

There was an error in the source code of the eMBR example. I have fixed this error and added a little more code. To get the latest source, go to:

[http://www.fysnet.net/fysos\\_embr.htm](http://www.fysnet.net/fysos_embr.htm)

MSECTCPY.CPP

The "usage" comment at the top of msectcpy.cpp is in error. It should read:

MSECTCPY image\_file\_name.img file\_to\_copy.bin 0