Memory Segmentation

x86 16-bit Real Mode

Memory Segmentation exist to allow for access over 64kb with 16-bit registers. This allows up to 1Mb of memory to be addressed in Real mode.

Segment Registers

Register	Name	Intended Purpose
CS	Code Segment	CPU fetches instructions from this segment
DS	Data Segment	Any data reference that does not use the stack
SS	Stack Segment	Any push or pop or data referencing the stack uses this segment
ES	Extra Segment	Default destination for string operations
FS	F Segment	No hardware defined use / Software Defined
GS	G Segment	No hardware defined use / Software Defined

Segment Address Translation

Say you have the following code:

```
mov ax, [es:di]
```

To find the linear address would be the equivalent of (es << 4) + di.

Example Math:

```
Segmented address 0x0045:0x5678 == Linear Address 0x05ac8

    0x00450  Segment 16-bits shifted left 4 bits
+    0x5678  Offset 16-bits
------------
    0x05ac8  Linear Address 20-bits
```

Example use:

```
mov ax, 0x6a78          ; Set segment value
mov es, ax              ; Segment registers can not be accessed directly.
mov di, 0x0003          ; Set the segment offset to 3.
mov ax, [es:di]         ; Put the value at the segmented address [0x6a78:0x0003] or
                        ; linear address [0x6a783] in ax.
```