

## Re: can somebody help me with the problem with tasm models

*Source:* <http://coding.derkeiler.com/Archive/Assembler/alt.lang.asm/2005-03/0357.html>

---

*From:* Beth (*BethStone21\_at\_hotmail.NOSPICEDHAM.com*)

*Date:* 03/09/05

Date: Wed, 09 Mar 2005 15:37:34 GMT

darkie wrote:

> *I am using tasm and using the assembler in dos ( real mode ). I am  
> quite confused with the .model directive. why exactly is the .model  
> tiny or large etc are used?*

Long story; When Intel created the x86 originally, it had 16-bit registers...now, when addressing memory with something like "[ bx ]", this is using the value of BX as an `_address_` to fetch values out of memory...memory itself is "byte addressable", meaning, basically, that it's just a long "array of bytes", each byte numbered from zero upwards throughout memory (each byte is like a "house" on a street, each house numbered from zero upwards until the end of the street, which is its "address" that pop onto the envelope to get your mail delivered to the right "house" :)...)

Right, if each byte in memory is given an "address", starting from zero and moving upwards by one for each byte, then, as the register only has 16-bits, there's only so much memory we can "address" with 16-bits...specifically, 16-bits can only cover a range of (integer) numbers from 0 to 65535 (more generally, for n bits, the range is 0 to  $(2^n)-1$ ...the "2" because each bit is binary, of course, to the power of how many bits and "minus one" because you're starting counting from zero, not one...there's  $2^n$  possible values but, because we start at zero, the "maximum value" is one less than that at  $(2^n)-1$  ;)...)

As each address is referring to a byte, then that's 65536 bytes ( $65536 * 8 = 524,288$  bits but, well, we never address bits directly but work with bytes instead :)...or, dividing by 1024 (the amount of bytes in a kilobyte :), then that's 64KB...

So, with only one 16-bit register, you can only address 64KB (with "byte addressable" memory but, basically, that's normally almost always the case, it's usually "taken as read" :)...well, this range was a bit "limited"...Intel wanted the PC to be able to have more than 64KB of RAM installed...

alt.lang.asm: Re: can somebody help me with the problem with tasm models

Therefore, Intel simply decided to use two 16-bit registers for "addressing"...more bits and you can "address" more memory...so, there could be one 16-bit "segment" register and another 16-bit "offset" register...and "addresses" can be composed of these two together as