

Forum:[Hack Forums](#)

Thread: [BIOS 16 bit to 32 bit](#)

bytekill - May 2, 2012, 11:18 pm

Hi.

I'm working on a small operating system, my problem is that my OS only works with 16 bit code and not registers that are 32 bit (like eax, ebx...).

I made up a pretty good boot loader in 16 bit that loads my kernel.

I want my kernel to change from "Real mode" to "Protected mode". So I search for it and found this code:

Code `mov ax, cr0` ; Moves the value

or `ax, 1` ; changes bit 1

`mov cr0, ax` OK so my problem here is that the assembler I'm using (NASM) gives me two error lines:

kernel.asm:59: error: invalid combination of opcode and operands

kernel.asm:61: error: invalid combination of opcode and operands

thanks in advance.

MUT4NT - May 3, 2012, 12:08 am

SO:

`mov ax, [cr0]`

or `ax, 1`

`mov [cr0], ax`

in real-address mode, it works with immediate values, you are confused.

and yes, it is possible to work with 32-bit registers in real-address mode but all depends on the situation

DexHacker - May 20, 2012, 7:07 am

I would advise you to switch to Fasm and before people jump on my back, there are reasons.

First fasm assembler was made to write an OS, so has many cool things to help you.

Second it is very easy to port to your OS (let's say half hour, once you have the right function)

To port nasm you would need to port gcc.

If you're going to switch to C code after booting, then stick with nasm.

Hostwinds - June 13, 2012, 3:42 pm

Bytekill, Check out <http://forum.osdev.org/> there are many good OS devs there as well as a wealth of knowledge in their wiki. Goodluck!