

Descriptor Cache

From OSDev Wiki

To allow for fast accesses to segmented memory, the x86 processor keeps a copy of each segment descriptor in a special descriptor cache. This saves the processor from accessing the GDT for every memory access made. The workings of this cache allows for some interesting side effects.

Contents

- 1 The cache
- 2 Unreal Mode
- 3 Other cache settings
- 4 Modifying the GDT

The cache

The cache for each segment selector (CS, SS, DS, ES, FS, GS) contains all of the bits and fields you would also find in the GDT. These include descriptor type, access rights, base and limit. In protected mode these fields are filled from the GDT or LDT whenever there is written to a segment selector (both implicitly and explicitly). In real mode, the processor generates entries internally as there is no GDT involved in this mode. Interestingly, not all fields are updated in real mode. Most notably that includes the segment limit.

Unreal Mode

Unreal Mode is a 'mode' where the processor runs in real mode while the segment limit does not equal 64KB (in most cases, its 4GB). Since real mode doesn't update the limit field, this state persists across segment register loads. Entering this mode is achieved easily by entering protected mode (where the limit can be changed), load the desired limit into the descriptor cache, then switch back to real mode.

Other cache settings

Other settings for the descriptor cache are also used in the x86 architecture. When the processor boots it enters real mode, but it starts executing at 0xffffffff by having a high base value loaded in the descriptor cache for CS. System management mode also loads non-standard values into the cache.

Modifying the GDT

Since each segment register holds a copy of the GDT's contents, updating the GDT will not change the cache even if that selector is currently in use. Therefore one needs to reload the segment registers to make sure the correct values are stored in the cache. This independence can also be used to discard the GDT without breaking the system. GRUB will for instance leave you with a known contents in the descriptor caches, but without GDT. To make sure things remain safe, interrupts will have to be disabled since an interrupt or exception will implicitly reload CS. If the GDT has been overwritten with other data, you are almost guaranteed to get a Triple Fault

Retrieved from "https://wiki.osdev.org/index.php?title=Descriptor_Cache&oldid=7000"

Category: X86 CPU

-
- This page was last modified on 6 January 2009, at 10:52.
 - This page has been accessed 16,060 times.