

The Linear Address Space

Source: <http://coding.derkeiler.com/Archive/Assembler/comp.lang.asm.x86/2004-04/0350.html>

From: Cameron Gibbs (*odly_at_hotmail.com*)

Date: 04/27/04

Date: Tue, 27 Apr 2004 07:43:16 +0000 (UTC)

Hey people.

I have been working through the IA-32 Software Developer's Manuals. I have read pretty much all of it in some detail (skipping mostly just tables and lists that can be referenced as needed). Still there is something very basic in my understanding that is lacking.

There is a Physical Address Space of 4GBytes (2^{32} bytes) which basically identifies all of the binary numbers that can be used as addresses for assembly language programming.

With the 36-bit address extensions it is possible to increase this to 64Gbytes.

The problem is that in a computer there is Hard Drive storage, RAM storage and a heap of other devices and the like that all require byte addresses.

Say for example you had a 180GByte HDD in your system, even with the 36 bit address extension mechanisms which bring the Physical Address Space up to 64GBytes, there are still just not enough Logical Addresses to reference the whole of the addresses on a system.

Also with Hard Drive partitions and with every program having to be put in a particular spot on the Hard Drive, it is a very rare thing that one would write a program that is to be put at address zero.

>>*From all this I have come to think that the Linear Address Space is simply an abstraction or a virtualization of the true state of all of the addresses within a computer so what I need to know is how are these abstractions that are used in a program "mapped" to the real addresses of the computer? How does this work?*

Also, I think that paging has something to do with RAM management and the "checking in" and "checking out" of data to and from RAM but frankly I can neither confirm nor deny this and hence I just cannot figure out what is actually going on with paging.

Am I on the right track with paging? – Please correct!

Might the mapping to real addresses have something to do with the Page Directory Base Register (PDBR) which is Control Register 3 (CR3)?

All help appreciated.

Cameron.

--

Reply to the group.

Procedurized mind at work.

Adhere to the sequence.

The PC & Electronic Data Map Project:

<http://members.dodo.com.au/~cameron111>

Outgoing mail is certified Virus Free.

Checked by AVG anti-virus system (<http://www.grisoft.com>).

Version: 6.0.658 / Virus Database: 421 - Release Date: 4/9/04