

Re: [OT] Documentation for PC Architecture

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-08/5096.html>

From: Nagendra Singh Tomar (*nagendra_tomar_at_adaptec.com*)

Date: 08/18/03

Date: Mon, 18 Aug 2003 22:04:35 +0530 (IST)

To: Jamie Lokier <jamie@shareable.org>

Jamie,

What is this 2nd address translation you are mentioning.

I always thought that for the sake of cleanliness we just forget about the 384K of memory starting from 640K. RAM anyway is cheap.

Pl correct me if I'm missing something.

Thanx

tomar

On Tue, 19 Aug 2003, Jamie Lokier wrote:

> *Herbert Pötzl wrote:*

> > *On Mon, Aug 18, 2003 at 11:11:53PM +0200, Paolo Ornati wrote:*

> > >

> > > > *Curiosity: since the memory addresses from 640KB to 1MB are reserved for*

> > > > *memory mapped I/O (video memory) and BIOS ROM... the correspondent range*

> > > > *in*

> > > >

> > > > *uh oh ...*

> > > >

> > > > *the REAL MEMORY isn't usable and so we lost 384KB of memory. Is this*

> > > > *right?*

> > > >

> > > > *for DOS, withouth upper memory manager yes ;)*

> > >

> > > *I'm talking about an OS in protected mode... in any case how*

> > > *can I access to this memory region if it's mapped for other things?*

> >

> > *this is usually done via the MMU which can do miraculous*

> > *things to memory and addresses ...*

> >

> > > *I've done some tests with a simple kernel which I wrote: all that region*

> > > *(except video memory at 0xb8000) results "read only"...*

Linux-Kernel: Re: [OT] Documentation for PC Architecture

> >
> > *because it is usually designated as rom area, which naturally*
> > *is read only ...*
> >
> > > *So I THINK YOU mean: "you can use more than 640KB in real mode using*
>
> > > *a memory manager that "remap" 0xC0000 (for example) to 0x100000 or*
> > > *something like it"*
> >
> > *basically a memory manager (depending on the processors*
> > *capabilities) does either use memory mapping or simple*
> > *memory exchanging between designated regions and above*
> > *1M+64k memory ...*
>
> *The MMU _is_ used to remap memory addresses. It is part of the CPU*
> *itself.*
> *But it translates what's called "virtual address" space to "physical*
> *address space". Physical addresses seemingly map directly to RAM and*
> *memory-mapped I/O.*
>
> *Paolo's question is, what happens to the 384k of _physical_ addresses*
> *starting at 0xa0000, which should correspond with 384k of actual*
> *physical RAM?*
>
> *If you use the MMU to map a virtual address to the physical addresses*
> *from*
> *0xa0000..0xfffff, then whichever virtual addresses you chose will map to*
> *video memory, ROM, BIOS etc.*
>
> *The answer is that after the MMU has translated, a _second_ address*
> *translation takes place, outside the CPU, which maps the physical*
> *addresses*
> *so that a hole is created in the RAM without any RAM going missing.*
> *This*
> *second translation is done by the motherboard chipset.*
>
> *Enjyo,*
> *-- Jamie*
>
> -
> *To unsubscribe from this list: send the line "unsubscribe linux-kernel"*
> *in*
> *the body of a message to majordomo@vger.kernel.org*
> *More majordomo info at <http://vger.kernel.org/majordomo-info.html>*
> *Please read the FAQ at <http://www.tux.org/kml/>*
>

-
To unsubscribe from this list: send the line "unsubscribe linux-kernel" in
the body of a message to majordomo@vger.kernel.org
More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Linux-Kernel: Re: [OT] Documentation for PC Architecture

Please read the FAQ at <http://www.tux.org/lkml/>