sign up log in tour help stack overflow careers

Stack Overflow is a question and answer site for professional and enthusiast programmers. It's 100% free, no registration required. Take the 2-minute tour ×

Memory mapped IO - how is it done?

I've read about the difference between port mapped IO and memory mapped IO, but I can't figure out how memory mapped Io is implemented in modern operating systems (windows or linux)

What I know is that a part of the physical memory is reserved to communicate with the hardware and there's a MMIO Unit involved in taking care of the bus communication and other memory-related stuff

How would a driver communicate with underlying hardware? What are the functions that the driver would use? Are the addresses to communicate with a video card fixed or is there some kind of "agreement" before using them?

I'm still rather confused

operating-system linux-kernel kernel device-driver

asked Mar 11 '12 at 11:25

paulAl
359 4 12

Have you looked at Using I/O Memory and Memory Mapping and DMA? – Appleman1234 Mar 11 '12 at 11:38

Let's see if I got it(in linux): 1) I assign a memory IO area at the address the hardware producer told me to 2) I use ioremap to get it translated from physical addresses to virtual addresses (so this resolves both segmentation and paging I suppose) 3) I use iowrite/ioread and similar to read and write in that area. Is this correct? – paulAl Mar 11 '12 at 11:58

That is correct, remember to free to allocate memory region after usage. – Appleman1234 Mar 11 '12 at 12:12

3 Answers

The following statement in your question is wrong:

```
What I know is that a part of the physical memory is reserved to communicate with the hardware
```

A part of the physical memory is **not** reserved for communication with the hardware. A part of the physical **address space**, to which the physical memory and memory mapped IO are mapped, is. This memory layout is permanent, but user programs do not see it directly - instead, they run into their own **virtual address space** to which the kernel can decide to map, wherever it wants, physical memory and IO ranges.

You may want to read the following articles which I believe contain answers to most of your questions:

- http://duartes.org/gustavo/blog/post/motherboard-chipsets-memory-map
- http://duartes.org/gustavo/blog/post/memory-translation-and-segmentation
- http://duartes.org/gustavo/blog/post/how-the-kernel-manages-your-memory

answered Mar 12 '12 at 2:22

Gnurou
1,889 1 7 17

Thank you, I read them all and I found them useful to my purposes – paulAl Mar 16 '12 at 21:32

Work on work you love. From home.

stackoverflowcareers

Are you asking about Memory mapped files, or memory mapped port-IO?

Memory mapped files are done by paging out the pages and intercepting page-faults to those addresses. This is all done by the OS by negotiation between the file-system manager and the page-fault handler.

Memory mapped port-IO is done at the CPU level by overloading address lines as port-IO lines which allow writes to memory to be translated onto the QPI bus lines as port-IO. This is all done by the processor interacting with the motherboard. The only other thing that the OS needs to do is to tell the MMU not to coalese reads and writes through the PAE must-writethrough and no-cache bits.

answered Mar 11 '12 at 18:18


SecurityMatt
3,645 6 19

---

http://en.wikipedia.org/wiki/Memory-mapped_I/O

http://www.cs.umd.edu/class/sum2003/cmsc311/Notes/IO/mapped.html

Essentially it is just a form of accessing the data, as if you are saving / reading from the memory. But the hardware will snoop on the address bus, and when it sees the address targetting for him, it will just receive the data on the data bus.

edited Jul 23 '14 at 16:07

answered Jul 23 '14 at 15:53


Peter Teoh
1,165 7 16
