

Stýrikerfi

Verkefni 26

Herbert Vilhjálmsson hev12@hi.is

Consider the connection of an input/output (I/O) device (or it's device controller respectively) to the main memory:

Spurning 1

Briefly describe why I/O is a bottleneck with respect to the overall performance of a computer system!

Lausn: Because it has to go through a few layers of controls and communications but most importantly just because of how incredibly slow it is compared to the CPU. Palli told us in Tölvunarfræði 2 that it was like sitting on the couch and wanting a chocolate, if it's in the CPU cache then it's like just picking it up and putting it in your mouth, but if you have to get it from the hard disk then it would be like calling someone and asking for the chocolate and it would arrive at your place in about 18 months or so. Also we saw in the last assignment how many operations the CPU could perform in the time it takes to look stuff up at the hhd.

Spurning 2

Briefly describe why in a multiprogramming system Direct Memory Access (DMA) helps (in comparison to Programmed I/O (PIO)) to reduce the I/O bottleneck!

Lausn: It helps because the CPU can just load up the instructions once to the DMA controller and then it can go about doing it's business while the controller is loading in data from the I/O device into memory. Once the controller is done loading the data in, it will send a single interrupt to the CPU and the CPU can decide if it needs another instruction to load more stuff or just fetch the data from memory.

Spurning 3

Briefly describe why in a monoprogramming system DMA does not reduce the I/O bottleneck!

Lausn: I would assume it doesn't do anything because as the name implies ,mono, it is only one thing running on the system, so if you need to load stuff in from the I/O device you don't have anything else to do while you wait for that data to be loaded. So if you are waiting for data and don't have anything else to do then the CPU is just sitting there idle and in that case DMA makes no difference and might just be a waste of money since we might as well just have PIO and let the CPU do the work instead of spending money on stuff that gives us noting in return.

If PIO is used, two approaches are possible to find out whether data needs to be transferred by the OS's device driver between the I/O device controller and the RAM:

- Polling (the CPU asks periodically the I/O device controller (busy waiting))
- Interrupt (the I/O device controller raises an interrupt)

Which approach should be used if

Spurning 4

the I/O device is fast? Justify your answer!

Lausn: If the I/O device is fast it should use the Interrupt, since it can just let the CPU know when it needs attention and then load the data in quickly and the CPU doesn't need to "waste" time checking status register and will just react to when the interrupt comes.

Spurning 5

the I/O device is slow? Justify your answer!

Lausn: I would say that a slow I/O device should use Polling, then it can write in a bunch of data on its slow pace and the CPU can just check every now and then if there is stuff or not there instead of waiting endlessly while the I/O device slowly feeds it data.