【OS】Day39

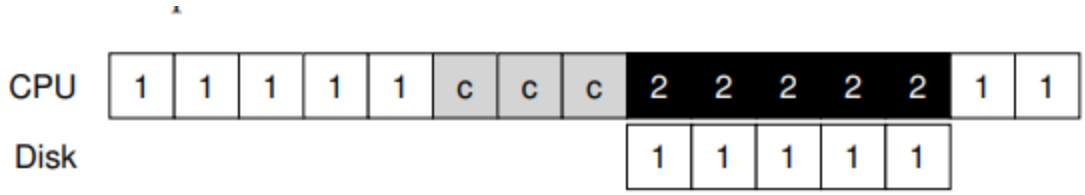
Class	Operating System: Three Easy Pieces
Date	@February 20, 2022

【Ch36】I/O Devices(2)

36.5 More Efficient Data Movement with DMA

When using programmed I/O(PIO) to transfer a large chunk of data to a device, the CPU is once again overburdened with a rather trivial task, and thus wastes a lot of time and effort that could better be spent running other processes.

The timeline illustrates the problem:



In the timeline, Process 1 is running and then wishes to write some data to the disk. It then initiates the I/O, which must copy the data from memory to the device explicitly, one word at a time. When the copy is complete, the I/O begins on the disk and the CPU can finally be used for something else.

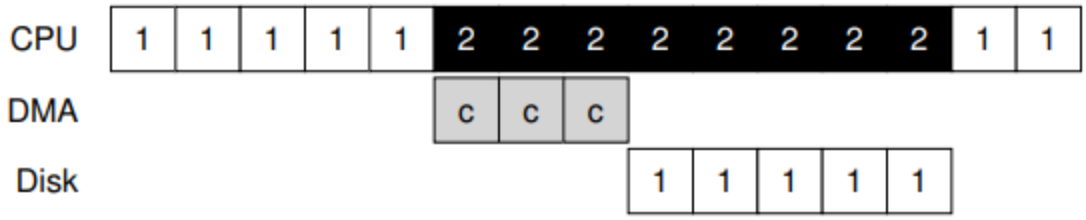
The solution to this problem is something we refer to as Direct Memory Access(DMA). A DMA engine is essentially a very specific device within a system that can orchestrate transfers between devices and main memory without much CPU intervention.

DMA works as follows:

- To transfer data to the device, the OS would program the DMA engine by telling it where the data lives in memory, how much data to copy, and which device to send it to.

- At that point, the OS is done with the transfer and can proceed with other work.
- When the DMA is complete, the DMA controller raises an interrupt, and the OS thus knows the transfer is complete.

The revised timeline:



From the timeline, we can see that the copying of data is now handled by the DMA controller.