I/O Systems

Practice Exercises

13.1 State three advantages of placing functionality in a device controller, rather than in the kernel. State three disadvantages.

13.2 The example of handshaking in Section 13.2 used 2 bits: a busy bit and a command-ready bit. Is it possible to implement this handshaking with only 1 bit? If it is, describe the protocol. If it is not, explain why 1 bit is insufficient.

13.3 Why might a system use interrupt-driven I/O to manage a single serial port, but polling I/O to manage a front-end processor, such as a terminal concentrator?

13.4 Polling for an I/O completion can waste a large number of CPU cycles if the processor iterates a busy-waiting loop many times before the I/O completes. But if the I/O device is ready for service, polling can be much more efficient than is catching and dispatching an interrupt. Describe a hybrid strategy that combines polling, sleeping, and interrupts for I/O device service. For each of these three strategies (pure polling, pure interrupts, hybrid), describe a computing environment in which that strategy is more efficient than is either of the others.

13.5 How does DMA increase system concurrency? How does it complicate hardware design?

13.6 Why is it important to scale up system bus and device speeds as the CPU speed increases?

13.7 Distinguish between a STREAMS driver and a STREAMS module.