Homework 1

- Test your understanding -

Prepared by

Lee Chang Yoon

32183641@dankook.ac.kr

September 7, 2021

Executive Summary

This report is for homework 1 for the system programming class. The main goal for this report is to explain how the hardware (CPU, memory, I/O devices) operates & collaborates in your computer under the following cases: Reading the first 512 bytes of the disk, getting the typed in keyboard value, and sending 1KB of data through the network card.

Case 1

Reading the first 512 bytes of the disk.

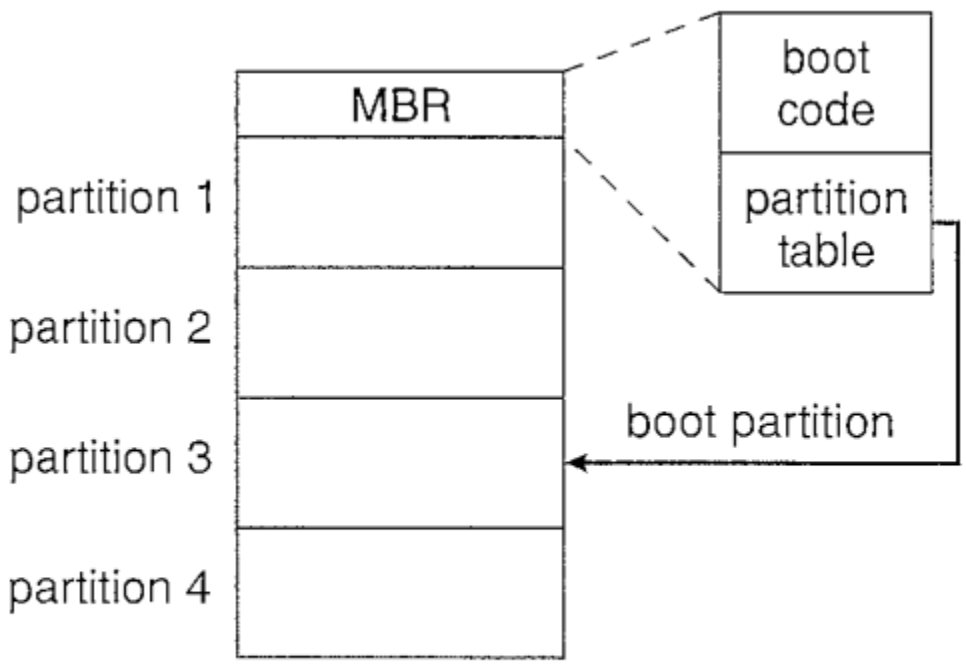

Figure 12.9 Booting from disk in Windows 2000.

Let's assume that the computer has only one HDD (Hard disk drive) that contains OS (Operating System), and the size of a sector of the disk is 512 bytes (Normally, the size of a sector is 512 bytes). Then, the first 512 bytes of the disk is now the first sector of the disk.

For most computers, when the computer turns on, the CPU reads the 0th address of the memory. The 0th address of the memory is ROM (Read-Only Memory), which contains data to run the computer. The code in the boot ROM instructs the disk controller to read the boot blocks into the memory (no device drivers are loaded at this point) and then starts executing that code. This code directs the system to read the boot code from MBR (Master Boot Loader). In addition to containing boot code, the MBR contains a table listing the partitions for the hard disk and a flag indicating which partition the system uses to be booted from, as illustrated in Figure 12.9. Once the system identifies the boot partition, it reads the first sector from that partition (boot sector) and continues with the remainder of the boot process, which includes loading the various subsystems and system services.

In short, when the computer reads the first 512 bytes of the disk, Boot Code in MBR is executed. Then the system identifies the boot partition, reads the first sector from that partition, and the computer starts to boot.

Case 2

Getting the typed in keyboard.

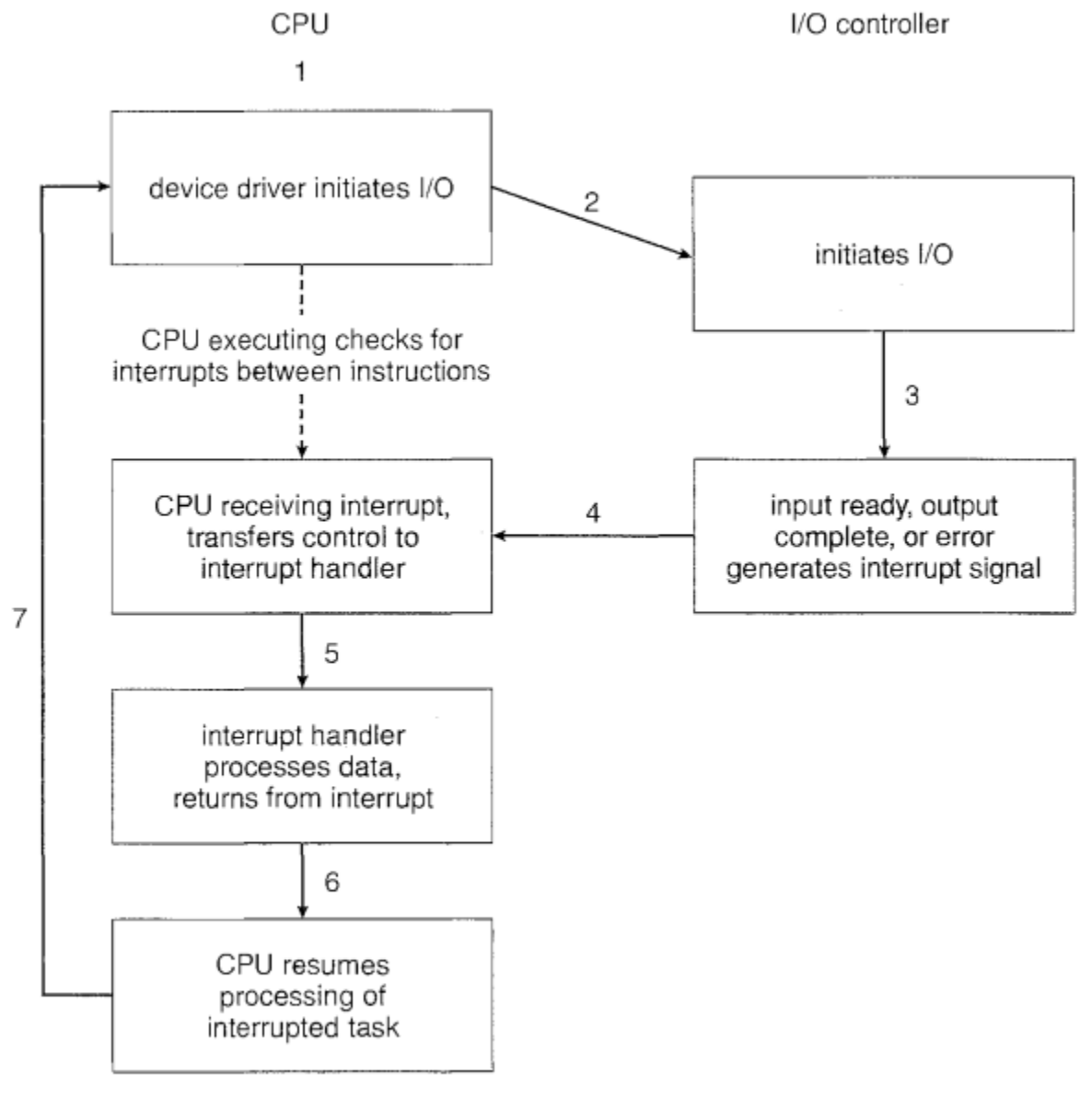

Figure 13.3 Interrupt-driven I/O cycle.

Most keys on the keyboard have a single numerical code in the 01-127 range. When we press down a key, the physical keypresses on the membrane below it creates an electrical connection, which gets detected by the keyboard controller, then transmits that specific numerical value as an electric impulse to the actual computer. However, some keys, such as special commands, work differently: they send more than one numerical value when pressed released.

Let's say we typed the keyboard and submitted some values to the actual computer. The computer's keyboard controller monitors the signal from the keyboard. Keyboard controller processes all of the data that comes from the keyboard and forwards it to the operating system. When the OS notifies that there is data from the keyboard, it checks if the keyboard data is a system-level command. Then, the OS passes the keyboard data onto the current application. The application determines whether the keyboard data is a command, such as Ctrl + s, which stores the file in the memory. If the keyboard data is not a command, the application accepts it as content, which can be anything like typing documents. If the current application does not accept keyboard data, it simply ignores the information. What if the keyboard data is a system-level command? If the current keyboard data is a system-level command, the keyboard controller generates an interruption. When the CPU detects that a controller has asserted a signal on the interrupt-request line, the CPU performs a state save and jumps to the interrupt-handle routine at a fixed address in memory. Then, the I/O cycle in Figure 13.3 occurs according to the typed command.

As a result, when we type some values through the keyboard to the actual computer, the keyboard controller gets the signal from the keyboard and OS determines whether a keyboard data is a system-level command or not. If keyboard data is not a command, it sends it to the application; the application identifies current data and deals with it. However, if the keyboard data is a system-level command, the keyboard controller generates an interruption, and the CPU deals with it.

Case 3

Sending 1KB of data through the network card.

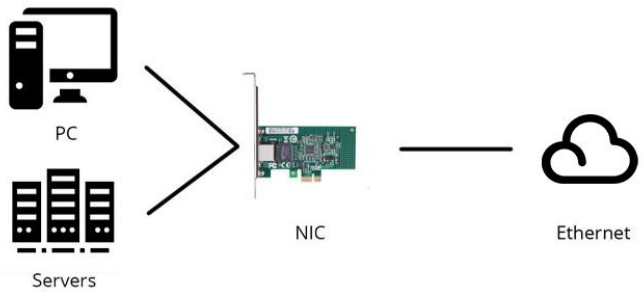

Figure 1

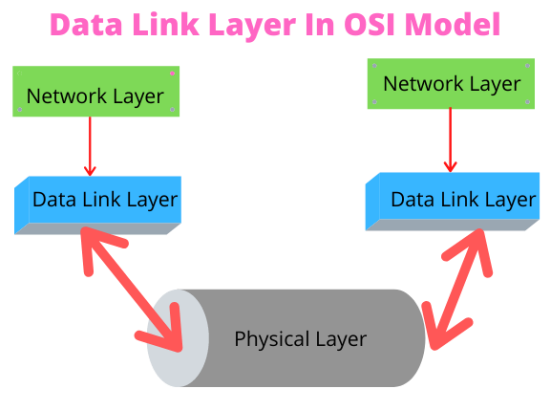

Figure 2

Let's talk about the physical and data link layer in OSI (Open Systems Interconnection Reference Model) layers before going through case 3. In Figure 2, the physical layer only transfers data through the communication cable. This layer does not care about what the data is and if it has an error or not. The data link layer checks the error and re-transfers transmitted data. In this layer, it communicates with the MAC (Media Access Control). NIC provides a computer with a dedicated, full-time connection to a network by implementing the physical layer necessary for communicating with a data link layer standard, such as Ethernet or Wi-Fi. Each card represents a device and can prepare, transmit and control the flow of data on the network.

The network card operates as a middleman between a computer and a data network. When we send 1KB of data, the computer will receive electrical impulses by sending the 1KB of data to the network card. Then, those impulses are received by the destination computer.

In short, when we send 1KB of data through the network card, the computer generates electrical impulses by sending the 1KB of data. Impulses will be received to the network card, and the network card will receive the electrical impulses to the destination computer.

Conclusion

By checking case1, case2, and case3, we can find out how the computer boots, takes input data from the user by peripherals and transfers the data files from the computer. There are more points to consider about how the computer operates in those cases. However, even though we have not considered the details, we have learned how the computer operates in some of the points.