**9.7 - What is polling used for? What are the disadvantages of polling? What is a better way to perform the same job?**

*Answer:*

Polling is the process where the computer or controlling device waits for an external device to check for its readiness or state, often with low-level hardware. For example, when a printer is connected via a parallel port, the computer waits until the printer has received the next character. These processes can be as minute as only reading one bit. This is sometimes used synonymously with 'busy-wait' polling. In this situation, when an I/O operation is required, the computer does nothing other than check the status of the I/O device until it is ready, at which point the device is accessed. In other words, the computer waits until the device is ready.

Disadvantages of polling are if there are too many devices to check, the time required to poll them can exceed the time available to service the I/O device.

Polling also refers to the situation where a device is repeatedly checked for readiness, and if it is not, the computer returns to a different task. Although not as wasteful of CPU cycles as busy waiting, this is generally not as efficient as the better option to polling, interrupt-driven I/O. By having the I/O device initiate communication to the CPU through an interrupt, it frees the CPU to perform other tasks. Sol: Student responses could be more creative, but generally, if user input is expected by an application, then the application would stall waiting for the buffer to fill up.  By using interrupt facility and special commands to inform the interface to issue an interrupt request signal whenever data is available from any device. In the meantime, the CPU can proceed for any other program execution.

**9.12 - In general, what purpose does an interrupt serve? Stated another way, suppose there were no interrupts provided in a computer. What capabilities would be lost?**

*Answer:*

The I/O device does not have direct access to the memory unit. A transfer from I/O device to memory requires the execution of several instructions by the CPU, including an input instruction to transfer the data from device to the CPU and store instruction to transfer the data from CPU to memory. In programmed I/O, the CPU stays in the program loop until the I/O unit indicates that it is ready for data transfer. This is a time-consuming process since it needlessly keeps the CPU busy. This situation can be avoided by using an interrupt facility.

By having the I/O device initiate communication to the CPU through an interrupt, it frees the CPU to perform other tasks. Sol: Student responses could be more creative, but generally, if user input is expected by an application, then the application would stall waiting for the buffer to fill up. This keeps the CPU busy unnecessarily which is usually solved by interrupts but if computer are not provided by interrupts it would face many problems like handling many processes at a time. By using interrupt facility and special commands to inform the interface to issue an interrupt request signal whenever data is available from any device.

While, in the meantime the CPU can proceed for any other program execution.

The interface meanwhile keeps monitoring the device. Whenever it is determined that the device is ready for data transfer it initiates an interrupt request signal to the computer. Upon detection of an external interrupt signal the CPU stops momentarily the task that it was already performing, branches to the service program to process the I/O transfer, and then return to the task it was originally performing.

**Chapter 9 Calculation Exercise***” If my CPU runs at 4.0GHz, and on average takes 10 clock cycles to complete an instruction, how many instructions will be completed in the time it takes to type "MY CPU IS RUNNING NOW"? Assume it takes 5 seconds to type the message. Show your work and how you arrived at the solution”.*

*Answer:*

Given that we have CPU speed at 4.0GHz.

At this speed the clock will tick 4 billion time per second.

Given that on average it takes 10 clock cycles to complete an instruction so we divide the clock speed by 10 clock cycles.

Therefore,

Number of instruction per second = (4 billion cycle per second)/ 10 cycles per instructions

= 400 million instruction per second.

Now, if it takes 5 seconds to type the message

Total number of instructions = (400 million instruction per second) \* 5 seconds

= ***2 billion instructions.***

*References:* Englander, I., & Wong, W. (2021). The architecture of computer hardware, systems software, and networking: An information technology approach. John Wiley & Sons.

*https://en.wikipedia.org/wiki/Polling\_(computer\_science)*

[https://www.geeksforgeeks.org/io-interface-interrupt-dma-mode/](https://www.geeksforgeeks.org/io-interface-interrupt-dma-mode/%20)