# What is a Processor

* The processor is the imaginary brain of the whole computer.
* By choosing the right model, you can speed up demanding applications, squeeze the most out of your graphics card and minimise power consumption.
* Keep in mind that the CPU socket must match the socket on your motherboard.

# Definition of interruption

* An interrupt is a hardware mechanism that allows the CPU to detect that a device needs its attention.
* The CPU has an interrupt request wire line which is checked by the CPU after each individual instruction is executed.
* When the CPU detects an interrupt signal on the interrupt request line, the CPU stops its currently executing task and responds to the interrupt by I/Oing the device by passing control to the interrupt handler.
* The interrupt handler resolves the interrupt by servicing the device.
* Although the CPU does not know when an interrupt would occur, since it can occur at any time, it must respond to an interrupt whenever it occurs.
* When the interrupt handler finishes executing the interrupt, then the CPU resumes execution of the task it stopped to respond to the interrupt.
* Software, Hardware, user, some bug in the programetc. can also cause an interrupt.
* The interrupt processing nature of the CPU leads to multitasking, i.e. the user can perform several different tasks simultaneously.
* If more than one interrupts are sent to the CPU, the interrupt handler helps in managing the interrupts that are waiting to be processed.
* As the interrupt handler is triggered by receiving an interrupt, it prioritizes the interrupts waiting to be processed by the CPU and arranges them into and queue to get serviced.

# Definition of polling

* As we saw in interrupts, input from an I/O device can arrive at any time with a CPU request to process it.
* Polling is a protocol that alerts the CPU that a device needs its attention. Unlike interrupts where the device tells the CPU that it needs CPU processing, in polling the CPU keeps asking the I/O device if it needs CPU processing.
* The CPU continuously polls each connected device to see if any device requires the CPU's attention.
* Each device has a ready-for-command bit that indicates the status of that device, i.e., whether it has any command to be executed by the CPU or not.
* If the command bit is set to 1 then it has some command to be executed elsewhere, if the bit is 0 then it has no commands. the CPU has a busy bit it indicates the state of the CPU whether it is busy or not.
* If the busy bit is set to 1, then it is busy executing some device command, otherwise it is 0.

## Algorithm for polling

1. When a device has some command to be executed by the CPU, it continuously checks the busy bit of the CPU until it appears (0).
2. Once the busy bit is set, the device sets the write bit in its command register and writes the byte to the output data register.
3. Now the device sets the (1) bit ready for the command.
4. When the CPU checks the device's command ready bit and finds that it is set (1), it sets (1) its busy bit.
5. The CPU then reads the device command register and executes the device command.
6. After the command is executed, the CPU clears (0) its ready bit, the device error bit, to indicate successful execution of the device command, and further clears (0) its busy bit to indicate that the CPU is free to execute another device command.
7. When the device interrupts, it notifies the CPU that it needs service, while when polling the CPU, it repeatedly checks to see if the device needs service.
8. Interrupt is a Hardware mechanism because the CPU has a wire, line request to interrupt what signal occurred. On the other side is Polling and Protocol which constantly checks the check bits to notify if the device has something to perform.
9. The interrupt handler handles the interrupts generated by the devices. On the other side, the polling processor services devices when they require it.
10. Interrupts are signaled by the interrupt request line. However, the Ready for Command bit indicates that the device needs servicing.
11. When an interrupt occurs, the CPU is only disturbed if it is interrupted by any device. On the other hand, when polling the CPU, the CPU loses a lot of CPU cycles by repeatedly checking the Ready on Command bits of each device.
12. An interrupt can occur at any time while the CPU is still polling the device at regular intervals.
13. Polling becomes ineffective when the CPU continues to poll the device and rarely finds any device ready for repair. On the other hand, interrupting becomes ineffective when a device repeatedly interrupts the CPU's processing.