## 

**DEPARTMENT OF COMPUTER SCIENCE & TECHNOLOGY**

**Subject Name:** Operating System  **Subject Code: CSH206B-T**

**Tutorial :3**

**Aim: To gain familiarity with introductory concepts of OS**

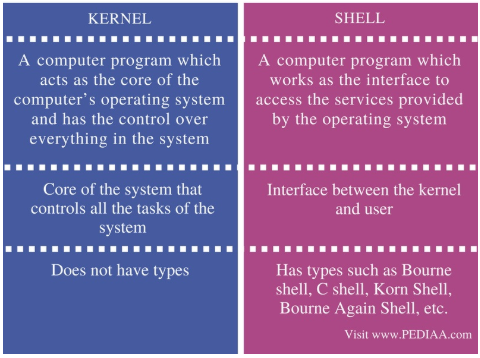
**Course Outcome : CO2**

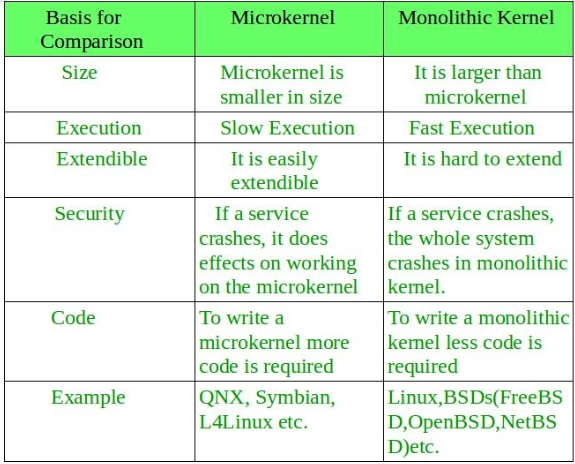
**Blooms Taxonomy: BT1,BT2, BT3, BT4**

1. Differentiate between
2. Kernel and Operating System

| **BASIS FOR COMPARISON** | **KERNEL** | **OPERATING SYSTEM** |
| --- | --- | --- |
| Basic | Kernel is an important part of the operating system. | Operating System is a system program. |
| Interface | Kernel is an interface between software and hardware of the computer. | Operating System is an interface between user and hardware of the computer. |
| Type | Monolithic kernels and Microkernels. | Single and Multiprogramming batch system, Distributed operating system, Realtime operating system. |
| Purpose | Kernel memory management, process management, task management, disk management. | In addition to the responsibilities of Kernel, Operating System is responsible for protection and security of the computer. |

1. Kernel and Shell

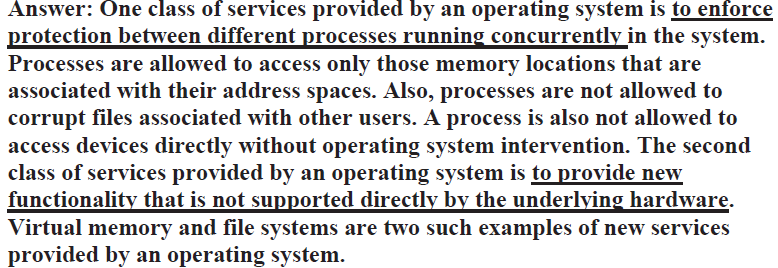


1. Monolithic Kernel and Micro Kernel
2. System Call and System Program

System calls provide an interface between the process and the operating system. System calls allow user-level processes to request some services from the operating system which process itself is not allowed to do. In handling the trap, the operating system will enter in the kernel mode, where it has access to privileged instructions, and can perform the desired service on the behalf of user-level process. It is because of the critical nature of operations that the operating system itself does them every time they are needed. For example, for I/O a process involves a system call telling the operating system to read or write particular area and this request is satisfied by the operating system.

System programs provide basic functioning to users so that they do not need to write their own environment for program development (editors, compilers) and program execution (shells). In some sense, they are bundles of useful system calls.

1. Define the Following: -
2. Trap: In **computing** and **operating systems**, a **trap**, also known as an exception or a fault, is typically a type of synchronous interrupt caused by an exceptional condition (e.g., breakpoint, division by zero, invalid memory access).
3. IVT: An "**interrupt vector table**" (IVT) is a data structure that associates a list of **interrupt** handlers with a list of **interrupt** requests in a **table** of **interrupt vectors**. Each entry of the **interrupt vector table**, called an **interrupt vector**, is the address of an **interrupt handler**.
4. ISR: Stands for "**Interrupt Service Routine**." An **ISR** (also called an **interrupt handler**) is a software process invoked by an **interrupt** request from a hardware device. It handles the request and sends it to the CPU, **interrupting** the active process.
5. The services and functions provided by an operating system can be divided into two main categories. Briefly describe the two categories and discuss how they differ.



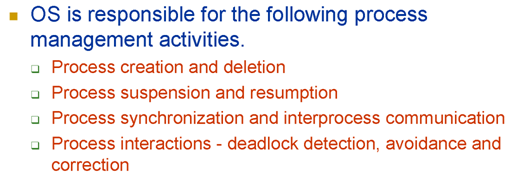
1. What is an Interrupt? Provide some examples where hardware and software interrupt is generated.

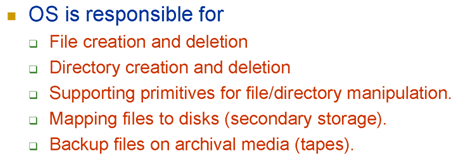
An interrupt is a signal from a device attached to a computer or from a program within the computer that requires the [operating system](https://whatis.techtarget.com/definition/operating-system-OS) to stop and figure out what to do next.

**Examples of s/w interrupts:**DOS Functions: Print a string message, Exit, Character Input, Printer Output.

**Examples of h/w interrupts:**pressing a [keyboard](https://en.wikipedia.org/wiki/Keyboard_(computing)) key or moving the [mouse](https://en.wikipedia.org/wiki/Mouse_(computing)) triggers hardware interrupts that cause the processor to read the keystroke or mouse position.

1. Explain the following: -
2. Dual Mode Operation
3. Memory Protection
4. I/O Protection
5. What are the five major activities of an operating system in regard to file management, Process Management?





1. Draw the Memory Structure for some processes and operating system with their legal memory addresses. The base and limit registers are loaded with the addresses 1050 and 1000, respectively. Suppose P1 and P2 reference the memory locations 2040 and 3052, respectively. Check if the processes will be allowed to execute.
2. Which of the following instructions should be privileged?
3. Switch from User Mode to Kernel Mode: privileged.
4. Updating Base and Limit Register: privileged.
5. Clear Memory Location: Privileged as clearing memory may prove fatal.
6. Read a clock: Unprivileged, every process should be able to read the clock.
7. Interrupts are disabled: Privileged so that a process cannot monopolize the cpu.
8. Executing a loop to enter user data: unprivileged.
9. Load a value in processor register: privileged.
10. Send a file to printer to print: unprivileged.