

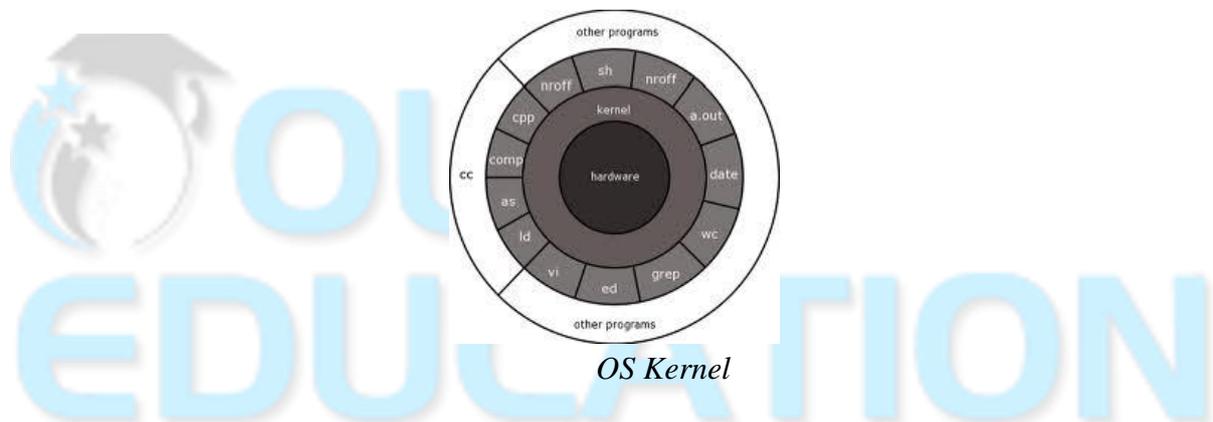
# Questions and Answers on Kernel and Shell in Operating Systems

## Q1. What is a kernel?

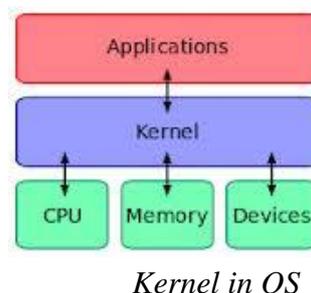
**Answer:** Kernel is actually the heart of operating systems. It acts as an interface between the applications and actual data processing done at hardware level (CPU, disk memory etc.).

Kernel lies in the centre of the operating system which manages the communication between the user level applications and the hardware installed on the system. In other words, we can say that it's a platform consisting the specific set of libraries and infrastructure for the new application made and thus help in interaction among them.

Here is a diagram that gives you a better view about what Kernel is in an Operating System



Let's take another picture of the Kernel in Operating system to clear this concept in our minds. In the second image below, the kernel lies in between the application and the hardwares like CPU, Memory and Devices and let them interact with the applications installed. Thus, it's behaving like a interaction level for the other two.



After clearing the introduction of the kernel in operating system. Let's find the types. There are following types of kernels developed-

- Monolithic Kernel
- Microlithic Kernel

- Hybrid or Modular Kernel
- Exo Kernel
- Nano Kernel

## Q2. What are the tasks performed by kernel?

**Answer:** There are following main tasks performed by kernel:-

- Memory Management
- Process Management
- Disk and File System Management
- Networking
- Security
- Graphical User Interface (GUI)
- Device Driver Management

Note: To know more about "Operating System Design/Kernel Architecture" read Operating System Design/Kernel Architecture

## Q3. Differentiate between Micro and Monolithic kernels.

**Answer:** The answer for this question comes from the names. There are following major differences-

**a.** Mono means everything in kernel space, whereas micro means minimal kernel space.

**b.** Monolithic Kernels perform Process management, Memory Management, File Input/Output. Here all OS services run along with the main kernel thread, thus also residing in the same memory area. Whereas,

In Micro-Kernel, very minimal support for the process management goes into the kernel. The micro-kernel approach consists of defining a simple abstraction over the hardware, with a set of primitives or system to implement minimal OS services such as address space management, thread management, and inter-process communication. All other services, including those normally provided by the kernel such as networking, are implemented in user-space programs referred to as *servers*.

**c.** The main disadvantage of monolithic kernels is the dependency between system components - a bug might crash the entire system - and the fact that large kernels also Main disadvantage with micro kernels is that large amount of system calls might slow down the system.

**d.** Examples of Monolithic Kernels are: Symbian os, AIX OS, Amoeba.  
Examples of Micro kernels are Mach, QNX, L4, Minix.

#### Q4. What are hybrid kernels?

**Answer:** Hybrid kernels can think of like extensions of micro kernels with some features of monolithic kernels. Basically, Hybrid kernels are a compromise between the monolithic and microkernel designs. This implies running some services (such as the network stack or the file system) in kernel space to reduce the performance overhead of a traditional microkernel, but still running kernel code (such as device drivers) as servers in user space. Hybrid kernels are used in most commercial operating systems such as Microsoft Windows NT, 2000, XP, Vista, and 7.

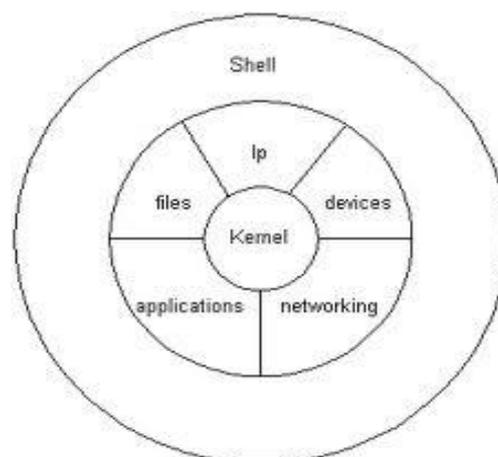
#### Q5. What are Exo-kernels?

**Answer:** These are the type of kernels that simply separate security from abstraction. It makes non-overridable parts of operating system multiplex the hardware securely. The goal is to avoid forcing any particular abstraction upon applications, instead allowing them to use or implement whatever abstractions are best suited to their task without having to layer them on top of other abstractions which may impose limits or unnecessary overhead.

#### Q6. What is a Shell in operating system?

**Answer:** A shell is simply software the gives a user interface to various operating system functions and services. So, in layman words "Your interface to operating system is called shell". Shells provide a way for you to communicate with the operating system. This communication is carried out either interactively (input from the keyboard is acted upon immediately) or as a shell script. A *shell script* is a sequence of shell and operating system commands that is stored in a file.

Here is a pic that gives you a brief idea about shell:



OS Shell

**Q7. What are the different types of shells available?**

**Answer:** Some most popular commercial shells are:-

- Korn shell
- Bourne Shell
- C shell
- POSIX shell

**Q8. List some salient features of shells in OS.**

**Answer:** Well, some relevant features are:-

**a. Wildcard substitution in file names (pattern-matching)**

Carries out commands on a group of files by specifying a pattern to match, rather than specifying an actual file name.

**b. Command aliasing**

Gives an alias name to a command or phrase. When the shell encounters an alias on the command line or in a shell script, it substitutes the text to which the alias refers.

**c. Command history**

Records the commands you enter in a history file. You can use this file to easily access, modify, and reissue any listed command.

**d. File name substitution**

Automatically produces a list of file names on a command line using pattern-matching characters.

**e. Input and output redirection**

Redirects input away from the keyboard and redirects output to a file or device other than the terminal. For example, input to a program can be provided from a file and redirected to the printer or to another file.

**f. Piping**

Links any number of commands together to form a complex program. The standard output of one program becomes the standard input of the next.

**g. Shell variable substitution**

Stores data in user-defined variables and predefined shell variables.

**Q9. What is Korn Shell?**

**Answer:** Korn shell is basically a UNIX shell that was developed by David Korn in Bell Labs. It has many features of C shell (csh) and Bourne Shell (bsh) and is one of the most efficient shells.

**Q10. What is C shell?**

**Answer:** C shell is a unix shell created by Bill Joy at the University of California at Berkeley as an alternative to UNIX's original shell, the Bourne shell. These two UNIX shells, along with the Korn shell, are the three most commonly used shells. The C shell program name is *csh*. The C shell was invented for programmers who prefer syntax similar to that of the C programming language.

