



1. **Kernel:** A **kernel** is that part of the operating system which interacts directly with the hardware and performs the most crucial tasks.
 - a. Heart of OS/Core component
 - b. Very first part of OS to load on start-up.
2. **User space:** Where application software runs, apps don't have privileged access to the underlying hardware. It interacts with kernel.
 - a. GUI
 - b. CLI terminal, PowerShell

A **shell**, also known as a command interpreter, is that part of the operating system that receives commands from the users and gets them executed.

Functions of Kernel:

1. **Process management:**
 - a. Scheduling processes and threads on the CPUs.
 - b. Creating & deleting both user and system process.
 - c. Suspending and resuming processes
 - d. Providing mechanisms for process synchronization or process communication.
2. **Memory management:**
 - a. Allocating and deallocating memory space as per need.
 - b. Keeping track of which part of memory are currently being used and by which process.
3. **File management:**
 - a. Creating and deleting files.
 - b. Creating and deleting directories to organize files.
 - c. Mapping files into secondary storage.
 - d. Backup support onto a stable storage media.
4. **I/O management:** to manage and control I/O operations and I/O devices
 - a. Buffering (data copy between two devices), caching and spooling.
 - i. Spooling
 1. Within differing speed two jobs.
 2. Eg. Print spooling and mail spooling.
 - ii. Buffering
 1. Within one job.
 2. Eg. Youtube video buffering
 - iii. Caching
 1. Memory caching, Web caching etc.

Types of Kernels:

1. Monolithic kernel
 - a. All functions are in kernel itself.
 - b. **Bulky in size.**
 - c. **Memory required to run is high.**
 - d. **Less reliable, one module crashes -> whole kernel is down.**
 - e. High performance as communication is fast. (Less user mode, kernel mode overheads)
 - f. Eg. Linux, Unix, MS-DOS.

2. Micro Kernel

- a. Only major functions are in kernel.
 - i. Memory mgmt.
 - ii. Process mgmt.
- b. File mgmt. and IO mgmt. are in User-space.
- c. smaller in size.
- d. More Reliable
- e. More stable
- f. Performance is slow.
- g. Overhead switching b/w user mode and kernel mode.
- h. Eg. L4 Linux, Symbian OS, MINIX etc.



3. Hybrid Kernel:

- a. Advantages of both worlds. (File mgmt. in User space and rest in Kernel space.)
- b. Combined approach.
- c. Speed and design of mono.
- d. Modularity and stability of micro.
- e. Eg. MacOS, Windows NT/7/10
- f. IPC also happens but lesser overheads

4. Nano/Exo kernels...

Q. How will communication happen between user mode and kernel mode?

Ans. Inter process communication (**IPC**).

1. Two processes executing independently, having independent memory space (Memory protection), But some may need to communicate to work.
2. Done by shared memory and message passing.