## **🔐 User Mode vs Supervisor Mode (Kernel Mode)**

### **🔹 1. What is it?**

Modern processors support at least two operating modes to **protect system resources** and **ensure safe multitasking**:

* **User Mode**: Limited access — for applications like browsers, games, etc.
* **Supervisor Mode (Kernel Mode)**: Full access — for OS-level tasks like memory management, I/O control, etc.

### **🔹 2. Key Differences**

| **Feature** | **User Mode** | **Supervisor Mode (Kernel)** |
| --- | --- | --- |
| Access to hardware | ❌ No direct access | ✅ Full access |
| CPU instruction set | Limited | Full (including privileged instructions) |
| Memory access | Restricted | Can access all memory |
| Who operates here? | User applications | Operating System kernel |
| Safety | Very safe (can't crash OS) | Risky if misused |

### **🔹 3. Why is it important?**

* Prevents **accidental or malicious code** from damaging the system.
* Supports **multitasking** and **security isolation**.
* Enables **system calls**: e.g., when a user app requests a file, it switches to Supervisor Mode.

### **🔹 4. Mode Switching**

When an application needs OS services, it uses a **system call**:

plaintext

CopyEdit

+-----------------------+

| User Application | ← User Mode

+-----------------------+

|

| (System Call)

v

+-----------------------+

| Operating System Code | ← Switches to Supervisor Mode

+-----------------------+

Example (in C):

// User mode code

open("data.txt", O\_RDONLY); // Triggers a syscall

### **🔹 5. Real-world Analogy**

Think of a **User Mode** as a guest at a hotel — they can use facilities but not enter the staff room. The **Supervisor Mode** is like the hotel manager — with master keys and full control.