# Linux Kernel Architecture

A Deep Dive into the Core of the OS

## 1. User Space (Applications & glibc):

- Contains user apps and the GNU C Library (glibc).
- glibc acts as the primary wrapper for system calls.

## 2. System Call Interface (SCI):

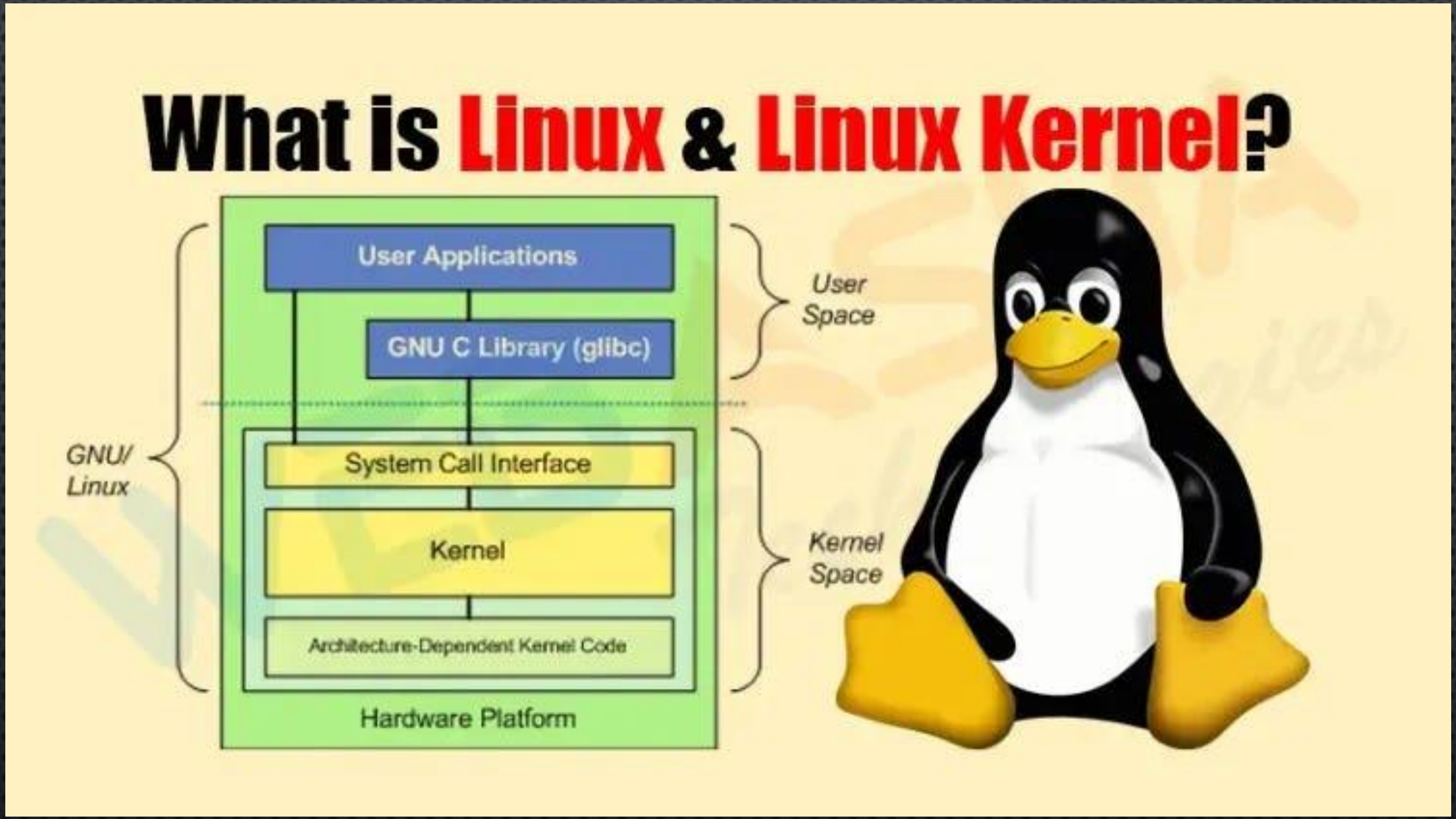
- The gatekeeper. It transitions requests from User Space to Kernel Space.
- Ensures security and provides a stable API for developers.

## 3. Kernel Space (The Engine):

- The 'Main Kernel' manages scheduling, memory, and filesystems.
- 'Arch-Dependent Code' optimizes the kernel for specific CPUs (x86, ARM).

## 4. Hardware Platform:

- The physical layer (CPU, RAM, Disk, NIC).
- Controlled by the kernel via Device Drivers.


What is Linux & Linux Kernel?
User Applications
GNU C Library (glibc)
User Space
GNU/ Linux
System Call Interface
Kernel
Kernel Space
Architecture-Dependent Kernel Code
Hardware Platform

# Architectural Breakdown: User vs. Kernel Space

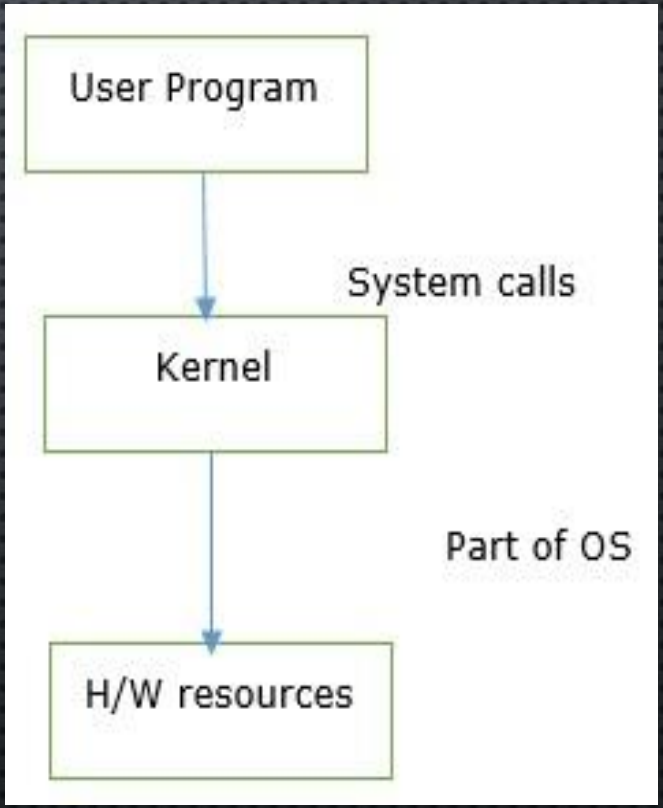
SYSTEM ARCHITECTURE DETAILS:

• TRAP MECHANISM: When glibc executes a System Call, it triggers a 'Software Interrupt' (Trap), forcing the CPU to switch from Ring 3 (User) to Ring 0 (Kernel) mode.

• MONOLITHIC DESIGN: Unlike Microkernels, Linux runs the Scheduler, VFS, and Network Stack within a single address space to eliminate IPC overhead and boost speed.

• ARCH-DEPENDENT LAYER: This is the 'Hardware Abstraction Layer' (HAL). It contains assembly code specific to CPUs (like x86_64 or ARM), allowing the core kernel to remain portable.

• GLIBC WRAPPER: Applications don't talk to the kernel; they talk to glibc. This ensures that if the kernel API changes, the application code doesn't have to.



# Process Scheduler

- Manages CPU time for all processes.
- Uses the Completely Fair Scheduler (CFS).
- Supports Preemption and Multi-core scalability.

# Virtual File System (VFS)

- An abstraction layer for storage.
- Supports multiple formats (ext4, XFS, Btrfs).
- Standardizes 'read/write' operations for applications.

# Network Stack

- Implements TCP/IP and other protocols.
- Uses a socket-based API for communication.
- Interfaces directly with Network Interface Cards (NICs).