1.12 Clustered Systems vs Multiprocessor Systems

A multiprocessor system consists of multiple CPUs sharing the same memory and operating under a single operating system instance.
A clustered system consists of multiple independent machines connected via a network, each with its own memory and OS.

To provide high availability, clustered systems require:
• Shared storage or replicated data
• Communication mechanisms between nodes
• Failure detection
• Cluster management software

1.13 Disk Access in a Two-Node Database Cluster

Method 1: Shared Disk Access
Both nodes access the same disk storage.
Advantages: Simple data consistency, easy failover
Disadvantages: Disk is a single point of failure, performance bottleneck

Method 2: Replicated Data
Each node has its own copy of the database.
Advantages: High availability, better performance
Disadvantages: Complex synchronization, consistency issues

1.14 Interrupts and Traps

Interrupts are signals sent to the CPU by hardware devices to request attention.
A trap is a software-generated interrupt caused by an error or system call.

Difference:
Interrupt $\rightarrow$ caused by hardware
Trap $\rightarrow$ caused by software

Yes, traps can be generated intentionally by user programs to request operating system services using system calls.

1.15 Linux HZ and jiffies

HZ defines the number of timer interrupts per second.
jiffies counts the total number of timer interrupts since boot.

System uptime in seconds can be calculated as:
uptime = jiffies / HZ

1.16 Direct Memory Access (DMA)

a. The CPU initializes the DMA controller by providing the device, memory address, and transfer size.

b. The DMA controller sends an interrupt to the CPU upon completion.

c. Yes, interference can occur due to:

• Memory bus contention

• Cache coherence issues

1.17 Secure OS Without Privileged Mode

Arguments For:

• Software-based protection techniques

• Language-based security

• Virtual machines

Arguments Against:

• No hardware enforcement

• User programs can bypass OS controls

• Difficult to ensure isolation

Conclusion: Security is possible but weaker without hardware support.

1.18 Multi-Level Cache Design in SMP Systems

Local caches reduce access latency for each core.

Shared caches improve data sharing and reduce duplication.

This design balances performance, scalability, and cache coherence efficiency.