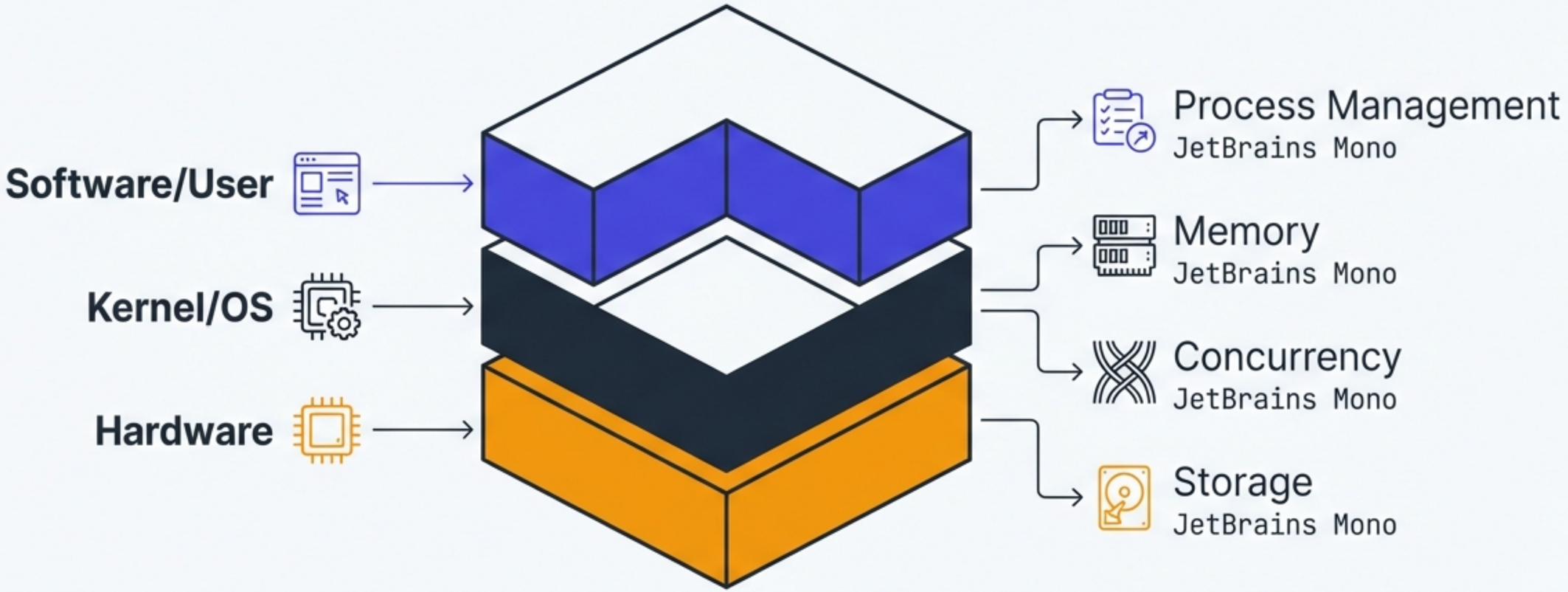
Operating Systems Masterclass

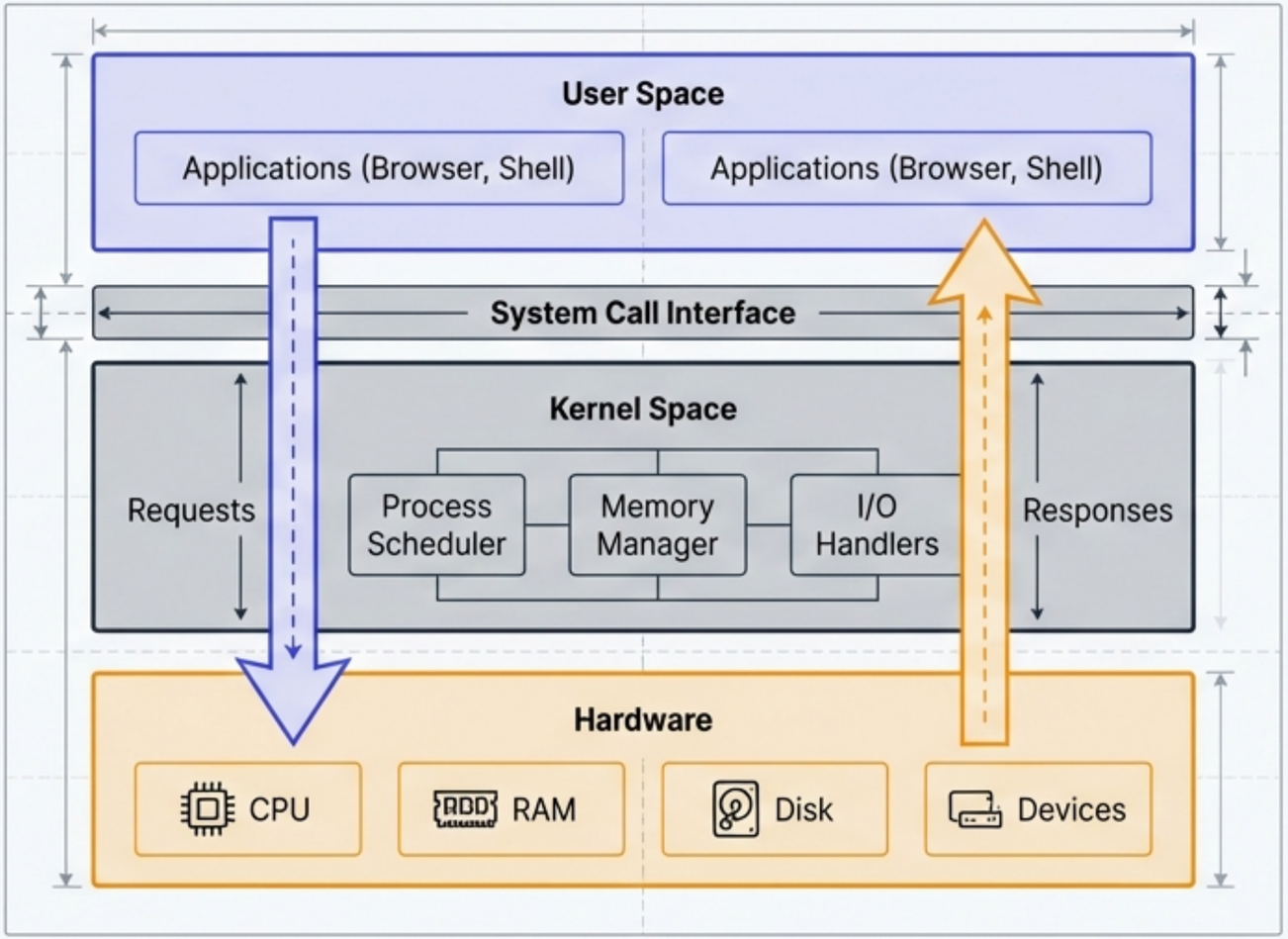
Architecture, Internals & System Design



COMPREHENSIVE REVIEW: CORE • INTERMEDIATE • ADVANCED

The Role of the Conductor: OS & Kernel

Mediating between user intent and hardware reality



OS vs. Kernel

The **OS** includes the full suite of system software (GUI, Shell, Tools).
The **Kernel** is the strictly core component bridging software to hardware.

Kernel Types

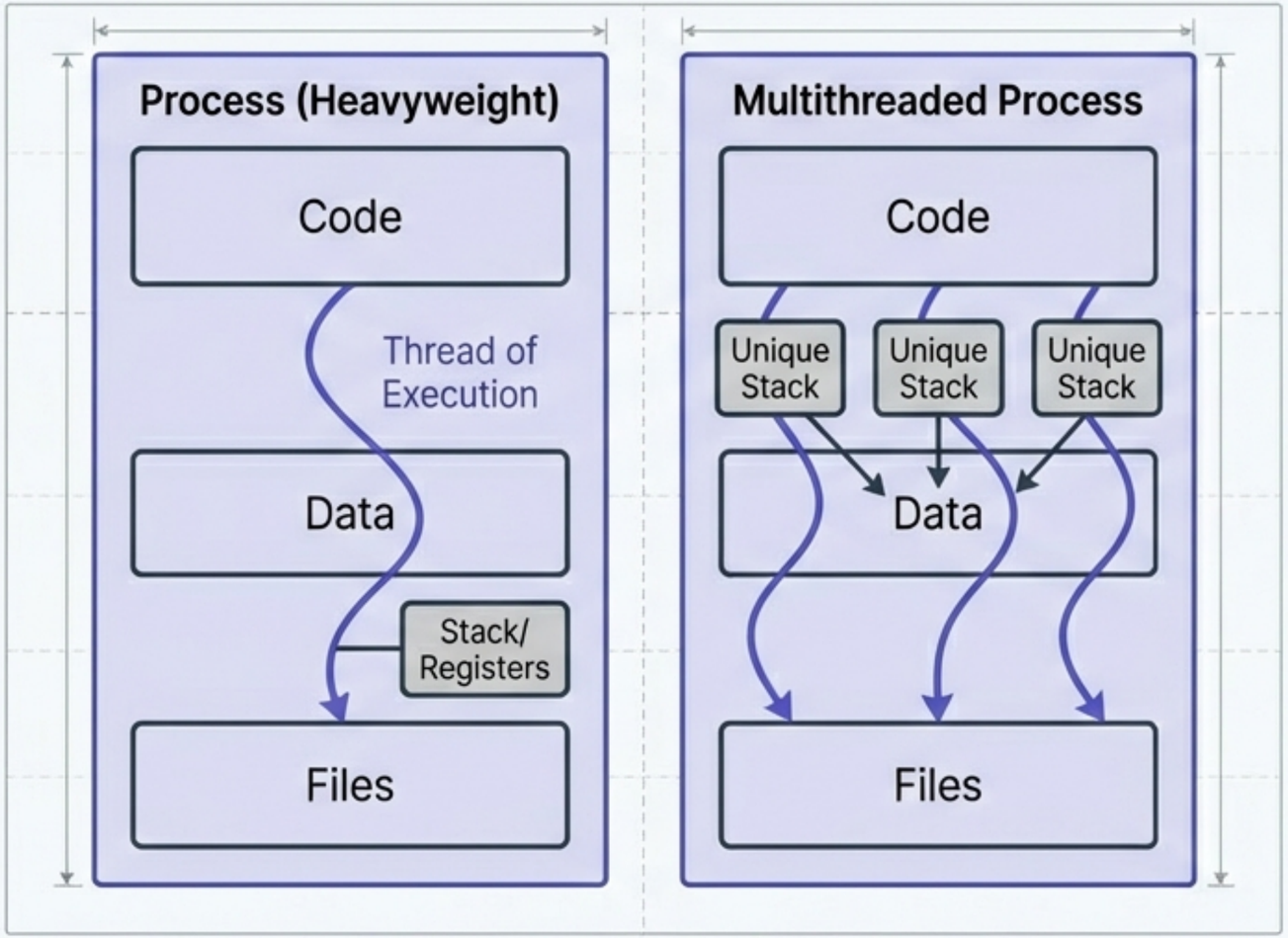
Monolithic: User and Kernel services in same address space (Faster, Larger).
Microkernel: Minimal services in kernel mode (Modular, Slower).

The Bootstrap

The initial program loaded by the POST (Power-On Self-Test) to initialize the OS components.

Units of Execution: Process vs. Thread

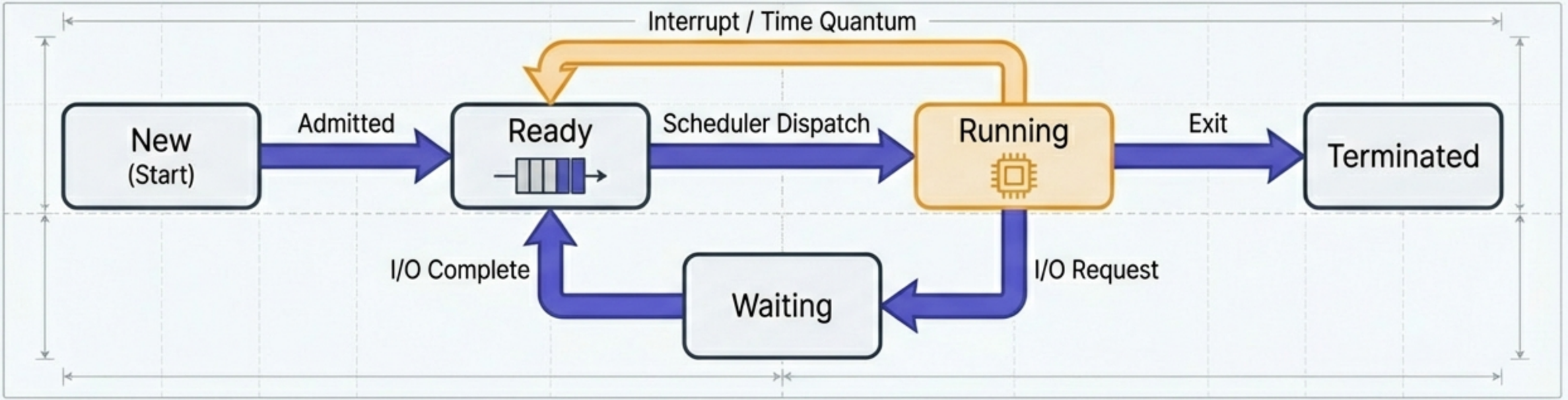
Heavyweight isolation versus lightweight concurrency.



Feature	Process	Thread
Memory	Isolated Address Space	Shared Memory (Heap/Data)
Creation Cost	High (OS allocates resources)	Low (Lightweight)
Communication	Slow (IPC mechanisms)	Fast (Direct shared access)
Context Switch	Heavy (Save full state)	Light (Save registers/stack)
Failure Scope	Isolated (Others run)	Fatal (Can crash process)

The Process Lifecycle & State Management

Understanding process transitions, data structures, and system anomalies.



PCB (Process Control Block)

The data structure tracking execution status. Contains `Program Counter`, `Registers`, and `Priority`. Stored in the Process Table.

Context Switching

The overhead of saving the state of the old process (to PCB) and loading the new one. No useful work is done during this switch.

Lifecycle Anomalies

Zombie: Finished execution but entry remains in process table (Parent hasn't read exit code).
Orphan: Parent terminated while child is still running.

Scheduling the CPU: Algorithms

Deciding which process runs next to maximize throughput. in Inter.

Scheduling Strategies

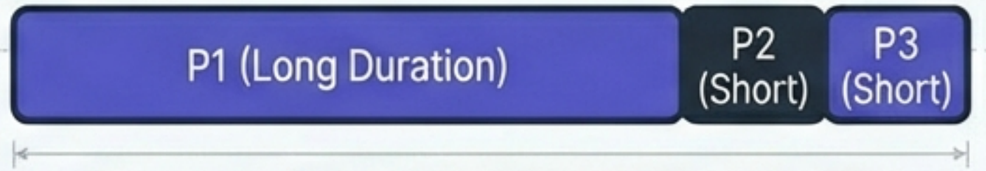
"Preemptive": OS allocates CPU for limited time (**Quantum**). Higher priority can interrupt. (Flexible, High Overhead).

"Non-Preemptive": Process holds CPU until termination or I/O wait. (Simple, Rigid).

"Dispatcher": The module that gives control of the CPU to the selected process.

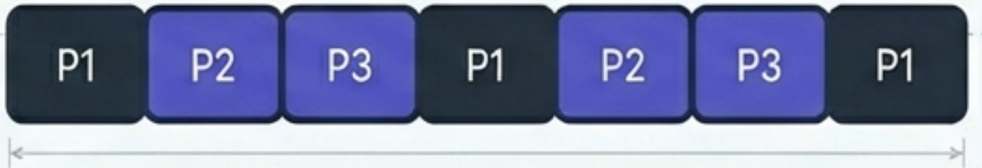
Algorithm Comparison

FCFS (First-Come, First-Served)



Convoy Effect: Short jobs wait for long job.

Round Robin (RR)

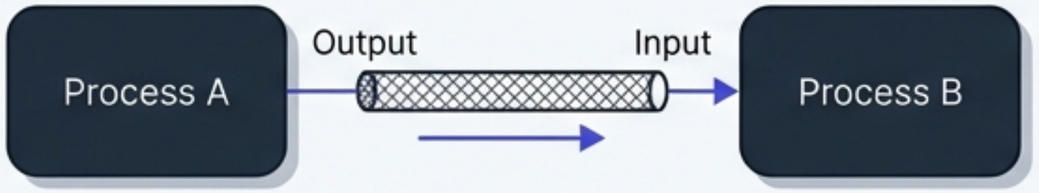


Time Quantum (Time Slicing). Fair, but higher context switching overhead.

Inter-Process Communication (IPC)

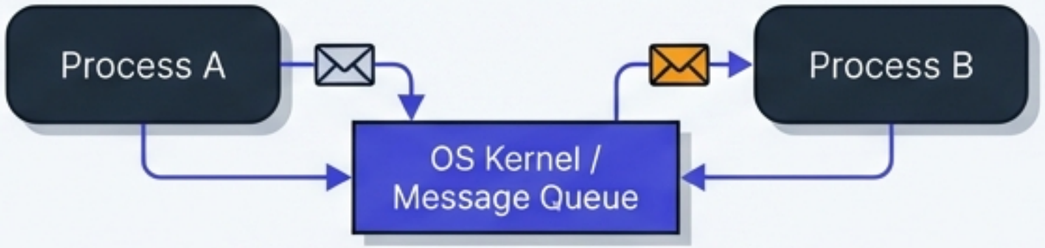
Mechanisms for isolated processes to exchange data.

Pipes



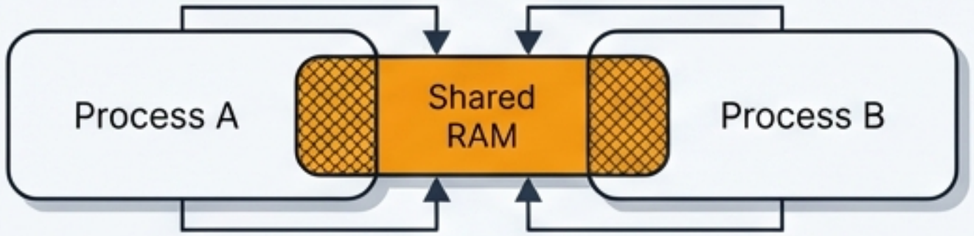
Unidirectional data flow.
Output of one becomes input of another.

Message Queue



Kernel-managed list of messages
Asynchronous communication.

Shared Memory



Fastest method. Direct memory access.
Requires synchronization to prevent conflicts.

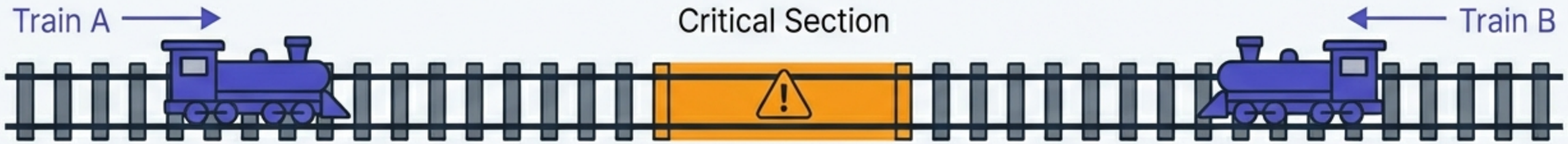
Sockets

Network-based communication.
Works across different machines/OS.

The Synchronization Challenge

Race Conditions and the Critical Section Problem.

Race Condition Visualization



When two threads access a shared resource (variable/file) simultaneously, the outcome depends on execution order.

Mutex (Lock)

A key. Only the holder can enter.
If locked, others wait.
Prevents simultaneous access.

Semaphore

A signaling integer. Non-negative. `Wait()` decriments, `Signal()` increments.
Can allow N simultaneous users (Counting Semaphore).

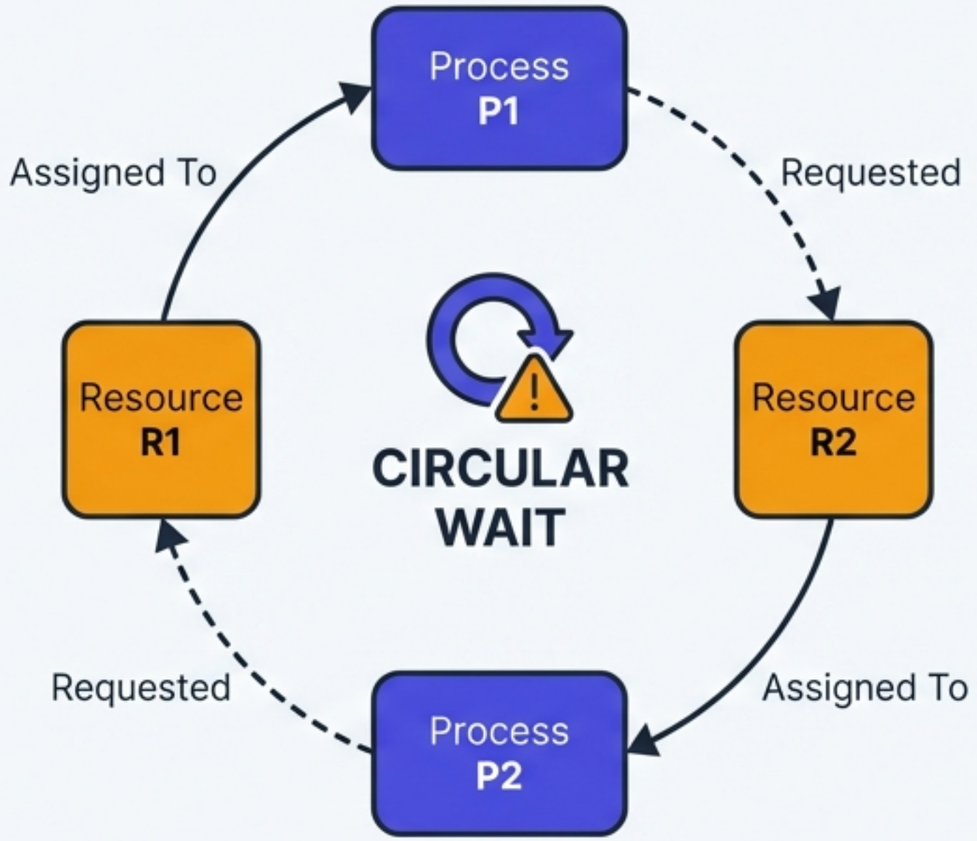
Peterson's Solution

Software-based algorithmic approach using two variables (`flag` and `turn`) to enforce mutual exclusion.

The Nightmare Scenario: Deadlock

A state where processes wait indefinitely for each other.

Resource Allocation Graph



The 4 Necessary Conditions (Coffman Conditions)

If one is broken, deadlock is impossible.

Mutual Exclusion

Resources cannot be shared (e.g., a printer).

Hold and Wait

Holding a resource while waiting for another.

No Preemption

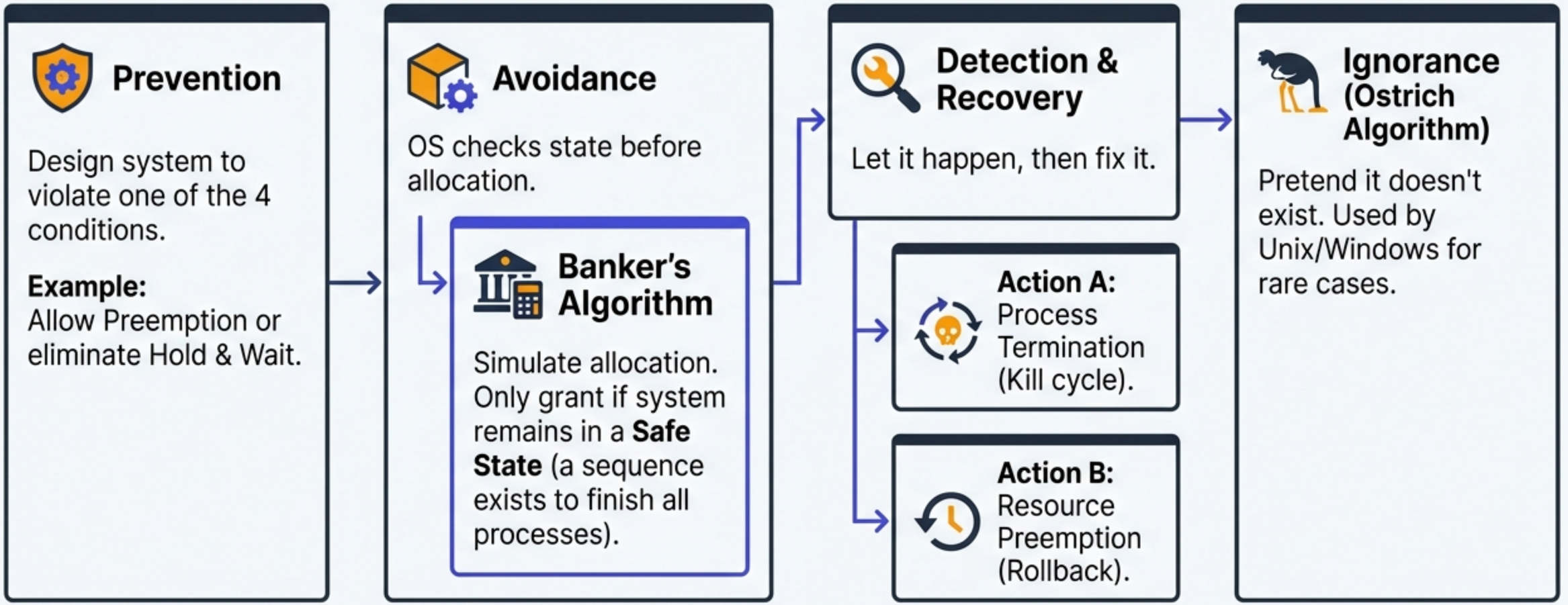
Resources cannot be forcibly taken away.

Circular Wait

A closed chain of processes waiting for each other.

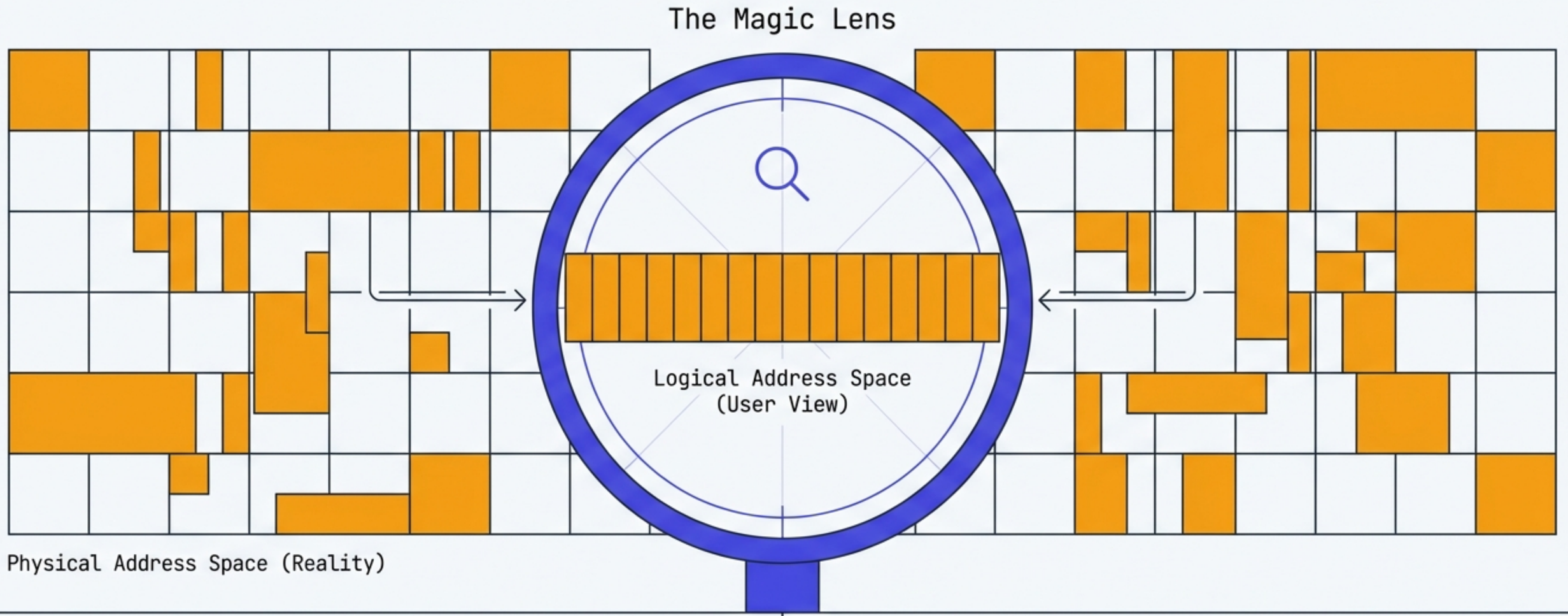
Handling Deadlocks

Strategies for Prevention, Avoidance, and Recovery.



Memory Abstraction: Virtual vs. Physical

The illusion of infinite, contiguous memory.

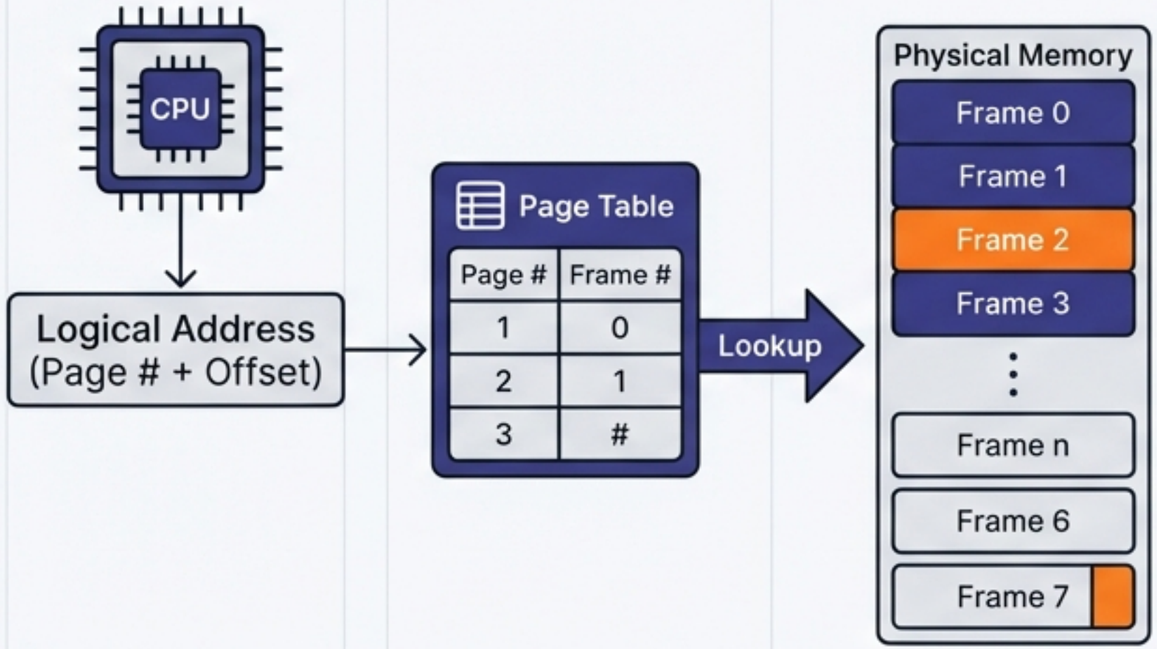


Memory Management Unit (MMU): Hardware device that translates Logical Addresses (CPU generated) to Physical Addresses (RAM location) at runtime.

Benefit: Allows execution of programs larger than physical memory and provides process isolation.

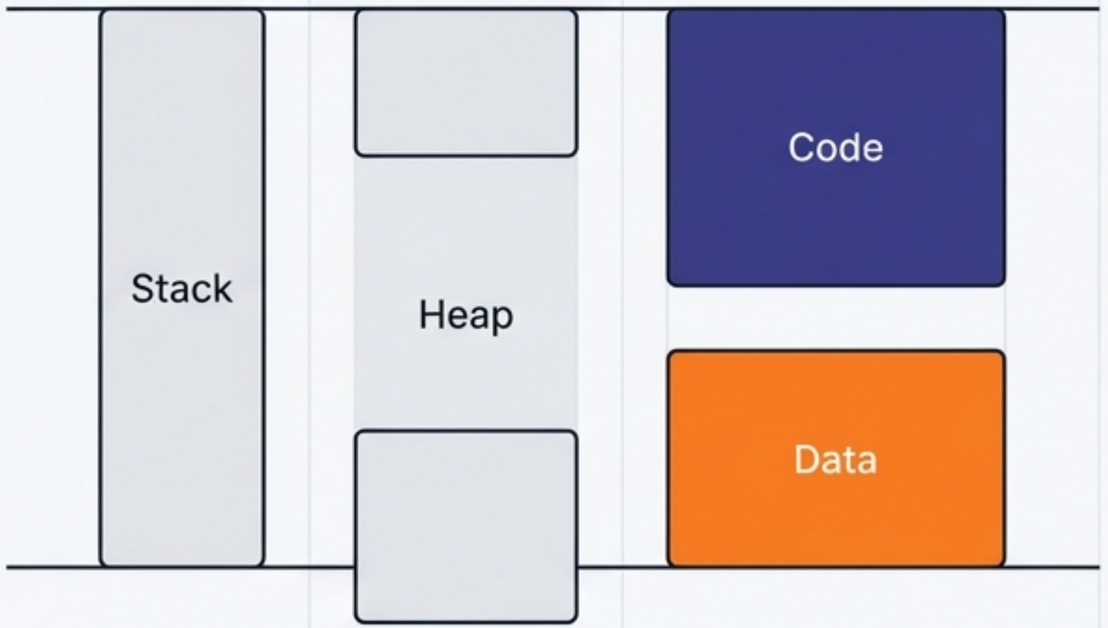
Organizing Memory: Paging & Segmentation

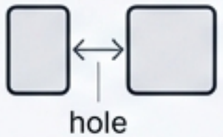
Paging (Fixed Size)



Invisible to user. Solves external fragmentation but causes internal fragmentation (wasted space inside last page).

Segmentation (Variable Size)





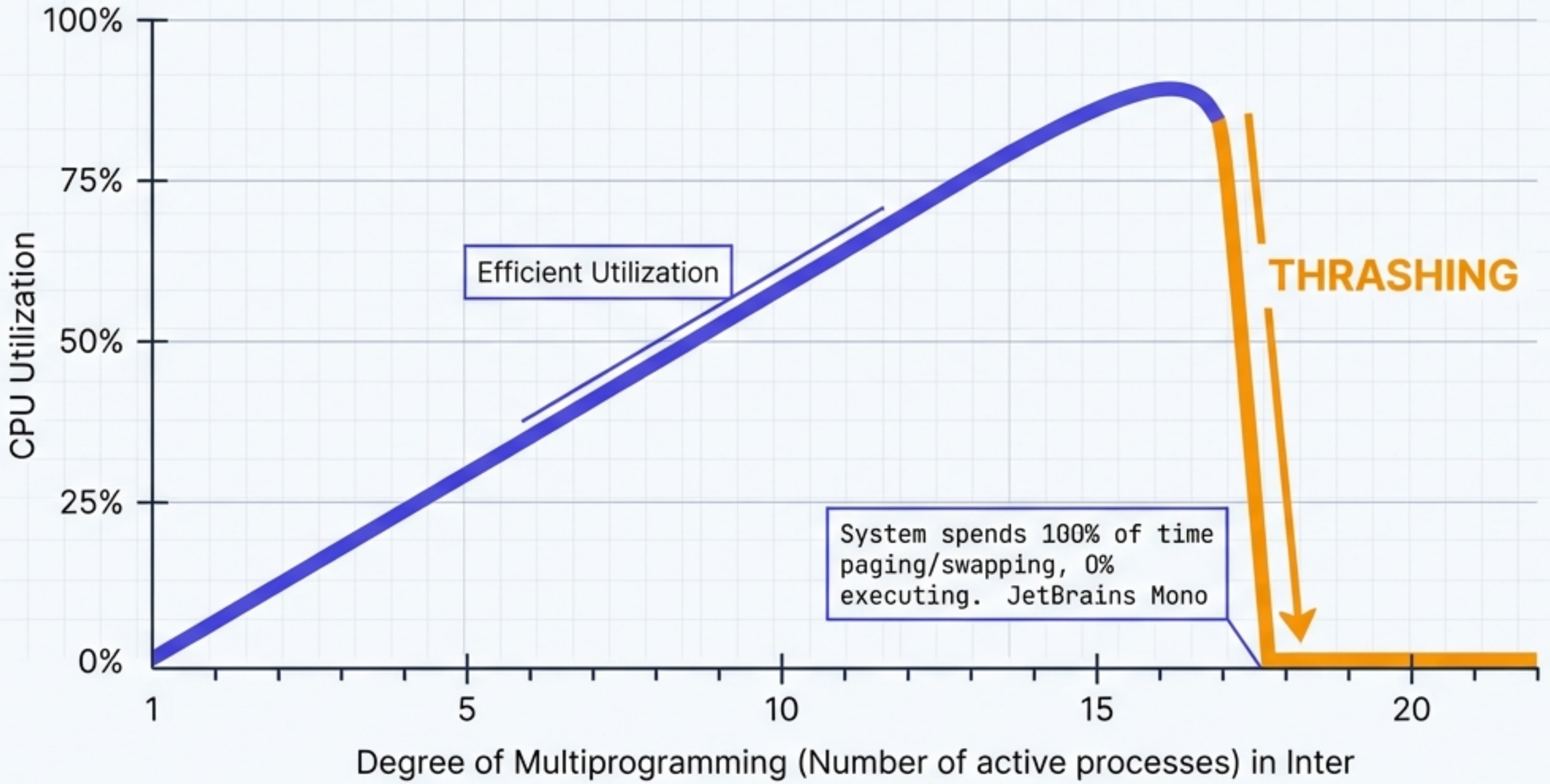
Visible to user logic. Segments vary in size. Causes external fragmentation (holes between segments too small to use).

Demand Paging: Loading pages only when a 'Page Fault' occurs.

Memory Performance & Pitfalls

The crucial balance between process execution and paging overhead.

The Thrashing Curve



- **Locality of Reference**: Programs tend to access specific clusters of memory repeatedly. Caching relies on this.
- **Belady's Anomaly:** A paradox in FIFO paging where adding *more* memory frames actually causes *more* page faults.

Storage & I/O Architecture

RAID Levels (Redundant Array of Independent Disks)

RAID 0 (Striping)

(Fast, No Redundancy)

RAID 1 (Mirroring)

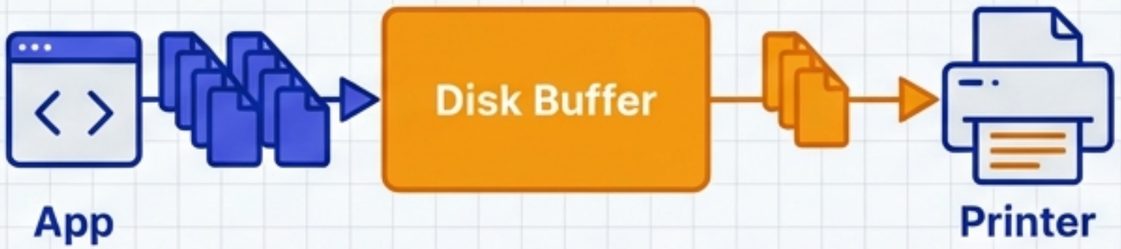
(Redundant, Expensive)

RAID 5 (Striping + Parity)

(Balanced)

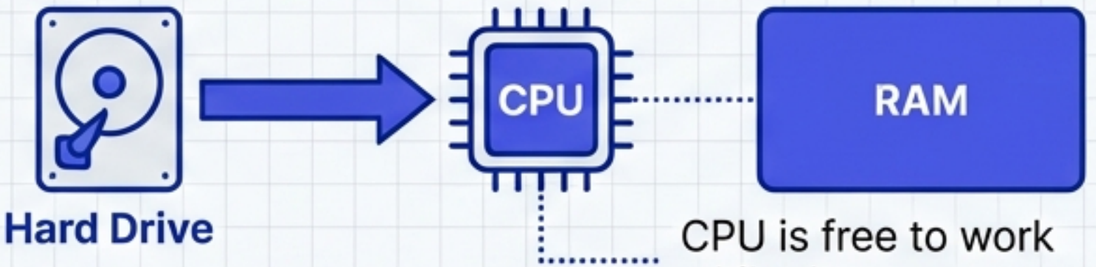
I/O Techniques

Spooling



Overlap computation with I/O.

DMA (Direct Memory Access)



CPU is free to work while data moves.

System Design & Terminology Cheat Sheet

Rapid-fire definitions for interview readiness.

Interrupt vs. Trap

Interrupt: Hardware signal (async) needing attention.
Trap: Software interrupt (sync) caused by error or system call.

Cycle Stealing

I/O controllers accessing RAM via the bus when the CPU isn't using it, subtly 'stealing' memory cycles.

Compaction

Shuffling memory contents to place all free space together in one large block. Solves external fragmentation.

Overlay

Loading only the required instructions of a program into memory, overwriting previously used parts. (Old technique).

Bit-Vector

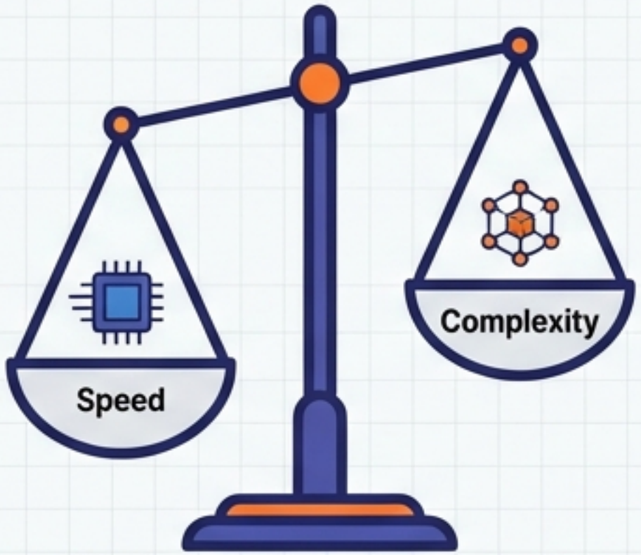
A string of bits (0010110) used to track free disk blocks. 1 = Free, 0 = Allocated.

Starvation vs. Aging

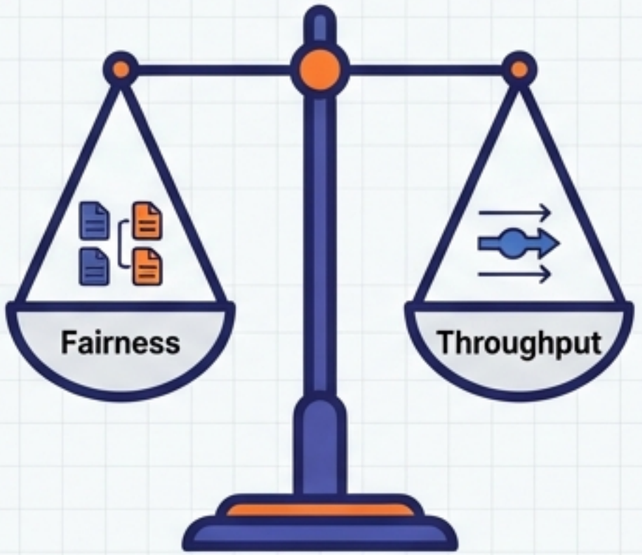
Starvation: Low priority process never runs.
Aging: Gradually increasing priority over time to ensure execution.

The Balancing Act: Key Takeaways

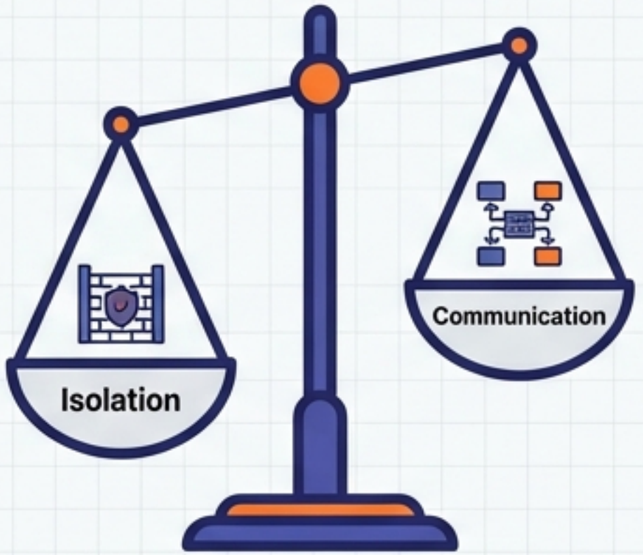
Operating System design is the management of trade-offs.



`Monolithic kernels` offer raw speed but difficult maintenance. **`Microkernels`** offer stability but suffer **IPC overhead**.



`Preemptive scheduling` is responsive but costly (**context switching**). **Non-preemptive** is efficient but allows **`starvation`**.



Processes provide safety through isolation. **Threads** provide speed through shared memory but introduce risk (**Race Conditions**).

A robust OS manages the chaos of hardware concurrency to provide the illusion of infinite, dedicated resources to the user.