<strong>Key Concepts and Entities:</strong>

- <strong>Operating System (OS):</strong> Manages hardware, applications, and user interaction.
- <strong>Computer System:</strong> Includes hardware, OS, applications, and users.

<strong>Important Examples and Figures:</strong>

- Figure 1.1: Components of a computer system.
- Figure 1.2: Example of a typical PC system.

<strong>Relationships between Ideas:</strong>

- OS allocates resources (e.g., CPU time, memory) to programs and users.
- OS manages I/O devices and user programs to prevent conflicts.
- Interrupts signal the CPU of events requiring attention.
- Storage includes memory and storage devices.
- I/O structure consists of device controllers, drivers, and buses.

<strong>Summary:</strong>

<strong>Operating System Functions:</strong>

- Provides user interface for hardware interaction.
- Optimizes system performance by allocating resources.
- Controls I/O devices and programs to ensure error-free operation.

<strong>Computer System Organization:</strong>

- Multiple CPUs and device controllers manage system resources.
- Interrupts enable timely handling of hardware events.
- Storage structure includes memory and storage devices.
- <p>Manages hardware, applications, and user interactions.</p>
- Allocates resources to programs and users.
- Controls I/O devices to prevent conflicts.

<strong>Computer System Organization</strong>:

- Includes hardware, OS, applications, and users.
- Interrupts signal the CPU of events requiring attention.
- Storage structure consists of memory (RAM, EEPROM) and storage devices (HDDs, NVM).
- I/O structure involves device controllers, drivers, and buses.

<strong>Additional Details</strong>:

- Interrupts: Notify the CPU of events and are mapped to service routines.
- Main memory (RAM) stores active programs and data.
- Secondary storage provides permanent data storage.
- Device controllers connect I/O devices to the computer.
- <p><strong>Key Concepts:</strong> Interrupts notify the CPU of events that require attention.</p>
- <strong>Important Figures:</strong> Timeline and I/O cycle illustrate interrupt handling.

- <strong>Relationship:</strong> Interrupt vector table maps interrupts to service routines.

<strong>1.2.2 Storage Structure</strong>

- Main memory (RAM) stores active programs and data.
- EEPROM stores nonvolatile data.
- Secondary storage (HDDs, NVM) provides permanent data storage.
- <strong>Notation:</strong> Bits (units of storage), bytes (8 bits), KB (1,024 bytes), MB (1,0242 byte

<strong>1.2.3 I/O Structure</strong>

- Device controllers connect I/O devices to the computer.
- Device drivers provide a standardized interface to devices.
- Manages hardware, applications, and user interactions.
- Allocates resources (CPU time, memory) to programs and users.
- Controls I/O devices to prevent conflicts.

- <strong>Key Concepts:</strong> Bit, byte, word, storage units (KB, MB, GB, TB, PB)
- <strong>Examples:</strong> 1 byte = 8 bits, 1 KB = 1,024 bytes

<strong>Computer-System Organization</strong>

- <strong>Key Concepts:</strong> Main memory, secondary storage, tertiary storage, cache memory
- <strong>Entities:</strong> HDDs (mechanical), NVM devices (electrical)

<strong>I/O Structure</strong>

- <strong>Key Concepts:</strong> DMA, interrupt
- <strong>Relationship:</strong> DMA bypasses CPU for data transfers, interrupts notify CPU of eve

<strong>Computer-System Architecture</strong>

- <strong>Key Concepts:</strong> Single-processor, multiprocessor, SMP, multicore
- <strong>Relationships:</strong> SMP distributes tasks equally, multicore integrates computing core

<strong>Definitions of Computer System Components</strong>

- <strong>Entities:</strong> CPU, processor, core, multicore, multiprocessor

<strong>Key Concepts:</strong>

- Storage: Bits, bytes, storage units (KB, MB, GB, TB, PB)
- System Organization: Main memory, secondary storage, cache memory
- I/O Structure: DMA, interrupts
- Architecture: Single-processor, multiprocessor, SMP, multicore

<strong>Entities and Relationships:</strong>

- CPU: Executes instructions
- Processor: Contains multiple CPUs (cores)
- Multicore: Multiple cores per CPU
- Multiprocessor: Equal CPUs sharing tasks
- Clustered Systems: Multiple systems connected for high availability

- Bootstrap: Initializes system and loads OS
- Kernel: Provides services to user programs
- Process: Program execution

<strong>Key Concepts:</strong>

- CPU: Executes instructions
- Processor: Chip with multiple CPUs (cores)
- Multicore: Multiple cores per CPU

<strong>1.3.2 Multiprocessor Systems</strong>

- Symmetric multiprocessing (SMP): Equal CPUs sharing tasks
- Multicore systems: Multiple cores on a single chip
- NUMA: Local memory for each CPU, shared interconnect
- Blade servers: Multiple processor boards in a single chassis

<strong>1.3.3 Clustered Systems</strong>

- Multiple individual systems (nodes)
- Loosely coupled
- High availability: Continuous service despite failures
- Graceful degradation: Reliability decreases with hardware failures
- Fault tolerance: Can handle single-component failures

<strong>1.4 Operating-System Operations</strong>

<strong>Key Concepts:</strong>

- Bootstrap program: Initializes system and loads OS
- Kernel: Provides services to user programs
- Process: Program execution
- <p>Manages hardware, applications, and user interactions.</p>
- Allocates resources (CPU time, memory) to programs and users.
- Controls I/O devices to prevent conflicts.

<strong>Computer System Organization:</strong>

- Includes hardware, OS, applications, and users.
- Interrupts signal the CPU of events requiring attention.
- Storage structure consists of memory (RAM, EEPROM) and storage devices (HDDs, NVM).
- I/O structure facilitates communication between devices and the OS.

<strong>Additional Details:</strong>

- <strong>Interrupts:</strong> Notify the CPU of events and are mapped to service routines.

- Main memory (RAM) stores active programs and data.
- Secondary storage provides permanent data storage.

- Device controllers connect I/O devices to the computer.
- Device drivers provide a standardized interface to devices.

<strong>Operating System Operations:</strong>

- Initializes system and loads OS (Bootstrap program).
- Provides services to user programs (Kernel).
- Executes instructions (CPU).
- Executes operating system code and protects it from user errors (Kernel Mode).
- Interface for user programs to access operating system services (System Calls).

<strong>Resource Management:</strong>

- <p>Enables shared data access for multiple hosts with software like Oracle Real Application Cluster
- Ensures conflict prevention through access control and locking (DLM).

<strong>Hadoop</strong>

- Framework for large data processing across distributed systems (single to thousands of nodes).
- Comprises a distributed file system, resource manager (YARN), and processing system (MapReduc

<strong>Operating System Operations</strong>

<strong>Dual-Mode and Multimode Operation:</strong>

- Shared data access for multiple hosts using Oracle Real Application Cluster.
- Prevents conflicts through access control and locking (DLM).

<strong>Hadoop</strong>

- Framework for processing large data in distributed systems.
- Components: distributed file system, resource manager (YARN), processing system (MapReduce).

<strong>Operating System Operations</strong>

<strong>Multiprogramming and Multitasking:</strong>

- Multiple programs run concurrently, maximizing CPU utilization.
- Multitasking allows switching between processes, improving responsiveness.

<strong>Virtual Memory:</strong>

- Extends memory capacity beyond physical limits.
- Maps logical memory to physical memory, optimizing memory usage.

<strong>File System:</strong>

- Manages data on secondary storage devices.

<strong>Dual-Mode Operation:</strong>

- <strong>User Mode:</strong> Executes user programs.
- <strong>Kernel Mode:</strong> Executes operating system code, providing protection.

<strong>System Calls:</strong>

- Interface for user programs to access operating system services.

<strong>Resource Management</strong>

- <strong>Process Management:</strong> Manages running programs (processes).
- <strong>Memory Management:</strong> Allocates and deallocates memory for processes.
- <strong>File-System Management:</strong> Organizes and controls file access.
- <p><strong>Key Concepts:</strong> Kernel mode (0) and user mode (1)</p>
- <strong>Examples:</strong> I/O control instructions require kernel mode.
- <strong>Relationships:</strong> Mode bit determines protection levels.

<strong>1.4.3 Timer</strong>

- <strong>Key Concept:</strong> Timer interrupts prevent CPU monopolization.
- <strong>Example:</strong> Timer triggers interrupts to regain control from user programs.
- <strong>Relationships:</strong> Ensures fair CPU allocation and prevents infinite loops.

<strong>1.5 Resource Management</strong>

<strong>1.5.1 Process Management</strong>

- <strong>Key Concept:</strong> Process as a running program instance.
- <strong>Examples:</strong> Compilers, word processors, social media apps.
- <strong>Relationships:</strong> Processes consume resources (CPU, memory, I/O).

<strong>1.5.2 Memory Management</strong>

- <strong>Key Concept:</strong> Main memory stores data and instructions.
- <strong>Examples:</strong> Algorithms allocate and deallocate memory space.
- <strong>Relationships:</strong> Optimizes CPU utilization and system response.

<strong>1.5.3 File-System Management</strong>

- <strong>Key Concept:</strong> Files as organized information storage units.
- <strong>Examples:</strong> Programs, data files (numeric, text, binary).
- <strong>Relationships:</strong> Organizes and controls file access.

<strong>1.5.4 Mass-Storage Management</strong>

- <strong>Key Concept:</strong> Secondary storage (HDDs, NVM) stores programs and data.
- <strong>Examples:</strong> OS manages storage devices for optimal access and performance.
- Allows multiple hosts to share data (e.g., Oracle Real Application Cluster)
- Prevents conflicts through access control and locking (DLM)

<strong>Operating System Operations:</strong>

<strong>Key Concepts:</strong>

- Memory management: Memory allocation for programs and data.
- File-system management: File organization and access.
- Mass-storage management: Secondary and tertiary storage handling.
- Cache management: Faster data storage for improved performance.
- I/O system management: Input/Output device control.

<strong>File-System Management</strong>

- Files as storage units, organized in directories.
- File creation, deletion, and access controlled by the operating system.

<strong>Mass-Storage Management</strong>

- Secondary storage (HDDs, NVM) for backup.
- Disk operations (mounting, unmounting, scheduling).
- Tertiary storage (tapes, optical media) for long-term storage.

<strong>Cache Management</strong>

- Caching for performance improvement.
- Cache size and replacement policies.
- Cache coherency for data synchronization.

<strong>I/O System Management</strong>

- I/O subsystem conceals device peculiarities.
- Device drivers for specific hardware.
- Buffering, caching, and spooling for I/O operations.