Database Management Systems

Dr. Santhosh Manikonda
Department of CSE
School of Engineering

Malla Reddy University, Hyderabad

Syllabus
UNIT - V

- Disk Storage, Basic File Structures:
  - Introduction
  - Secondary Storage Devices
  - Buffering of Blocks
  - Placing File Records on Disk
  - Operations on Files.

Storage Hierarchy

- The collection of data in a computerized database requires physical storage.
- The Database Management System (DBMS) retrieves, updates, and processes data stored in these media.
- Computer storage media fall into two main categories:
  - Primary storage
  - Secondary storage.

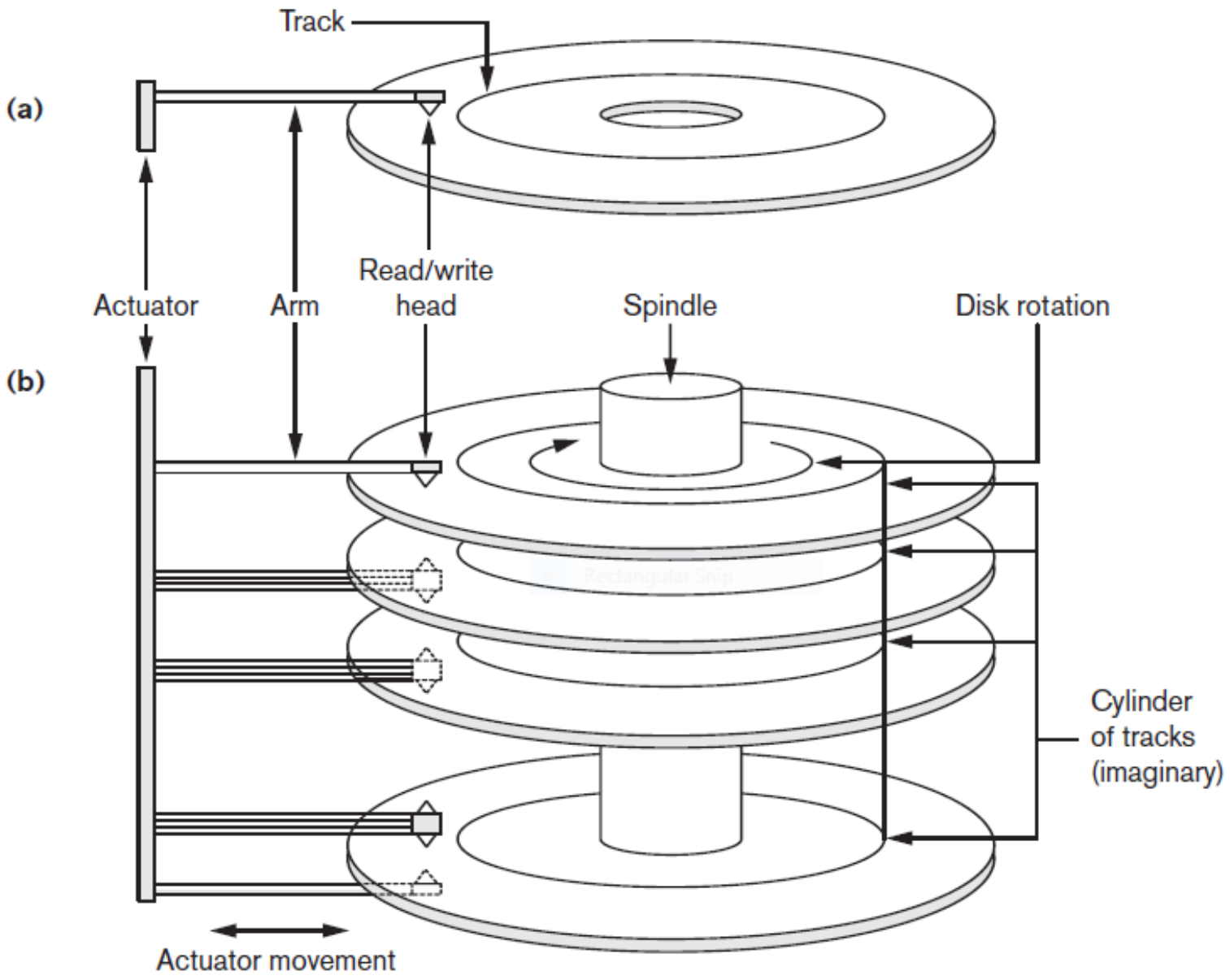
Primary Storage

- Includes storage media directly operated on by the computer's CPU.
- Examples are main memory and cache memories.
- Offers fast access to data but has limited storage capacity.
- Main memory capacities have been growing but remain more expensive with less capacity compared to secondary and tertiary storage.

Secondary Storage

- Includes magnetic disks, optical disks (CD-ROMs, DVDs), and tapes.
- Hard-disk drives are classified as secondary storage, while optical disks and tapes are considered tertiary storage.
- These devices have larger capacities, are cost-effective, but provide slower access to data compared to primary storage.
- Secondary storage has a larger capacity, lower cost, and slower data access than primary storage.
- Data in secondary storage cannot be processed directly by the CPU; it must first be copied into primary storage for processing

Secondary Storage



Secondary Storage

- A disk can be single-sided (storing data on one surface) or double-sided (both surfaces used).
- To boost storage, disks are grouped into a disk pack, with many disks and surfaces.
- Data is stored in concentric circles called tracks, and tracks with the same diameter on different surfaces form a cylinder.
- Disks have a few hundred to a few thousand tracks, each with a capacity ranging from tens of Kbytes to 150 Kbytes.
- Tracks are divided into smaller blocks or sectors.
- Block sizes, set during disk formatting, range from 512 to 8192 bytes.
- Disks may use a technique called Zone Bit Recording (ZBR) to optimize sector distribution.

Secondary Storage

- Disks are random access devices, and data transfer occurs in units of disk blocks.
- The disk read/write head is crucial, reading or writing data.
- Disk controllers manage disk drives and interface with the computer system.
- Locating data on a disk involves seek time, rotational delay (latency), and block transfer time.
- Seek time is the time to position the read/write head - *5 to 10 msec on desktops and 3 to 8 msecs on servers*
- Rotational delay depends on disk speed - *15,000 rpm, the time per rotation is 4 msec*
- Block transfer time is needed to move data.
- To improve efficiency, multiple consecutive blocks may be transferred together.
- Disk access times range from 9 to 60 msec.
- File structures aim to minimize block transfers, and organizing related information on contiguous blocks is crucial.

Secondary Storage

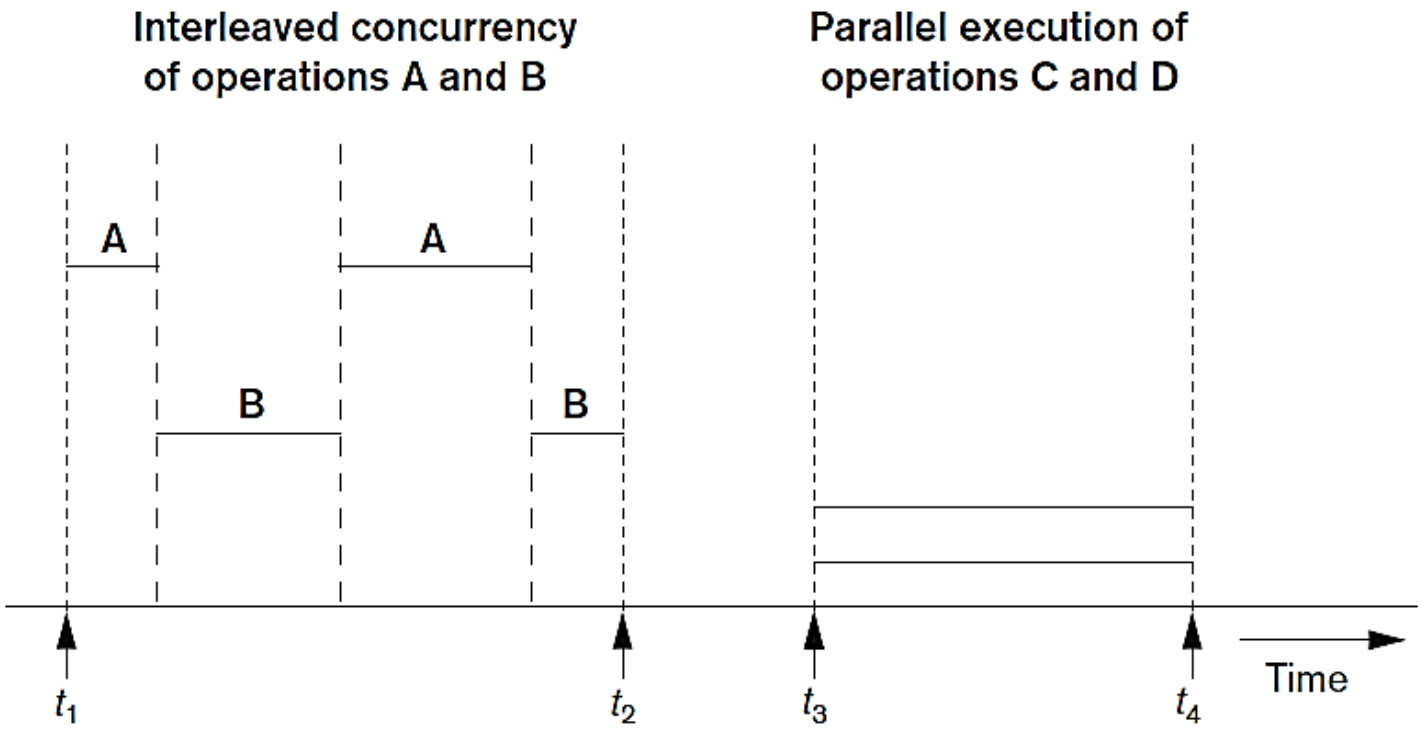
- A disk is a random-access device.
- Data transfer between main memory and disk occurs in units called disk blocks.
- A disk block's hardware address includes a cylinder number, track number (surface within the cylinder), and block number (position within the track).
- Modern disk drives often use a single number, the Logical Block Address (LBA), automatically mapped by the disk drive controller. LBA ranges from 0 to n+1 blocks (assuming n is the total disk capacity).
- The address of a buffer (a reserved area in main storage) is provided - For a read command, the disk block is copied into the buffer; for a write command, the buffer's contents are copied into the disk block.
- Occasionally, several consecutive blocks, known as a cluster, may be transferred as a unit. In such cases, the buffer size adjusts to match the cluster's byte count.

Buffering of blocks

- When multiple blocks need to be transferred from disk to main memory, using several reserved buffers in main memory can speed up the process.
- While one buffer is being read or written, the CPU can process data in another buffer.
- An independent disk I/O processor (controller) allows the transfer of a data block between memory and disk independently and in parallel to CPU processing.
- Two processes can proceed concurrently - *enhancing efficiency.*

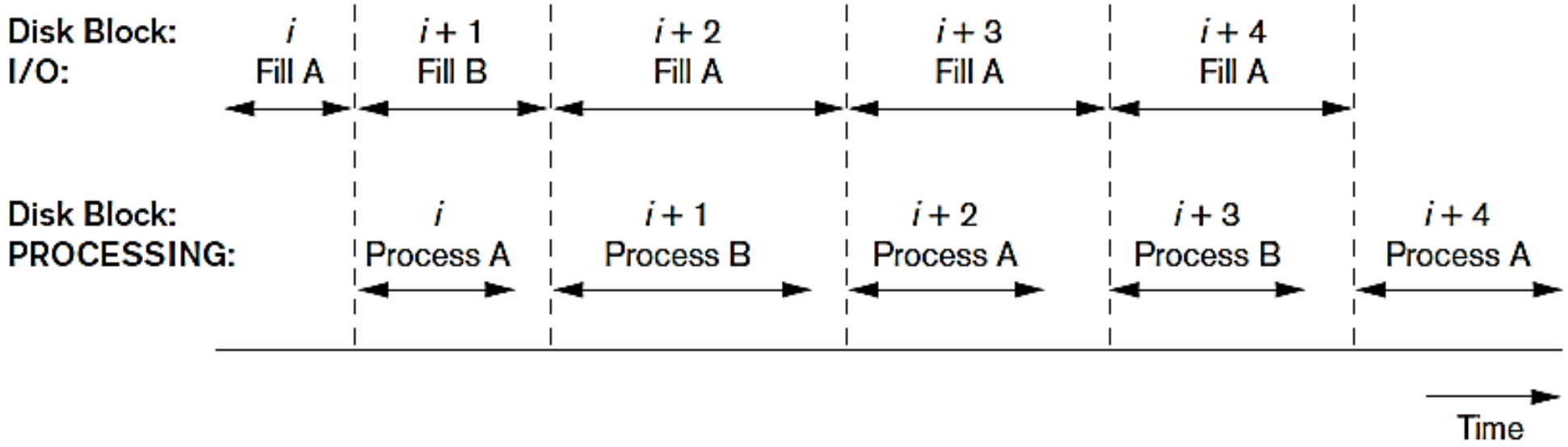
Buffering of blocks

- A and B - concurrently in an interleaved fashion
  - *Single CPU*
- C and D - concurrently in a parallel fashion
  - *A separate disk I/O processor or multiple CPU processors*
- Buffering is most beneficial when processes can run concurrently in a parallel fashion



Buffering of blocks

- Double buffering - parallel reading and processing, when the time to process a disk block in memory is less than the time to read the next block and fill a buffer.
- The CPU can start processing a block as soon as its transfer to main memory is complete, while the disk I/O processor simultaneously reads and transfers the next block into a different buffer.
- Double buffering enables continuous reading or writing of data on consecutive disk blocks, eliminating seek time and rotational delay for all but the first block transfer.
- The approach keeps data ready for processing, reducing waiting times in programs.



Placing Files Records on Disk

- Data is commonly stored as records, each comprising related data values or items.

- Records describe entities and attributes, for instance, an *EMPLOYEE* record representing an employee entity with fields like *Name*, *Birth_date*, *Salary*, or *Supervisor*.

- A record type or format definition is made up of field names and their corresponding data types.
- Data types, associated with each field, define the range of values a field can have.

- Standard programming data types include numeric (integer, long integer, floating point), string (fixed or varying length), Boolean, and date/time.

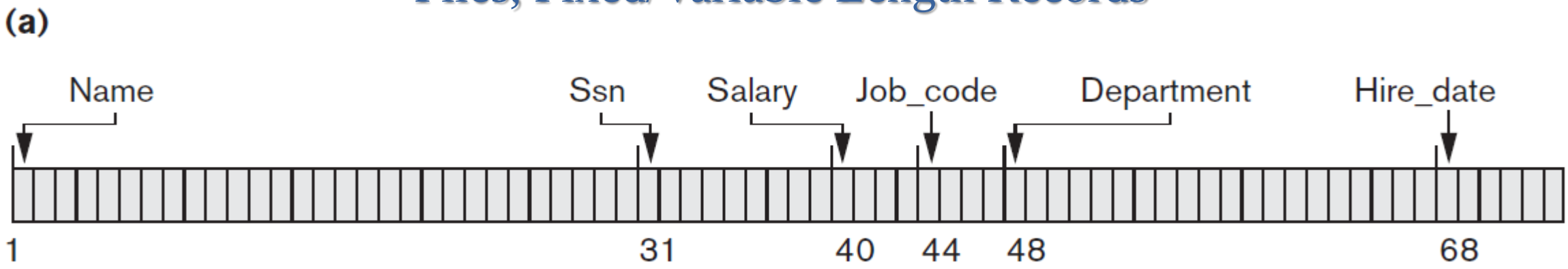
Placing Files Records on Disk

- The number of bytes required for each data type is fixed for a specific computer system.
- In a record type definition, fields are specified with their respective data types and sizes.
- For specific database applications, there may be a need to store large unstructured objects (BLOBs – Binary Large Objects), like images or audio streams. BLOBs are stored separately from the record with a pointer in the record.

Files, Fixed/Variable Length Records

- A file is a sequence of records
  - If all records in a file are of the same type and size, it's made up of fixed-length records.
  - If records have different sizes, it's made up of variable-length records.
- Reasons for variable-length records
  - Variable-length fields – *Name of employee*
  - Repeating fields – *Fields that may have multiple values*
  - Optional fields – *May or may not have values for all*
  - Mixed records of different types – *Different record types kept together*
- Fixed-length records have a predictable size, making field access easy for programs.
- For files logically requiring variable-length records, fixed-length records can be used with special values or placeholders for absent data.

Files, Fixed/Variable Length Records



Files, Fixed/Variable Length Records

- A file is a sequence of records
  - If all records in a file are of the same type and size, it's made up of fixed-length records.
  - If records have different sizes, it's made up of variable-length records.
- Reasons for variable-length records
  - Variable-length fields – *Name of employee*
  - Repeating fields – *Fields that may have multiple values*
  - Optional fields – *May or may not have values for all*
  - Mixed records of different types – *Different record types kept together*
- Fixed-length records have a predictable size, making field access easy for programs.
- For files logically requiring variable-length records, fixed-length records can be used with special values or placeholders for absent data.

Operations on Files

- Operations on files are categorized into retrieval and update operations.
- Retrieval operations focus on locating records without altering data, allowing examination and processing of field values.
- Update operations involve file modification through record insertion, deletion, or field value modification.
- Selection conditions, also known as filtering conditions, are employed in both retrieval and update operations to specify criteria for record selection.

File Operations in DBMS

- The specific operations for locating and accessing file records can differ across systems.
- Here are representative operations often utilized by high-level programs, such as DBMS software.
- The descriptions may refer to program variables in the context of these operations.

File Operations in DBMS

- Open: Prepares the file for reading or writing, allocates buffers, retrieves the file header, and sets the file pointer to the beginning.
- Reset: Sets the file pointer of an open file to the beginning.
- Find (Locate): Searches for the first record satisfying a condition, transfers the block with that record into a buffer, and sets it as the current record.
- Read (Get): Copies the current record from the buffer to a program variable, potentially advancing the current record pointer to the next record.

File Operations in DBMS

- FindNext: Searches for the next record satisfying a condition, transfers the block with that record into a buffer, and sets it as the current record.
- Delete: Deletes the current record and updates the file on disk to reflect the deletion.
- Modify: Modifies field values for the current record and updates the file on disk.
- Insert: Inserts a new record by locating the insertion point, transferring the block into a buffer, writing the record, and updating the disk.
- Close: Completes file access by releasing buffers and performing cleanup operations.