Database Management System – 46 (Hashing Techniques – External Hashing)

Ajay James
Asst. Prof in CSE
Government Engineering College Thrissur

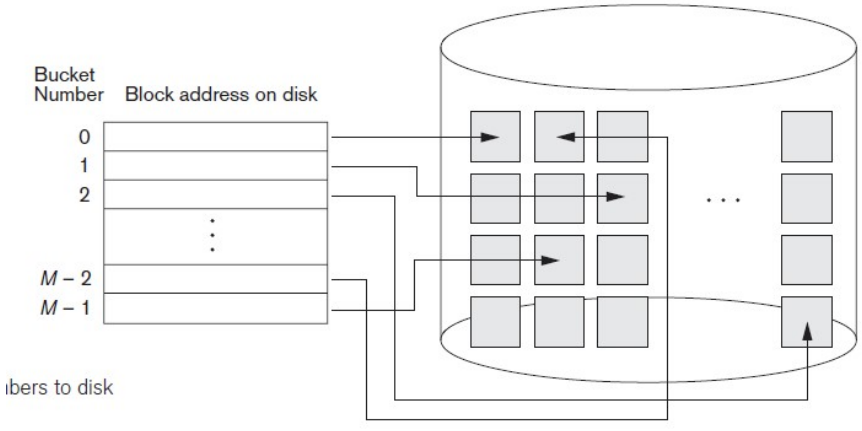
Outline

- External hashing for disk files
- Extendible hashing
- Dynamic hashing
- Linear hashing

External Hashing for Disk Files

- Hashing for disk files is called external hashing
- Target address space is made of buckets, each of which holds multiple records
- Bucket is either one disk block or a cluster of contiguous disk blocks
- Hashing function maps a key into a relative bucket number
- Table maintained in the file header converts the bucket number into the corresponding disk block address

External Hashing (static hashing)



Handling overflow for buckets

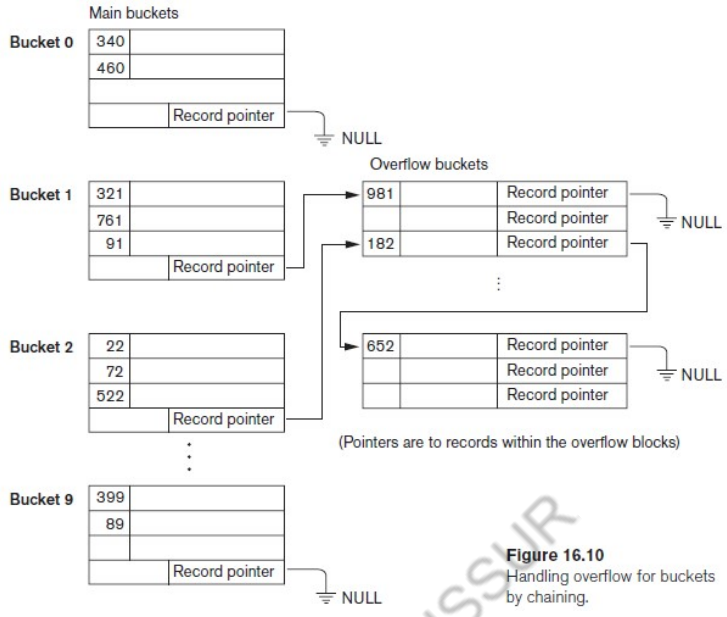


Figure 16.10
Handling overflow for buckets by chaining.

Hashing techniques that allow dynamic file expansion

- Extendible hashing
- Dynamic hashing
- Linear hashing

Extendible hashing

- A type of directory—an array of 2^d bucket addresses—is maintained
- **d** is called the **global depth** of the directory
- Integer value corresponding to the **first (highorder) d bits** of a hash value is used as an index to the array
- Address in that entry determines the bucket in which the corresponding records are stored
- Does not have to be a distinct bucket for each of the 2^d directory locations
- Several directory locations with the same first **d'** bits for their hash values may contain the same bucket
- A local depth **d'**—stored with each bucket—specifies the number of bits on which the bucket contents are based

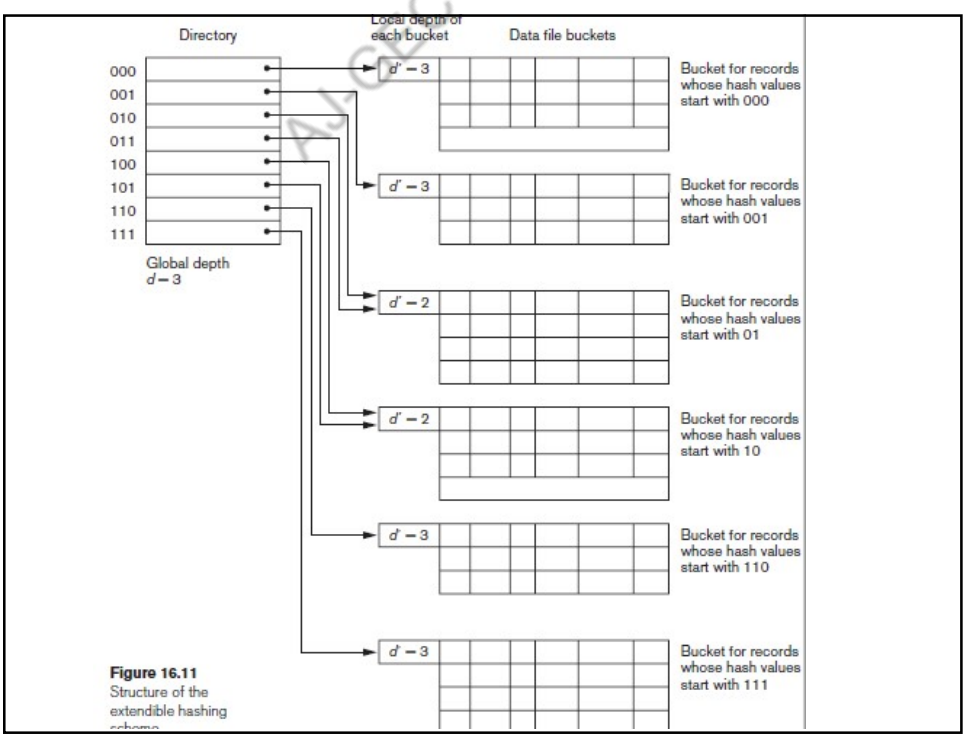


Figure 16.11
Structure of the extendible hashing

Extendible hashing

- Advantages
  - Performance of the file does not degrade as the file grows
  - No space is allocated in extendible hashing for future growth
  - Splitting causes only minor reorganization
  - More expensive is when the directory has to be doubled
- Disadvantage
  - Directory must be searched before accessing the buckets themselves, resulting in two block accesses

Dynamic Hashing

- Addresses of the buckets are either the n high-order bits or n − 1 high-order bits
- Storage of records in buckets is somewhat similar to extendible hashing
- Difference - organization of the directory
- Maintains a tree-structured directory with two types of nodes:
  - Internal nodes that have two pointers—the left pointer corresponding to the 0 bit (in the hashed address) and a right pointer corresponding to the 1 bit
  - Leaf nodes—these hold a pointer to the actual bucket with records.

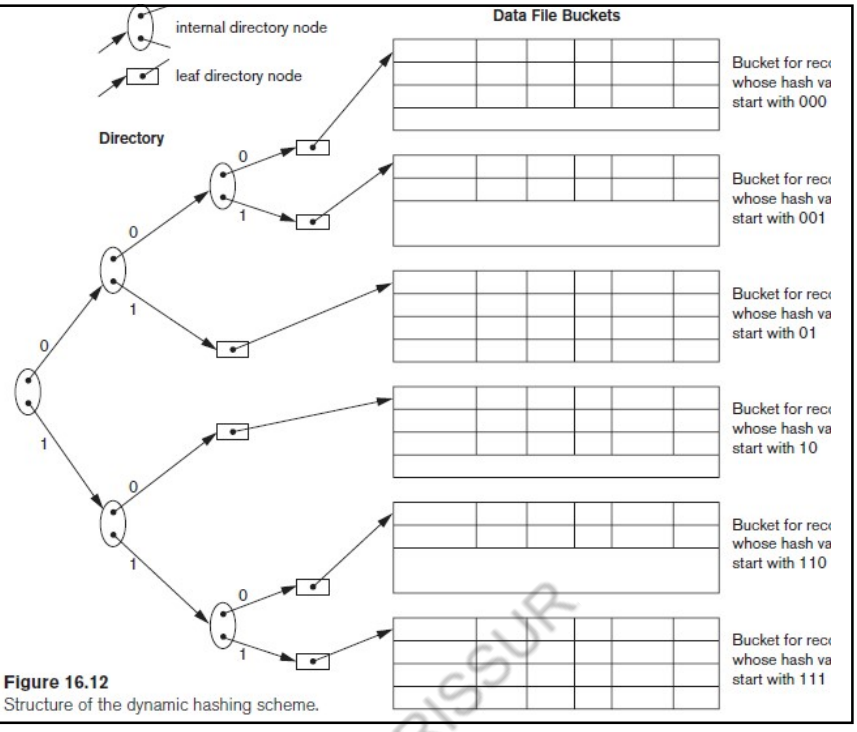


Figure 16.12
Structure of the dynamic hashing scheme.

Linear Hashing

- To allow a hash file to expand and shrink its number of buckets dynamically without needing a directory
- File starts with M buckets numbered 0, 1, … , M − 1 and uses the mod hash function h(K) = K mod M
- Called the initial hash function hi
- When a collision leads to an overflow record in any file bucket, the first bucket in the file—bucket 0—is split into two buckets: the original bucket 0 and a new bucket M at the end of the file
- Records originally in bucket 0 are distributed between the two buckets based on a different hashing function $h_{i+1}(K) = K \bmod 2M$.

Reference

- Elmasri R. and S. Navathe, Database Systems: Models, Languages, Design and Application Programming, Pearson Education 6th edition and 7th edition

Thank you