Data Management and File Organization

Sequential Files

Part one: Pile files

Topics

- Definitions:
  - Entity
  - Attribute
  - Record
- Files, Sequential Files
- Sequential File Operations and Timings

Review: Motivation

- Large data cannot be stored in the main memory of computers because:
  - The capacity of the main memory (RAM) is small
  - The data in the main memory is lost when we turn-off the computer
- Solution:
  - Using files stored on hard disks

Questions:

- What is a file?
- How do we put data in a file?
- Is there any structure in the data stored in a file?
- How can we make file access fast?

- We will find answers to these questions

Basic Definitions

- Entity: Anything that we store some data about
- Attribute: Any property that helps to identify an entity. Ex. Name, height, age
- Data: Any value given to an attribute, Ex. ‘Ali’ for attribute Name
- Record: A group of attribute values which uniquely identifies an entity.

Ex. For entity ‘student’, the record is

Student <Student ID, name, surname, major, year started, address, phone>

<200511777, Ali, Yildiz , Computer Eng. , 2005, Ankara, 1234567 >

Basic Definitions

- File: A set of related records.

  Ex. Student file (Set of student records)

  Hospital file (Set of patient records)

Exceptions: Text files and binary files

Sequential Files

- Definition: A sequential file is a file which is read from beginning to end.

- Types:
  - Unsorted sequential files (Pile files): A set of records with no order
  - Sorted sequential files : The records are sorted in the order of an attribute

File Header

Header is part of the file which includes data such as:

- Record size
- Block size
- Number and type of indexes
- Address of the last block

Header is read into memory when the file is opened

Sequential File Operations and Timings

- Fetch one record T_F
- Fetch next record T_N
- Insert a record T_I
- Update a record T_U
- Delete a record T_D
- Exhaustive reading of the file T_X
- Re-Organize a file T_Y

Fetch One Record

- Find and read a record given an attribute value.
  - Example. Find student record with Student ID=200612345
- In a pile file on average half of the blocks are read
- T_F = s + r + b/2*ebt

  b: number of blocks in the file

Example

- Find T_F given:
  - Total number of records = 100,000
  - Blocking factor (Bfr) = 6
  - Number of blocks (b) = 100,000/6 =16667
  - ebt = 0.84 msec
  - s=16msec
  - r=8.3msec

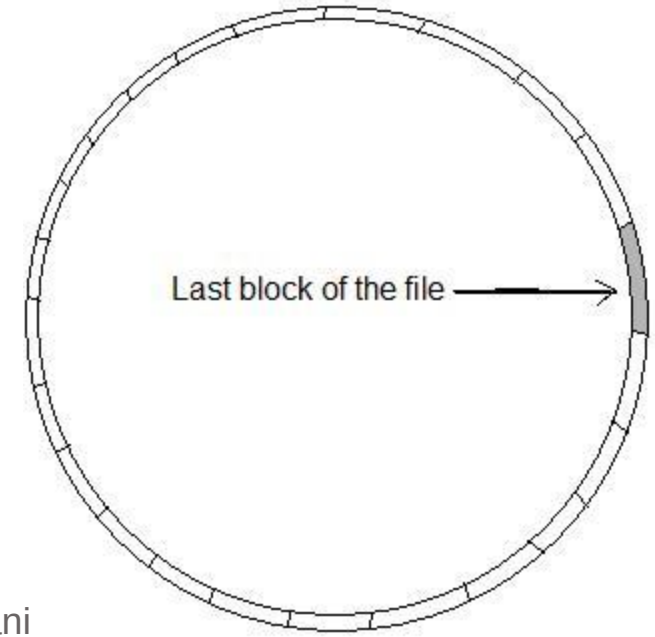
Fetch Next Record

- Find and read the next record in order of an attribute value.

- If the file is not sorted $T_N = T_F$ (Each fetch is independent)

Insert a New Record

- Insert is always done at the last block
  - Time to read the last block = s + r + btt
  - Time to write back the block = 2r
  - T_I = s + r + btt + 2r



Update a Record

- To update, first the block is read, then the record is updated and the block is written back
- Time to read the block = s + r + b/2*ebt = T_F
- Time to write back the block = 2r
- $T_U = T_F + 2r$

Delete a Record

- To delete a record, we mark it as deleted
- First read the block (s + r + b/2*ebt)
- Update the mark and write the block (2r)
- $T_D = T_F + 2r$

Mark	Record
1	Rec1
0	Rec2
0	Rec3
1	Rec4

Exhaustive Reading of a File

- Case 1: Read records without any attribute order

$$T_X \text{ (Beginning to End)} = b*ebt + s + r$$

Example:

b = 16667

ebt = 0.84 msec

s=16 msec

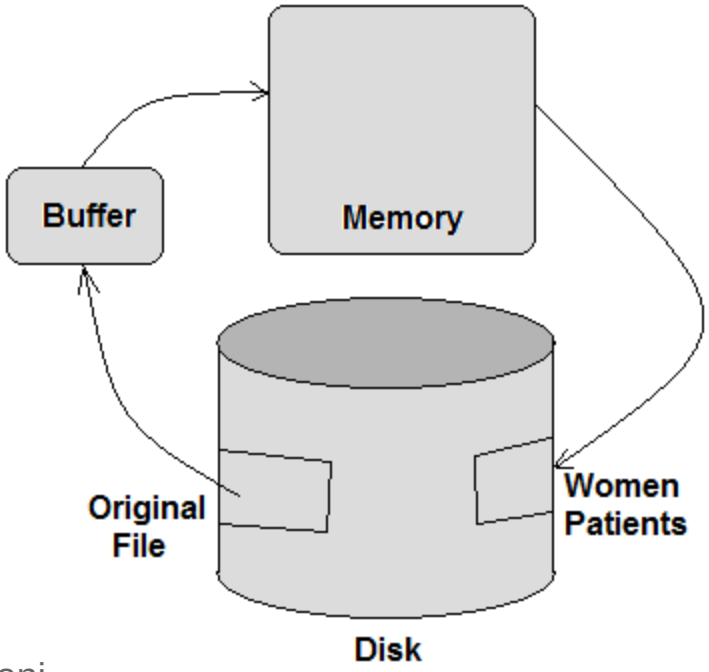
r=8.3 msec

Example

- Assume a hospital file records include illness, and duration of stay in hospital too. To find the average duration of stay in hospital for each illness, an exhaustive reading of the file without any attribute order is used

Example

- Write records of the hospital file which show a female patient in a new file. An exhaustive reading without any attribute order is used here too.



Exhaustive Reading of a File

- Case 2: with the order of an attribute value
- Tx (order of an attribute) = Number of records x T_N

Example

- Find the time needed to read the hospital file with 100,000 records.
- Assume
  - Bfr = 10
  - ebt = 0.84 msec
  - s=16 msec
  - r=8.3 msec

Solution

- Case 1: Without any attribute order
  - Tx = b*ebt + s + r
  - b = 100,000 /10 = 10,000 (number of blocks, Bfr is 10)
  - Tx = 10,000 * 0.84 + 16 + 8.3 = 8.424 s

Solution

- Case 2: with the order of an attribute
  - Tx = 100,000 * T_N
  - Assume T_N = 4224 ms
  - About 4.89 days

Re-Organizing a File

- In delete operation records are not physically deleted from the file.
- Operations in a file with many marked records are slow
- In re-organizing a file, marked records are deleted physically by copying active records to a new file.
- Old file is deleted

Re-organizing files

- Time to read the file= s + r + b*ebt
- Time to write active records=s+r+(n/Bfr)*ebt

  (n is the number of active records)

  T_Y = s+r+b*ebt + s+r+(n/Bfr)*ebt

- If two disks are available then
  - Read from disk one and write to disk two at the same time
  - T_Y = s+r+b*ebt

Assignment

- Assume a pile file has 100,000 records in it. Bfr=5, and 25% of the records are marked as deleted. Find T_F for this file.
- Now assume the file has been re-organized. Find T_F again and compare with your answer before re-organizing the file.
- Use: s=16msec, r=8.3msec , ebt = 0.84msec

Summary

- File is a set of related records
- Sequential files are read from beginning to end
- File I/O operation timings depend on:
  - File size
  - Blocking factor
  - Order of reading records

Questions?