Design and Specification of File Structures

Dr. Prajish Prasad, FLAME University

File Structures

- File Structure -
  Representations for data in files + **operations** for accessing the data.
- Allows applications to read, write and modify data
- **Supports finding the data** that matches some search criteria or **reading through the data** in some particular order.

Material taken from - https://www.site.uottawa.ca/~nat/Courses/csci2111.html

Data Storage

What are different types of storage you are familiar with?

- Hard disk
- USB
- SSD
- Memory cards
- CD
- Floppy disks
- Cloud storage
- Game cartridges

Data Storage

- Primary Storage ➔ Memory
- Secondary Storage ➔ Hard Disk / Tape / CDRom / USB
- Tertiary Storage ➔ Archival data - Offline Disk/Tape/ USB not directly available to the computer.
- Storage on the Cloud

Material taken from - https://www.site.uottawa.ca/~nat/Courses/csci2111.html

Characteristics of Primary Storage

Main Memory

Storage - 8/16 Gb

Speed - 25.6 GB/sec

Characteristics of Secondary Storage

Hard Disk

Storage - 1Tb

Speed - 160 MB/sec

Main Memory

Storage - 8/16 Gb

Speed - 25.6 GB/sec

Hard Disk

Storage - 1Tb

Speed - 160 MB/sec

Memory vs Secondary Storage

- Secondary storage such as disks can pack thousands of megabytes in a small physical location.

- Computer Memory (RAM) is limited.

- However, relative to Memory, access to secondary storage is extremely slow
  [E.g., getting information from RAM takes 10^{-9} seconds (= 120 nanoseconds) while getting information from Disk takes $30. 10^{-3}$ seconds (= 30 milliseconds)]

Material taken from - https://www.site.uottawa.ca/~nat/Courses/csci2111.html

Key Challenge:
Improving Secondary Storage Access Time

Solution: Improving File Structures

General Goals

- Get the information we need with **one access to the disk**.
- If that's not possible, then get the information with **as few accesses as possible**.
- **Group information** so that we are likely to get everything we need with only one trip to the disk.

Material taken from - https://www.site.uottawa.ca/~nat/Courses/csci2111.html

Activity

Mimicking file system design using cards

- Get the information we need with **one access to the disk**.
- If that's not possible, then get the information with **as few accesses as possible**.
- **Group information** so that we are likely to get everything we need with only one trip to the disk.

Fixed vs Dynamic Files

- It is relatively easy to come up with file structure designs that meet the general goals when the files never change.

- When files grow or shrink when information is added and deleted, it is much more difficult.

Material taken from - https://www.site.uottawa.ca/~nat/Courses/csci2111.html

History of File Structures

Tape Structures

- Early Work assumed that files were on tape.

- Access was sequential and the cost of access grew in direct proportion to the size of the file.

Material taken from - https://www.site.uottawa.ca/~nat/Courses/csci2111.html

Disks and Indexes

- As files grew very large, unaided sequential access was not a good solution.
- Disks allowed for **direct access**.
- Indexes made it possible to keep a list of **keys** and **pointers** in a small file that could be searched very quickly.
- With the key and pointer, the user had direct access to the large, primary file.

Tree Structures

- As indexes also have a sequential flavour, when they grew too much, they also became difficult to manage.
- The idea of using **tree structures to manage the index** emerged in the early 60's.
- However, trees can grow very unevenly as records are added and deleted, resulting in long searches requiring many disk accesses to find a record.

Material taken from - https://www.site.uottawa.ca/~nat/Courses/csci2111.html

Balanced Trees

- In 1963, researchers came up with the idea of **AVL trees** for data in memory.

- AVL trees, however, did not apply to files because they work well when tree nodes are composed of single records rather than dozens or hundreds of them.

Material taken from - https://www.site.uottawa.ca/~nat/Courses/csci2111.html

B Trees

- In the 1970's came the idea of B-Trees

- Require an $O(\log_k N)$ access time where
  - N - number of entries in the file
  - K - number of entries indexed in a single block of the B-Tree structure
- B-Trees can guarantee that one can find one file entry among millions of others with only 3 or 4 trips to the disk

Hash Tables

- Retrieving entries in 3 or 4 accesses is good, but it does not reach the goal of accessing data with a single request.

- From early on, Hashing was a good way to reach this goal with files that do not change size greatly over time.

- Extendible Dynamic Hashing guarantees one or at most two disk accesses no matter how big a file becomes.

Creating Files

Go to your FLAME gradebook/student management system

Create a prototype of all/most of the **data** in the system using classes

Each student object should be written onto the disk as a file

Group Presentation - 17th Jan - 10 marks (towards the Graded Assignment Section)

Prepare a 15 minute presentation on the storage device assigned to your group

Go into the details of -

- The history
- The physical structure of the storage
- The logical structure of the storage (how is data stored)
- Operations on the storage (how data read,write,modify etc. happens)
- Performance

```
Jash and Nishtha: Hard Disks
Shizuka and Melwina: SSD
Viraj and Manu: Magnetic Tapes
Amoga and Kushal: RAM and ROM
Jeet and Archit: Floppy Disks
Abhyuday and Suyash: CD-ROM
```