

COMP282

Advanced Data Structures

Lecture 09b

File Structures and External Storage

Introduction

- Normally memory is used to hold and organize data
- Files can also be used to hold and organize data.
 - Data exists beyond execution of the program
 - External storage generally provides far, far more storage space than does memory. (120Gig vs $\frac{1}{2}$ gig)
 - Far, far slower to access than internal memory

File access

- Sequential Access
 - Data is stored in blocks (or “nodes”) and there is some information in each block that allows you to locate the next block.
 - Operates similarly to a linked list.
- Random Access
 - Data is stored according to some predetermined addressing scheme
 - Operates similarly to an array.
 - Not all external storage supports truly random access

Records

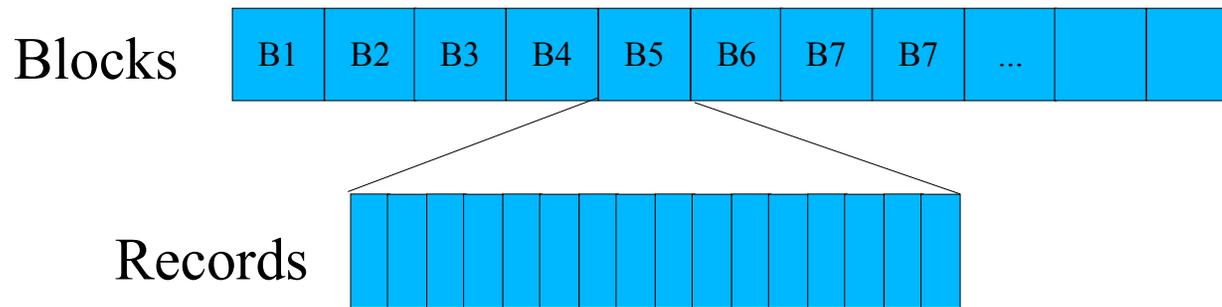
- Data files generally consist of one or more “records”.
 - Records can be a simple value or they can be a complex aggregate structure such as an “employee record”
 - We assume that all the records are of the same type for simplicity.

Blocks

- Records are organized into file structures known as “blocks”.
 - The size, location and configuration of a block is usually determined by the Operating System and underlying hardware.
 - Since programs have no control over this size the number of records that a program can store in a block is determined by the size of each record.
 - Blocks with simple value records can contain more records per block than records of aggregate structures.

Topology

- Basic topology



- Numbering of blocks and records in conjunction with fixed sizes of blocks and records allows for the computation of an offset amount to locate any particular block.
- Topology is similar to an array of arrays.

Cacheing

- Book assumes that you can only read or write to the external storage in full blocks.
 - This is usually a good model and matches what the operating system usually does.
 - Operating systems don't waste time fetching a single byte for a read request. If you ask for a single byte then an entire record (or more) is read. The single byte is returned and the rest of the data is cached in memory.
 - One of the reasons a modern operating system such as Unix or Window\$ XP require so much memory.

Buffers

- Programs can't always rely on the operating system to buffer data.
 - Its not efficient
 - Program algorithm may require blocks of data at a time.
- Programs must allocate buffers sufficiently large enough to store an entire block at once.
- The buffered data is only a copy of the data stored externally.

Efficiency of Algorithms

- Due to the speed limits of external storage the time required to fetch or write a block of data is far longer than the time required to operate on the Block's data
 - It is therefore necessary that attempts are made to minimize the number of read/write operations that an algorithm is required to perform to accomplish its goal.

What needs External Storage?

- Many tasks require vast amounts of data
- So much data that it requires external storage
 - In fact sometimes more external storage than can be accessed by a single machine
- RDMS: Relation Database Management Systems
- Simulation Data
- Game Theories
- Sorting and Hashing are two examples we will cover in the following lectures.