

7 Reclaiming Spaces in Files

Copyright © 2004, Binnur Kurt

Strategies for Record Deletion

2. Deleting Fixed-Length Records and Reclaiming Space Dynamically

- How to use the space of deleted records for storing records that are added later?
- Use an “AVAIL LIST”, a linked list of available records.
- A header record stores the beginning of the AVAIL LIST
- When a record is deleted, it is marked as deleted and inserted into the AVAIL LIST. The record space is in the same position as before, but it is logically placed into AVAIL LIST

File Organization

192

Motivation

- ▶ Let us consider a file of records (fixed length or variable length)
- ▶ We know how to create a file, how to add records to a file, modify the content of a record. These actions can be performed physically by using the various basic file operations we have seen (fopen, fclose, fseek, fread, fwrite)
- ▶ What happens if records need to be deleted?
- ▶ There is no basic operation that allows us to remove part of a file. Record deletion should be taken care by the program responsible for file organization

File Organization

190

Example

List Header

RRN=4

Simpson	Seinfeld	-1	Cramer	2	Edwards
0	1	2	3	4	5

If we add a record, it can go to the first available spot in the AVAIL LIST where RRN=4.

File Organization

193

Strategies for Record Deletion

- ▶ How to delete records and reuse the unused space?
- ### 1. Record Deletion and Storage Compaction
- Deletion can be done by marking a record as deleted
 - Note that the space for the record is not released, but the program that manipulates the file must include logic that checks if record is deleted or not.
 - After a lot of records have been deleted, a special program is used to squeeze the file-that is called **Storage Compaction**

File Organization

191

Strategies for Record Deletion

3. Deleting Variable-Length Records

- Use an AVAIL LIST as before, but take care of the variable-length difficulties
- The records in AVAIL LIST must store its size as a field.
- RRN can not be used, but exact byte offset must be used
- Addition of records must find a large enough record in AVAIL LIST.

File Organization

194

Reclaiming Spaces in Files 7

Example

List Header 42

Simpson|B|Seinfeld|J|*-1|10|Schumaer|M|*21|30|

0	1	2	3	4
10 bytes	11 bytes	10B	11B	30 bytes

Addition of records must find a large enough record in AVAIL LIST.

File Organization
195

Reclaiming Spaces in Files 7

Placement Strategies for New Records

3. Worst-Fit Strategy

- AVAIL LIST is sorted by decreasing order of size.
- Largest record is used for holding new record; unused space is placed again in AVAIL LIST.

► Example:

- AVAIL LIST: size=60,size=50,size=22,size=10
- record to be added: size=20
- Which record from AVAIL LIST is used for the new record?

File Organization
198

Reclaiming Spaces in Files 7

Placement Strategies for New Records

► There are several strategies for selecting a record from AVAIL LIST when adding a new record:

1. First-Fit Strategy

- AVAIL LIST is not sorted by size.
- First record large enough to hold new record is chosen.

► Example:

- AVAIL LIST: size=10,size=50,size=22,size=60
- record to be added: size=20
- Which record from AVAIL LIST is used for the new record?

File Organization
196

Reclaiming Spaces in Files 7

How to Choose Between Strategies

► We must consider two types of fragmentation within a file:

► **Internal Fragmentation**

- wasted space within a record.

► **External Fragmentation**

- space is available at AVAIL LIST, but it is so small that cannot be reused.

File Organization
199

Reclaiming Spaces in Files 7

Placement Strategies for New Records

2. Best-Fit Strategy

- AVAIL LIST is sorted by size.
- Smallest record large enough to hold new record is chosen.

► Example:

- AVAIL LIST: size=10,size=22,size=50,size=60
- record to be added: size=20
- Which record from AVAIL LIST is used for the new record?

File Organization
197

Reclaiming Spaces in Files 7

Study this!

► For each of the following approaches, which type of fragmentation arises, and which placement strategy is more suitable?

► When the added record is smaller than the item taken from AVAIL LIST:

► Leave the space unused within record

- type of fragmentation: **internal**
- suitable placement strategy: **best-fit**

► Return the unused space as a new available record to AVAIL LIST

- type of fragmentation: **external**
- suitable placement strategy: **worst-fit**

File Organization
200

Ways of Combating External Fragmentation

- ▶ Coalescing the Holes
 - if two records in AVAIL LIST are adjacent, combine them into a larger record
- ▶ Minimize fragmentation by using one of the previously mentioned placement strategies
 - for example: worst-fit strategy is better than best-fit strategy in terms of external fragmentation when unused space is returned to AVAIL LIST