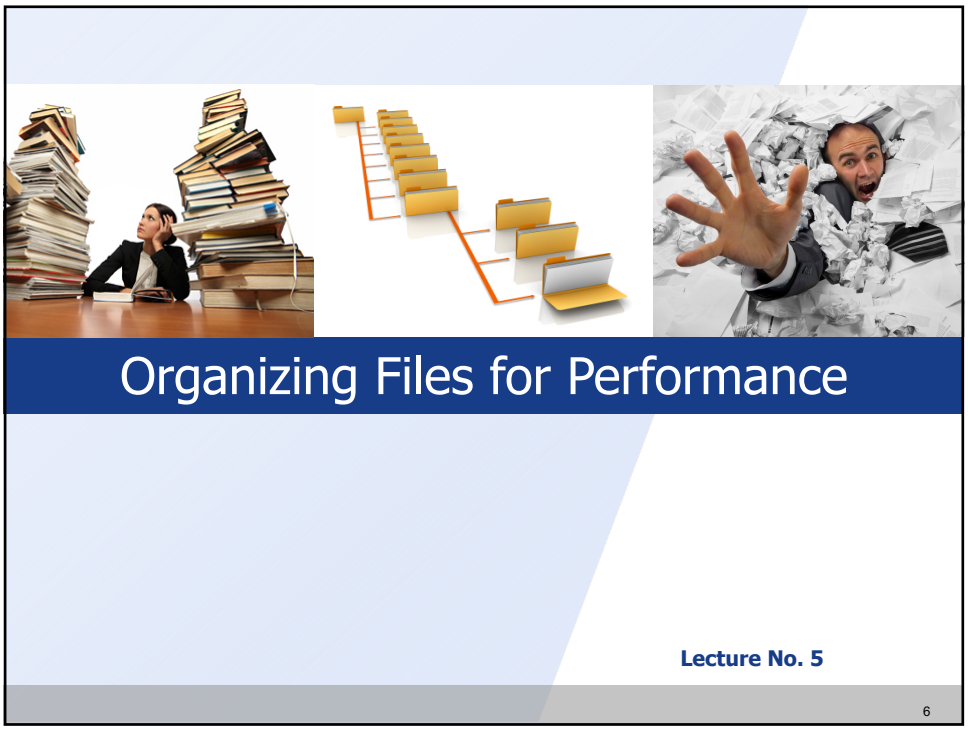
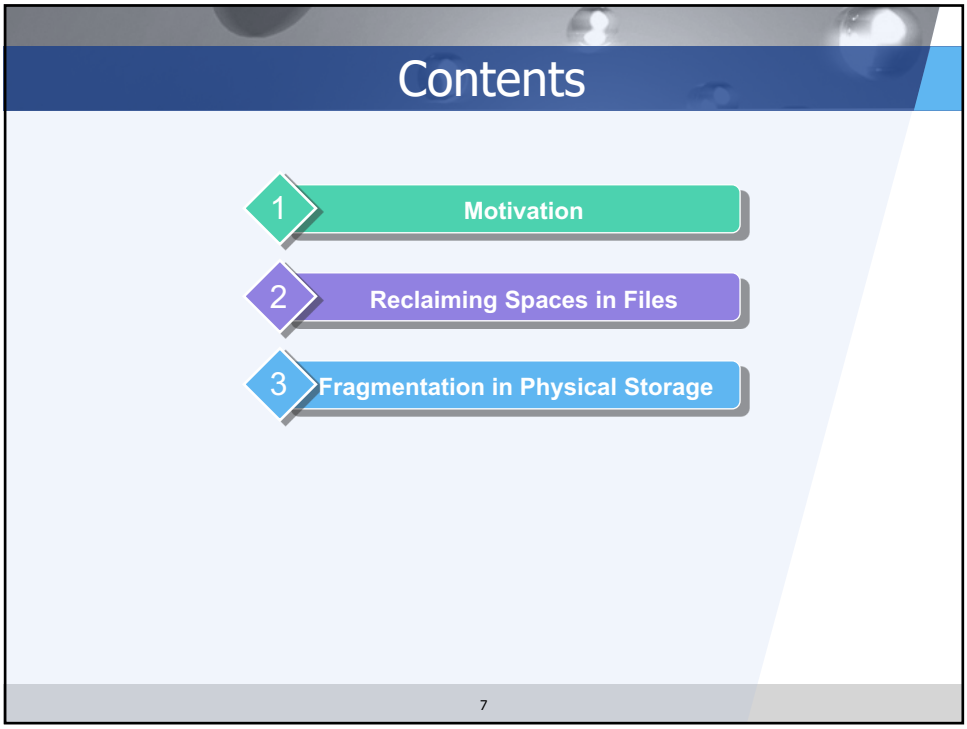
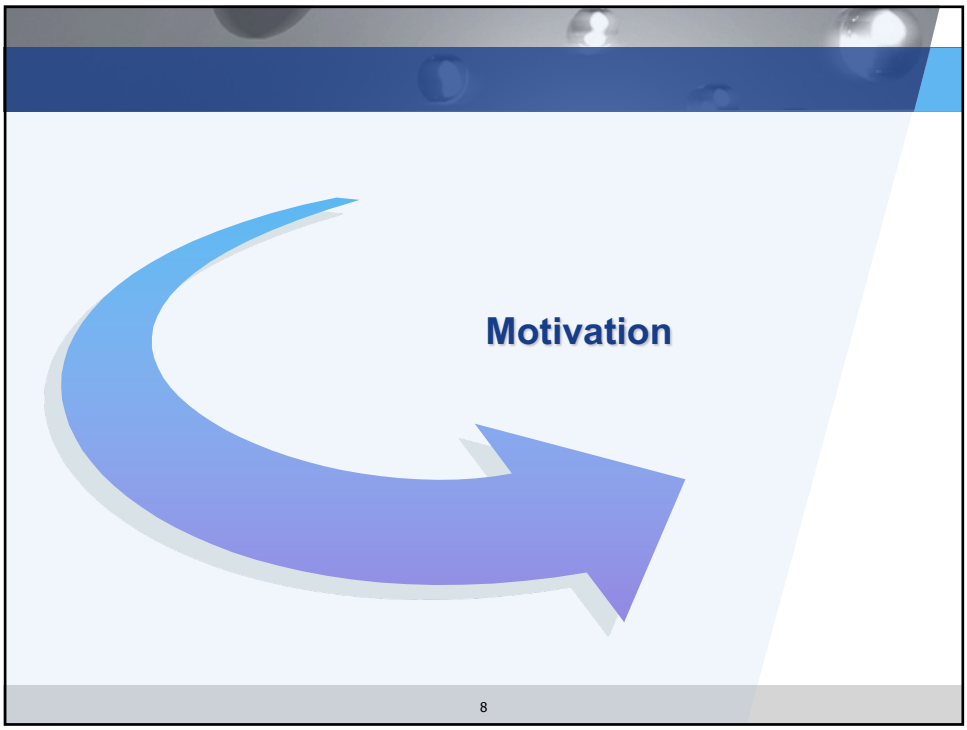

Organizing Files for Performance
Lecture No. 5
6

6


Contents
1 Motivation
2 Reclaiming Spaces in Files
3 Fragmentation in Physical Storage
7

7



8

Motivation

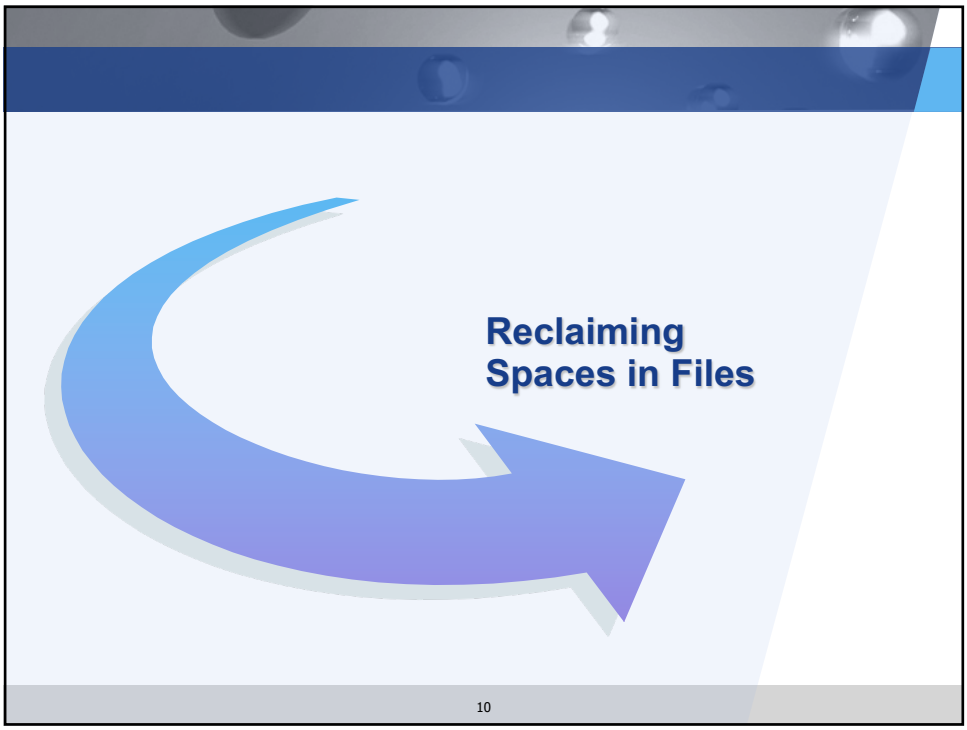
- Let us consider a file of records (fixed length or variable length)
- We know how to create a file, how to add records to a file, modify the content of a record. These actions can be performed physically by using the various basic file operations we have seen (open, close, seek, read, write)

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

9

9

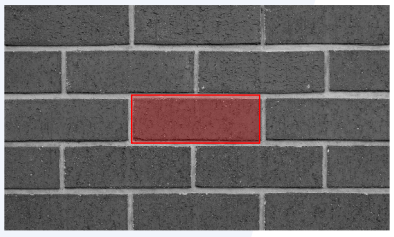


10

Strategies for Record Deletion

- How to delete records and reuse the unused space?

1. Record Deletion and Storage Compaction
2. Deleting Fixed-Length Records and Reclaiming Space Dynamically
3. Deleting Variable-Length Records



File of fixed length records

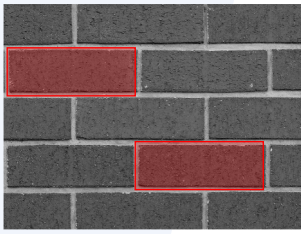
Deleted Record

11

Strategies for Record Deletion

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.



File of fixed length records

- After a lot of records have been deleted, a special program is used to squeeze the file-that is called **Storage Compaction**

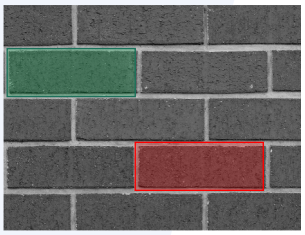
12

12

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



File of fixed length records

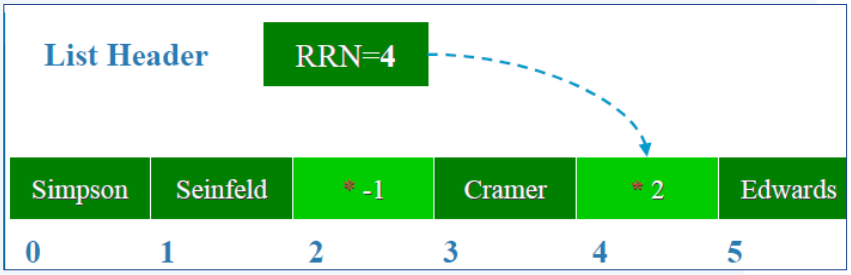
- When a record is deleted, it is marked as deleted and inserted into the AVAIL LIST. The record space is in the same position as before, but it is logically placed into AVAIL LIST

13

13

Strategies for Record Deletion

2. **Deleting Fixed-Length Records and Reclaiming Space Dynamically**



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

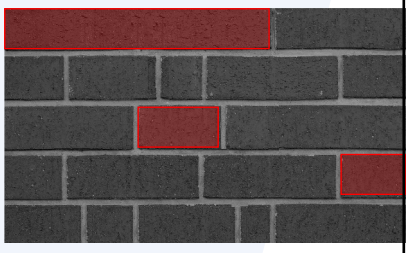
14

14

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.



File of variable length records

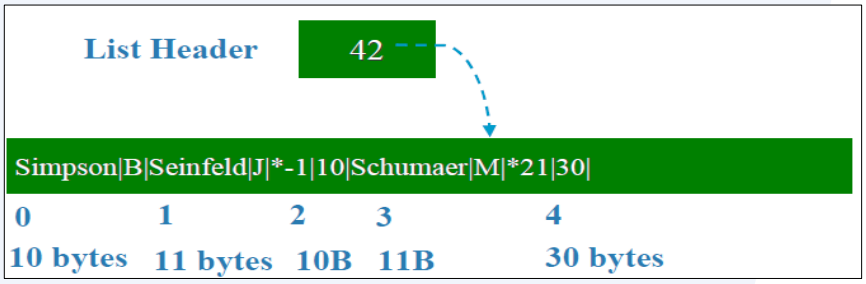
- RRN can not be used, but exact byte offset must be used
- Addition of records must find a large enough record in AVAIL LIST.

15

15

Strategies for Record Deletion

3. **Deleting Variable-Length Records**



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

16

16

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?

17

17

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?

18

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?

19

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.

20

20

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

21

21

22

Physical File Storage

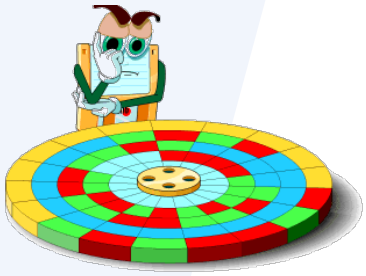
- Each of your disks contains its own index file so that information about its contents is always available when the disk is in use.
- When you save a file, the operating system looks at the index file to see which clusters are empty. It selects one of these empty clusters, records the file data there, and then revises the index file to include the new file name and its location.

23

23

Physical File Storage

- A file that does not fit into a single cluster spills over into the next contiguous (meaning adjacent) cluster, unless that cluster already contains data.
- When contiguous clusters are not available, the operating system stores parts of a file in noncontiguous (nonadjacent) clusters.



24

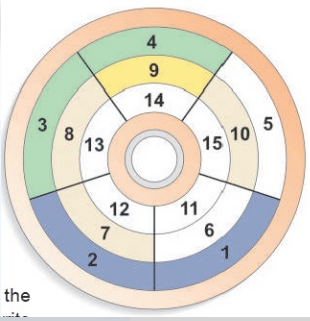
24

Physical File Storage

Master File Table

File	Cluster	Comment
MFT	1	Reserved for MFT files
DISK USE	2	Part of MFT that contains a list of empty sectors
Bio.txt	3, 4	Bio.txt file stored in clusters 3 and 4
Jordan.wks	7, 8, 10	Jordan.wks file stored noncon-tiguously in clusters 7, 8, and 10
Pick.bmp	9	Pick.bmp file stored in cluster 9

Bio.txt is stored in contiguous clusters.
Jordan.wks is stored in noncontiguous clusters.
A computer locates and displays the *Jordan.wks* file by looking for its name in the Master File Table.



the

25

25

What happens when a file is deleted

- When a file is deleted, the operating system simply changes the status of the file's clusters to "empty" and removes the file name from the index file.
- The file name no longer appears in a directory listing, but the file's data remains in the clusters until a new file is stored there.
- It is possible to purchase utilities that recover a lot of this supposedly deleted data.

26

26

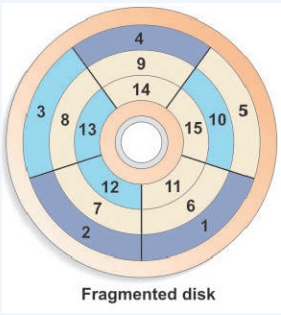
Fragmentation & Defragmentation

- As a computer writes files on a disk, parts of files tend to become scattered all over the disk.
- These **fragmented files** are stored in noncontiguous clusters. Drive performance generally declines as the read-write heads move back and forth to locate the clusters containing the parts of a file.
- To regain peak performance, you can use a **defragmentation utility**, such as Windows Disk Defragmenter, to rearrange the files on a disk so that they are stored in contiguous clusters

27

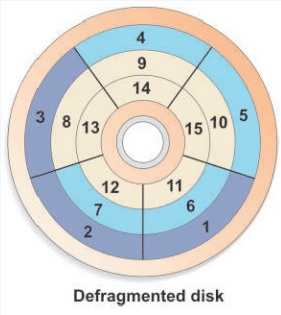
27

Fragmentation & Defragmentation



Fragmented disk

On the fragmented disk, the purple, orange, and blue files are stored in noncontiguous clusters.



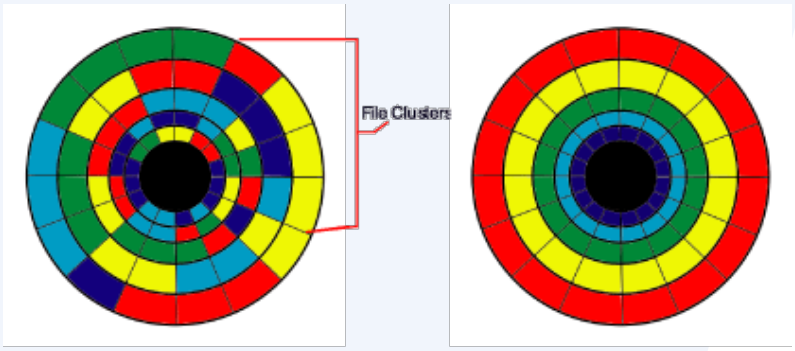
Defragmented disk

When the disk is defragmented, the sectors of data for each file are moved to contiguous clusters.

28

28

Fragmentation & Defragmentation



Fragmented Files

Defragmented Files

29

29