

File Management

1

File Management

- File management system consists of system utility programs that run as privileged applications
- Input to applications is by means of a file
- Output is saved in a file for long-term storage

2

File System Properties

- Long-term existence
- Sharable between processes
- Structure

3

File Operations

- Create
- Delete
- Open
- Close
- Read
- Write

4

Terms Used with Files

- **Field**
 - Basic element of data
 - Contains a single value
 - Characterized by its length and data type
- **Record**
 - Collection of related fields
 - Treated as a unit
 - Example: employee record

5

Terms Used with Files

- **File**
 - Collection of similar records
 - Treated as a single entity
 - Have file names
 - May restrict access
- **Database**
 - Collection of related data
 - Relationships exist among elements

6

Typical Operations

- Retrieve_All
- Retrieve_One
- Retrieve_Next
- Retrieve_Previous
- Insert_One
- Delete_One
- Update_One
- Retrieve_Few

7

File Management Systems

- The way a user or application may access files
- Programmer does not need to develop file management software

8

Objectives for a File Management System

- Meet the data management needs and requirements of the user
- Guarantee that the data in the file are valid
- Optimize performance
- Provide I/O support for a variety of storage device types

9

Objectives for a File Management System

- Minimize or eliminate the potential for lost or destroyed data
- Provide a standardized set of I/O interface routines
- Provide I/O support for multiple users

10

Minimal Set of Requirements

- Each user should be able to *create, delete, read, write* and *modify* files
- Each user may have *controlled access* to other users' files
- Each user may *control* what type of accesses are allowed to the users' files
- Each user should be able to *restructure* the user's files in a form appropriate to the problem

11

Minimal Set of Requirements

- Each user should be able to *move data between files*
- Each user should be able to *back up* and *recover* the user's files in case of damage
- Each user should be able to *access* the user's files by using *symbolic names*

12

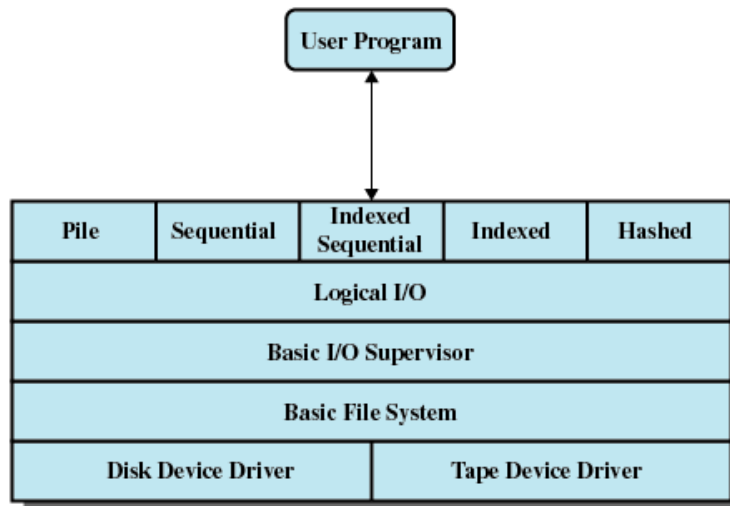
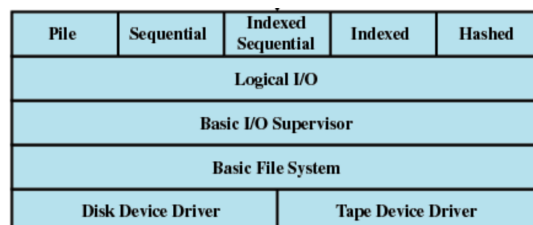


Figure 12.1 File System Software Architecture

Device Drivers

- Lowest level
- Communicates directly with peripheral devices
- Responsible for starting I/O operations on a device
- Processes the completion of an I/O request



Basic File System

- Physical I/O
- Deals with exchanging blocks of data
- Concerned with the placement of blocks
- Concerned with buffering blocks in main memory

File	Sequential	Indexed Sequential	Indexed	Hashed
Logical I/O				
Basic I/O Supervisor				
Basic File System				
Disk Device Driver			Tape Device Driver	

File	Sequential	Indexed Sequential	Indexed	Hashed
Logical I/O				
Basic I/O Supervisor				
Basic File System				
Disk Device Driver			Tape Device Driver	

Basic I/O Supervisor

- Responsible for file I/O initiation and termination
- Control structures are maintained
- Concerned with selection of the device on which file I/O is to be performed
- Concerned with scheduling access to optimize performance
- Part of the operating system

Logical I/O

- Enables users and applications to access records
- Provides general-purpose record I/O capability
- Maintains basic data about file

File	Sequential	Indexed Sequential	Indexed	Hashed
Logical I/O				
Basic I/O Supervisor				
Basic File System				
Disk Device Driver			Tape Device Driver	

Access Method

- Reflect different file structures
- Different ways to access and process data

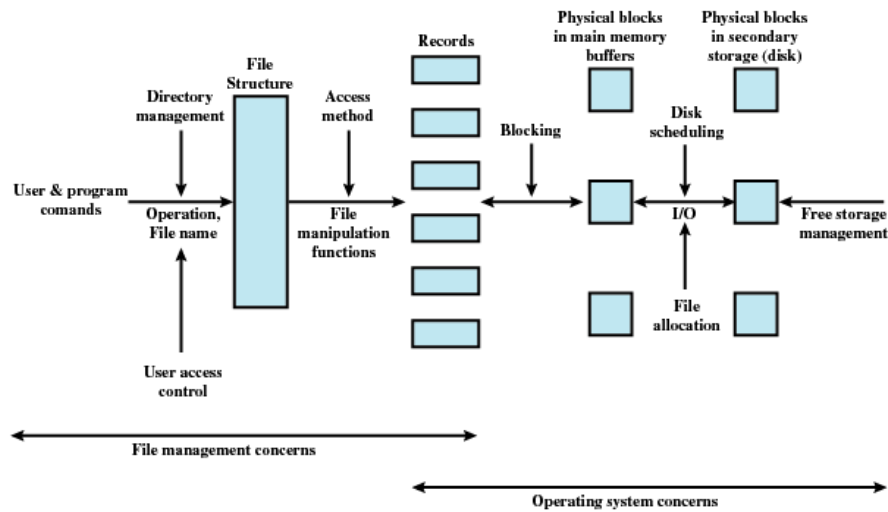


Figure 12.2 Elements of File Management

19

File Management Functions

- Identify and locate a selected file
- Use a directory to describe the location of all files plus their attributes
- On a shared system describe user access control
- Blocking for access to files
- Allocate files to free blocks
- Manage free storage for available blocks

20

Criteria for File Organization

- Short access time
 - Needed when accessing a single record
 - Not needed for batch mode
- Ease of update
 - File on CD-ROM will not be updated, so this is not a concern

21

Criteria for File Organization

- Economy of storage
 - Should be minimum redundancy in the data
 - Redundancy can be used to speed access such as an index
- Simple maintenance
- Reliability

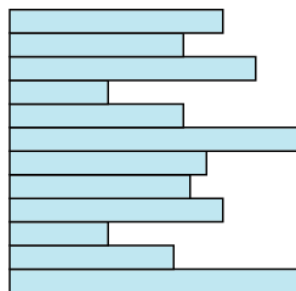
22

File Organization

- The Pile
 - Data are collected in the order they arrive
 - Purpose is to accumulate a mass of data and save it
 - Records may have different fields
 - No structure
 - Record access is by exhaustive search

23

Pile



Variable-length records
Variable set of fields
Chronological order

(a) Pile File

24

File Organization

- The Sequential File
 - Fixed format used for records
 - Records are the same length
 - All fields the same (order and length)
 - Field names and lengths are attributes of the file
 - One field is the key field
 - Uniquely identifies the record
 - Records are stored in key sequence

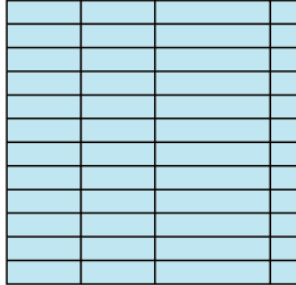
25

File Organization

- The Sequential File
 - New records are placed in a log file or transaction file
 - Batch update is performed to merge the log file with the master file
- Problems accessing records
 - need to “scan” through file
 - sequential storage proves limited
 - organize sequential file physically as linked list

26

Sequential File



Fixed-length records
Fixed set of fields in fixed order
Sequential order based on key field

(b) Sequential File

27

File Organization

- Indexed Sequential File
 - Maintain key characteristics of sequential file
 - records organized in sequence base on key field
 - Two new features are added
 - index to file to support random access
 - overflow file

28

File Organization

- Index
 - allows to quickly reach the vicinity of the desired record
 - Contains key field and a pointer to the main file
 - Index is searched to find highest key value that is equal to or precedes the desired key value
 - Search continues in the main file at the location indicated by the pointer

29

File Organization

- Comparison of sequential and indexed sequential
 - Example: a file contains 1 million records
 - sequential file: on average 500,000 accesses are required to find a record
 - indexed sequential:
 - if index contains 1000 entries, it will take on average 500 accesses to find the key
 - now the search continues in main file at the location indicated
 - on the average $500,000/1000 = 500$ accesses

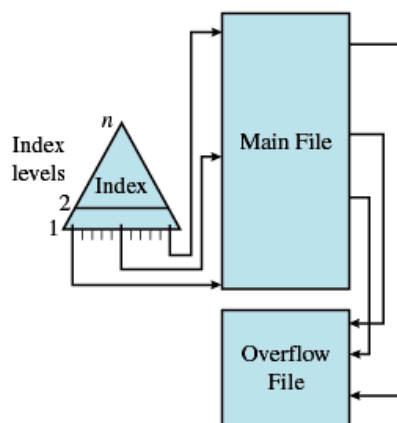
30

File Organization

- **Overflow**
 - New records are added to an overflow file
 - Record in main file that precedes it is updated to contain a pointer to the new record
 - The overflow is merged with the main file during a batch update
 - Multiple indexes for the same key field can be set up to increase efficiency

31

Indexed Sequential File



(c) Indexed Sequential File

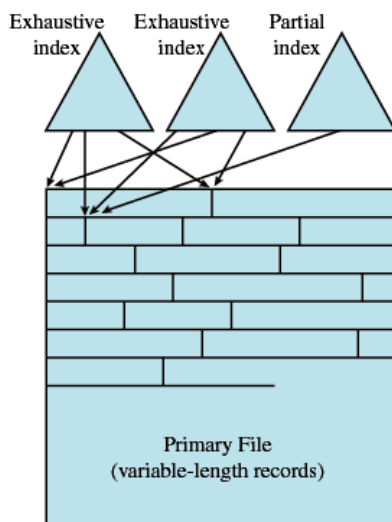
32

File Organization

- Indexed File
 - Indexed sequential is limited to the use of a single key (based on single field of the file)
 - Uses multiple indexes for different key fields
 - May contain an exhaustive index that contains one entry for every record in the main file
 - May contain a partial index

33

Indexed File



(d) Indexed File

34

File Organization

- The Direct or Hashed File
 - Directly access a block at a known address
 - Key field required for each record

35

Table 12.1 Grades of Performance for Five Basic File Organizations [WIED87]

File Method	Space		Update		Retrieval		
	Attributes		Record Size		Single record	Subset	Exhaustive
	Variable	Fixed	Equal	Greater			
Pile	A	B	A	E	E	D	B
Sequential	F	A	D	F	F	D	A
Indexed sequential	F	B	B	D	B	D	B
Indexed	B	C	C	C	A	B	D
Hashed	F	B	B	F	B	F	E

A = Excellent, well suited to this purpose $\approx O(r)$
 B = Good $\approx O(o \times r)$
 C = Adequate $\approx O(r \log n)$
 D = Requires some extra effort $\approx O(n)$
 E = Possible with extreme effort $\approx O(r \times n)$
 F = Not reasonable for this purpose $\approx O(r^2)$

where

r = size of the result
 o = number of records that overflow
 n = number of records in file

36