Comp 346

Tutorial 10

File Management

Files

- Data collections created by users
- The File System is one of the most important parts of the OS to a user
- Desirable properties of files:

Long-term existence

- files are stored on disk or other secondary storage and do not disappear when a user logs off

Sharable between processes

- files have names and can have associated access permissions that permit controlled sharing

Structure

- files can be organized into hierarchical or more complex structure to reflect the relationships among files

File Systems

- Provide a means to store data organized as files as well as a collection of functions that can be performed on files
- Maintain a set of attributes associated with the file
- Typical operations include:
    - Create
    - Delete
    - Open
    - Close
    - Read
    - Write

File Structure

Four terms are commonly used when discussing files:

- Field
- Record
- File
- Databas
  e

File Structure

- Files can be structured as a collection of records or as a sequence of bytes
- UNIX, Linux, Windows, Mac OS's consider files as a sequence of bytes
- Other OS's, notably many IBM mainframes, adopt the collection-of-records approach; useful for DB
- COBOL supports the collection-of-records file and can implement it even on systems that don't provide such files natively.

The File Manager

- File Manager is also called the **File management system** and is the software responsible for:
  - Creating, deleting, modifying, controlling access to files

- Provides support for libraries of programs to online users, for spooling operation, and for interactive computing

Université Concordia University

Responsibilities of the File Manager

- **Four tasks**
  - Keep track of where each file is stored
  - Implement a policy that will:
    - Determine where and how files are stored
    - Efficiently use available storage space
    - Provide efficient file access
  - Allocate each file when a user has been cleared for access to it, then record its use
  - File deallocation
    - File returned to storage
    - Communicate file availability to others waiting for it

Responsibilities of the File Manager (continued)

- **File Manager's Policy** determines:
  - File storage location
  - System and user access files via device-independent commands
- Who will have access to which file. This depends on two factors:
- Flexibility of access to information (Factor 1)
  - Shared files
  - Providing distributed access
  - Allowing users to browse public directories

Responsibilities of the File Manager (continued)

- Subsequent protection (Factor 2)
  - Protect files against system malfunctions
  - Security checks
    - Account numbers, passwords, lockwords
- **File allocation**
  - Activate secondary storage device, load file into memory, update records
- **File deallocation**
  - Update file tables, rewrite file (if revised), notify waiting processes of file availability

Definitions

- **Field**
  - Group of related bytes (e.g. last_name, first_name, age) that can be identified by the user with a name, type, and size.
- **Record**
  - Group of related fields
- **File**
  - Group of related records that contains information to be used by specific application programs to generate reports. This type of file contains data and is sometimes called a flat file because it has no connections to other files and has no dimensionality.

Definitions (continued)

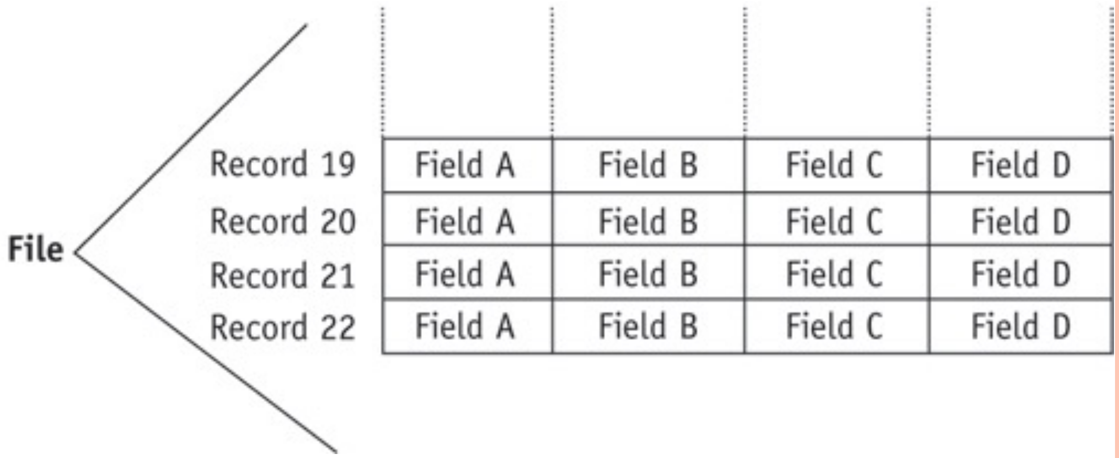
- **Databases**
  - Groups of related files
  - Interconnected at various levels
    - Give users flexibility of access to stored data
- **Program files**
  - Contain instructions
- **Data files**
  - Contain data
- **Directories**
  - Are special files with listings of filenames and their attributes

Université Concordia University

DEFINITIONS (CONTINUED)

(figure 8.1)

Files are made up of records. Records consist of fields.



Record 19	Field A	Field B	Field C	Field D
Record 20	Field A	Field B	Field C	Field D
Record 21	Field A	Field B	Field C	Field D
Record 22	Field A	Field B	Field C	Field D

File Management System Objectives

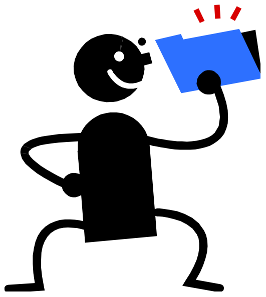
- Meet the data management needs of the user
- Guarantee that the data in the file is valid
- Optimize performance
- Provide I/O support for a variety of storage device types
- Minimize the potential for lost or destroyed data
- Provide a standardized set of I/O interface routines to user processes
- Provide I/O support for multiple users in the case of multiple-user systems

Minimal User Requirements

- Each user:

1. should be able to create, delete, read, write and modify files
2. may have controlled access to other users' files
3. may control what type of accesses are allowed to the files
4. should be able to restructure the files in a form appropriate to the problem
5. should be able to move data between files
6. should be able to back up and recover files in case of damage
7. should be able to access his or her files by name rather than by numeric identifier

File Sharing



Two issues arise when allowing files to be shared among a number of users:

- access rights
- management of simultaneous access

Access Rights

- ***None***
  - the user would not be allowed to read the user directory that includes the file
- ***Knowledge***
  - the user can determine that the file exists and who its owner is and can then petition the owner for additional access rights
- ***Execution***
  - the user can load and execute a program but cannot copy it
- ***Reading***
  - the user can read the file for any purpose, including copying and execution
- ***Appending***
  - the user can add data to the file but cannot modify or delete any of the file's contents
- ***Updating***
  - the user can modify, delete, and add to the file's data
- ***Changing protection***
  - the user can change the access rights granted to other users
- ***Deletion***
  - the user can delete the file from the file system

User Access Rights

Owner
- usually the initial creator of the file
- has full rights
- may grant rights to others

Specific Users
- individual users who are designated by user ID

User Groups
- a set of users who are not individually defined

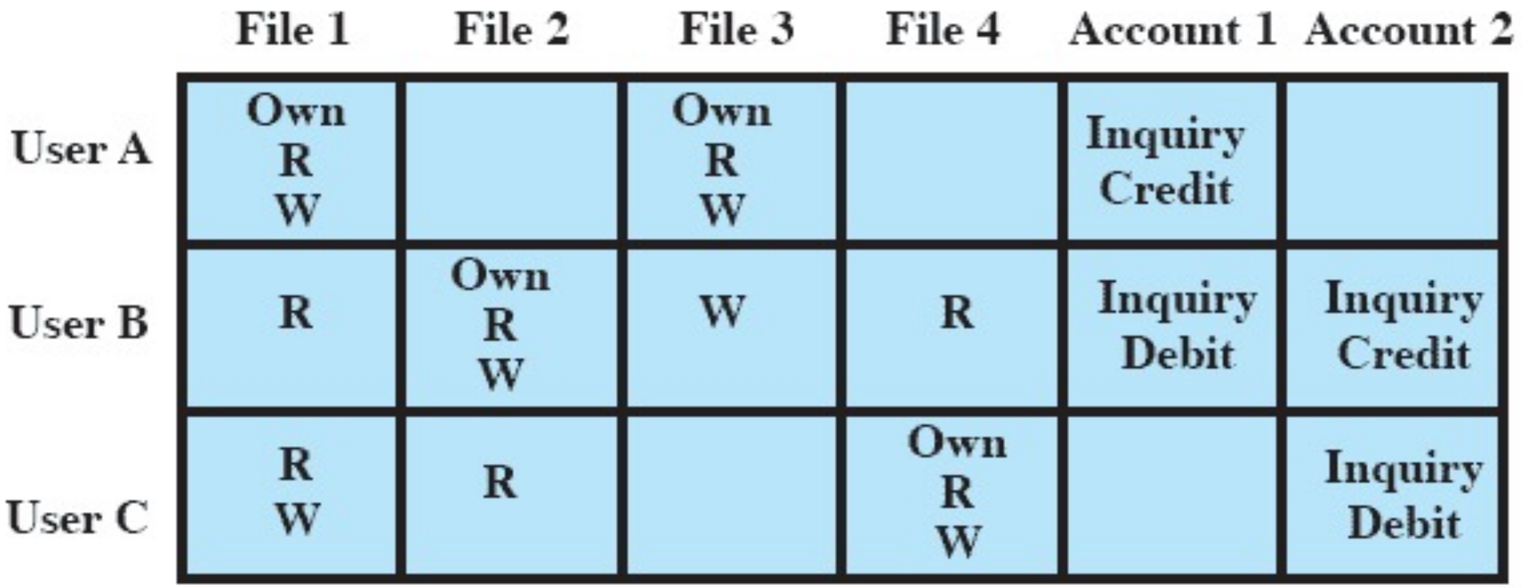
All
- all users who have access to this system
- these are public files

Access Control

- In a system with multiple users, it's important to protect one user's objects (files, directories) from other users.
- Two levels of protections:
  - Logon verifications: guarantees you have the right to log onto the system
  - Access determination: guarantees you have permission to access a specific object
- Access matrix, access lists, capability lists: techniques for determining access rights.

Access Matrix

- The basic elements are:
  - **subject** – an entity capable of accessing objects
  - **object** – anything to which access is controlled
  - **access right** – the way in which an object is accessed by a subject



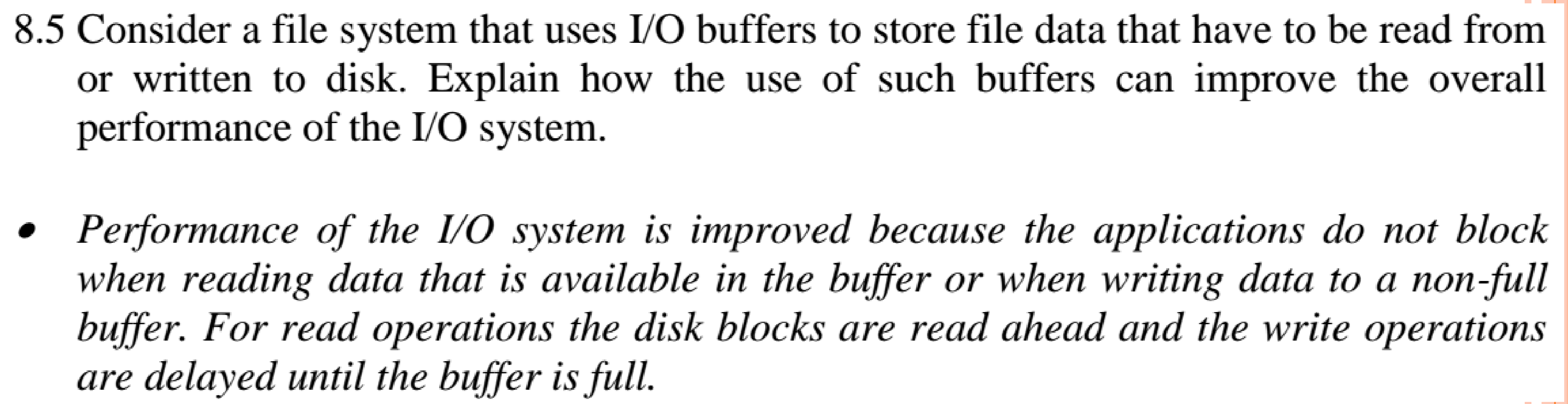
	File 1	File 2	File 3	File 4	Account 1	Account 2
User A	Own R W		Own R W		Inquiry Credit	
User B	R	Own R W	W	R	Inquiry Debit	Inquiry Credit
User C	R W	R		Own R W		Inquiry Debit

(a) Access matrix

PROBLEM 1

8.5 Consider a file system that uses I/O buffers to store file data that have to be read from or written to disk. Explain how the use of such buffers can improve the overall performance of the I/O system.

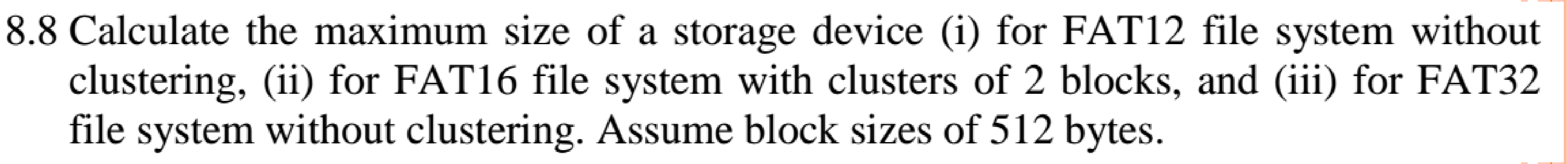
SOLUTION 1



8.5 Consider a file system that uses I/O buffers to store file data that have to be read from or written to disk. Explain how the use of such buffers can improve the overall performance of the I/O system.

- *Performance of the I/O system is improved because the applications do not block when reading data that is available in the buffer or when writing data to a non-full buffer. For read operations the disk blocks are read ahead and the write operations are delayed until the buffer is full.*

Problem 2



8.8 Calculate the maximum size of a storage device (i) for FAT12 file system without clustering, (ii) for FAT16 file system with clusters of 2 blocks, and (iii) for FAT32 file system without clustering. Assume block sizes of 512 bytes.

SOLUTION 2

8.8 Calculate the maximum size of a storage device (i) for FAT12 file system without clustering, (ii) for FAT16 file system with clusters of 2 blocks, and (iii) for FAT32 file system without clustering. Assume block sizes of 512 bytes.

(i) $2^{12} \times 512 = 4K \times 512 = 2\ Megabytes.$

(ii) $2^{16} \times (2 \times 512) = 64K \times 1K = 64\ Megabytes.$

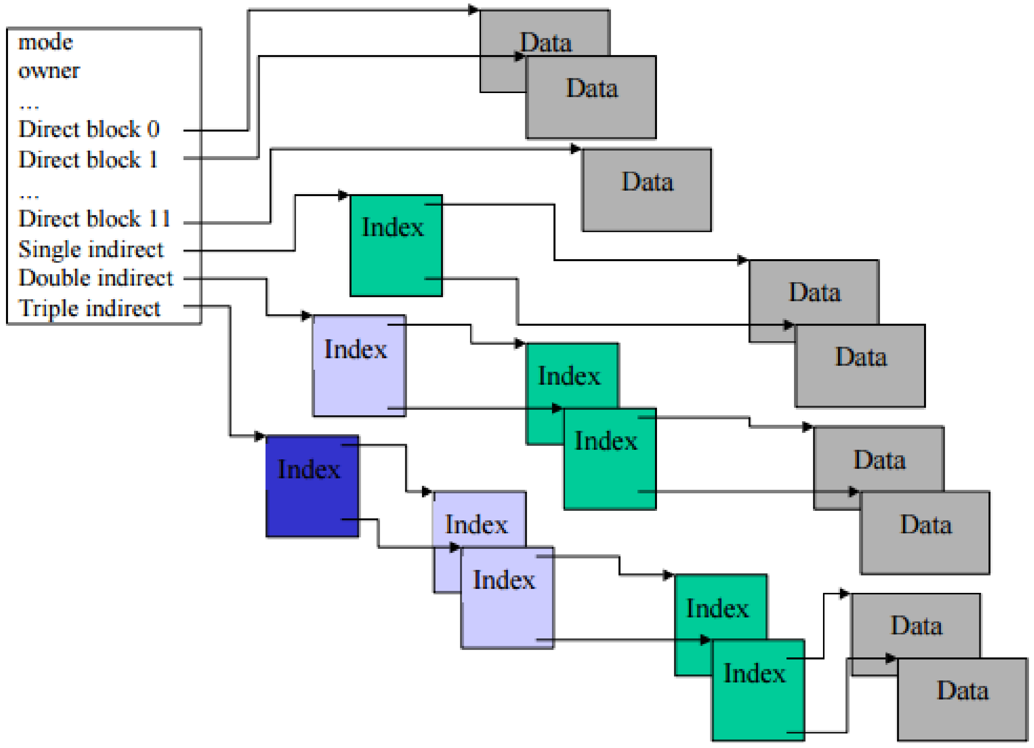
(iii) $2^{32} \times 512 = 4G \times 512 = 2\ Terabytes.$

UNIVERSITÉ Concordia UNIVERSITY

Problem 3

8.10 Consider a Unix file system having a block size of 2 kilobytes and a block pointer size of 32 bits. Calculate the maximum size of a file.

Reminder: Unix File System Structure



- Disk block management

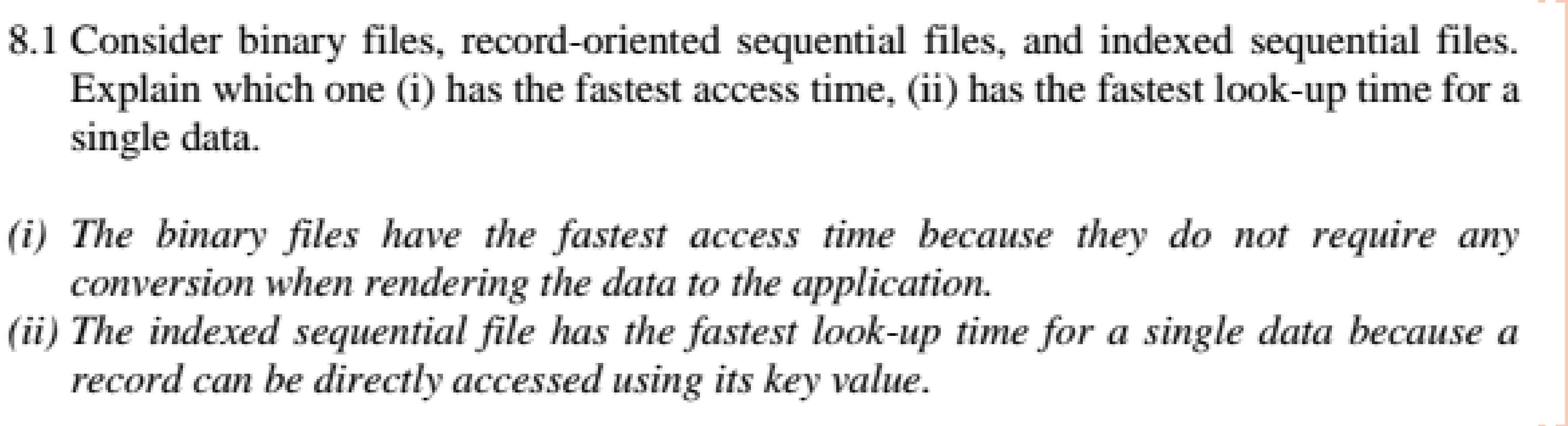
SOLUTION 3

- $Nb\ pointer\ per\ index\ bloc = floor 2048/4 = 2048$ (2^11)/4(bytes)=512
- S0 = Size of only direct pointer $=12*2K=24K$
- $S1 = Size\ of\ only\ single\ indirect\ pointer = 512*2K$
- $S2 = Size\ of\ only\ double\ indirect\ pointer = 512*2*2K$
- $S3 = Size\ of\ only\ triple\ indirect\ pointer = 512*3*2K$
- $Maximum\ file\ size\ (full\ inode) = S0+S1+S2+S3$

Problem 4

- Consider binary files, record-oriented sequential files, and indexed sequential files. Explain which one (i) has the fastest access time, (ii) has the fastest look-up time for a single data.

SOLUTION 4



8.1 Consider binary files, record-oriented sequential files, and indexed sequential files. Explain which one (i) has the fastest access time, (ii) has the fastest look-up time for a single data.

(i) The binary files have the fastest access time because they do not require any conversion when rendering the data to the application.

(ii) The indexed sequential file has the fastest look-up time for a single data because a record can be directly accessed using its key value.

Problem 5

- Consider a file system having disk block sizes of 512 bytes and a file having a size of 3452 bytes and a record size of 78 bytes.
- (i) Calculate the number of records that the file requires.
- (ii) Calculate the disk block factor for the file.
- (iii)Calculate the amount of internal fragmentation in the last disk block.
- (iv) Find out the disk block number (relatively to zero) that stores the 24th record in the file.

Université Concordia University

Solution 5

8.2 Consider a file system having disk block sizes of 512 bytes and a file having a size of 3452 bytes and a record size of 78 bytes.

(i) Calculate the number of records that the file requires.

- *Number of records* $= \lceil 3452/78 \rceil = \lceil 44.25 \rceil = 45$

(ii) Calculate the disk block factor for the file.

- $B_f = \lfloor 512/78 \rfloor = \lfloor 6.56 \rfloor = 6$

(iii) Calculate the amount of internal fragmentation in the last disk block.

- $f_i = 512\%78 = 44$

(iv) Find out the disk block number (relatively to zero) that stores the 24th record in the file.

- $i = \lfloor (24-1)/6 \rfloor = 3$

Problem 6

- For the contiguous, linked, and indexed block allocation techniques explain which one (i) provides the fastest transfer time, (ii) permits an easier file extension, and (iii) provides the fastest access time to individual blocks.

Solution 6

8.3 For the contiguous, linked, and indexed block allocation techniques explain which one (i) provides the fastest transfer time, (ii) permits an easier file extension, and (iii) provides the fastest access time to individual blocks.

(i) The contiguous block allocation technique provides the fastest transfer time because the blocks are in the same vicinity, which minimizes the number of disk head displacements.
(ii) The linked block allocation technique permits an easier file extension because a new block can be allocated anywhere on the disk.
(iii)The indexed block allocation technique provides the fastest access time because a specific disk block can be directly accessed through an index lookup.

Refrences

- Operating Systems: Internals and Design Principles, Chapter 12 File Management, Seventh Edition By William Stallings
- Understanding Operating Systems Fifth Edition, Chapter 8 File Management
- Problem Solving questions by François Gingras Myriam Kharma, Prof. Titus Kerly

UNIVERSITÉ Concordia UNIVERSITY