File Management

Chapter 12

Files

- Used for input to a program
- Output of application saved in a file for long-term storage

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

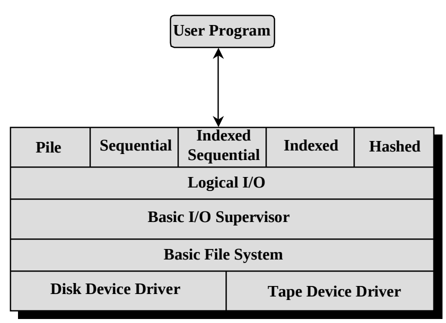
Terms Used with Files

- **File**
  - collection of similar records
  - treated as a single entity
  - have unique file names
  - may restrict access
- **Database**
  - collection of related data
  - relationships exist among elements

File Management System

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

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

Basic I/O Supervisor

- Responsible for file I/O initiation and termination
- Control structures are maintained
- Concerned with scheduling access to optimize performance
- Part of the operating system

Logical I/O

- Allows users and applications to access records
- Maintains basic data about file

Access Method

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

Functions of File Management

- 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

File Directories

- **Contains information about files**
  - attributes
  - location
  - ownership
- **Directory itself is a file owned by the operating system**
- **Provides mapping between file names and the files themselves**

Simple Structure for a Directory

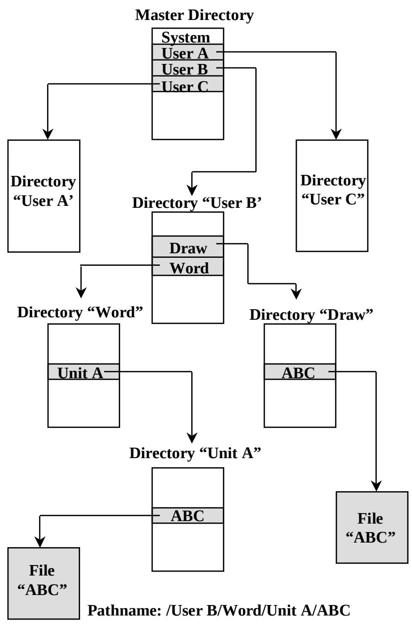
- **List of entries, one for each file**
- **Sequential file with the name of the file serving as the key**
- **Provides no help in organizing the files**
- **Forces user to be careful not to use the same name for two different files**

Two-level Scheme for a Directory

- **One directory for each user and a master directory**
- **Master directory contains entry for each user**
  - provides address and access control information
- **Each user directory is a simple list of files for that user**
- **Still provides no help in structuring collections of files**

Hierarchical, or Tree-Structured Directory

- **Master directory with user directories underneath it**
- **Each user directory may have subdirectories and files as entries**



Hierarchical, or Tree-Structured Directory

- **Files can be located by following a path from the root, or master, directory down various branches**
  - this is the pathname for the file
- **Can have several files with the same file name as long as they have unique path names**

Hierarchical, or Tree-Structured Directory

- Current directory is the working directory
- Files are referenced relative to the working directory

File Sharing

- Way to control access to a particular file
- Users or groups of users are granted certain access rights to a file

Access Rights

- **None**
  - user may not know of the existence of the file
  - user is not allowed to read the user directory that includes the file
- **Knowledge**
  - user can only determine that the file exists and who its owner is

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

Access Rights

- **Updating**
  - the user can modify, deleted, and add to the file's data. This includes creating the file, rewriting it, and removing all or part of the data
  - Changing protection
  - user can change access rights granted to other users
- **Deletion**
  - user can delete the file

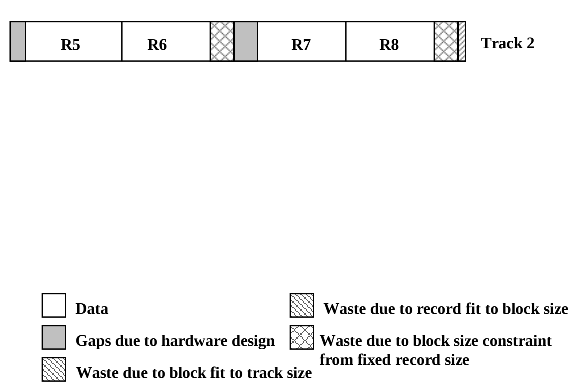
Access Rights

- **Owners**
  - has all rights previously listed
  - may grant rights to others using the following classes of users
    - specific user
    - user groups
    - all for public files

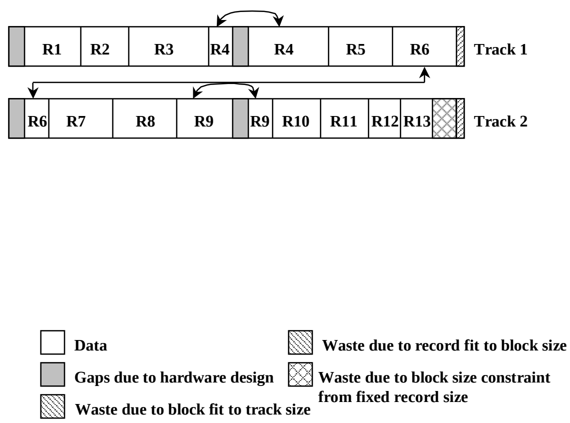
Simultaneous Access

- User may lock entire file when it is to be updated
- User may lock the individual records during the update
- Mutual exclusion and deadlock are issues for shared access

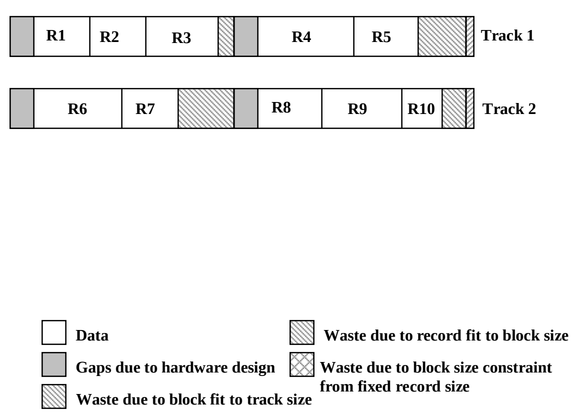
Record Blocking Methods - Fixed Blocking



Record Blocking Methods - Variable Blocking:Spanned



Record Blocking Methods - Variable Blocking:UnSpanned



Secondary Storage Management

- Space must be allocated to files
- Must keep track of the space available for allocation
- Space is allocated as one or more contiguous units or portions

Preallocation

- Need the maximum size for the file at the time of creation
- Difficult to reliably estimate the maximum potential size of the file
- Tend to overestimated file size so as not to run out of space

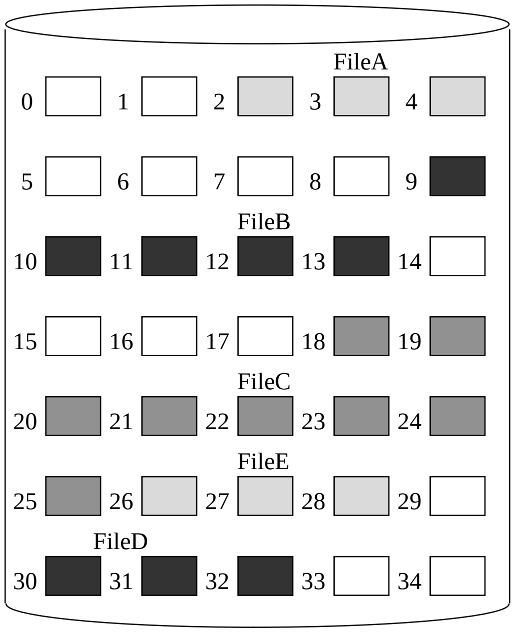
Portion Size

- **Contiguity of space increases performance**
- **Large number of small portions increases the size of tables needed**
- **Fixed-size simplifies the reallocation of space**
- **Variable-size minimizes waste of unused storage**

Methods of File Allocation

- **Contiguous allocation**
  - single set of blocks is allocated to a file at the time of creation
  - only a single entry in the file allocation table
    - starting block and length of the file
- **Fragmentation will occur**
- **Will become difficult to find contiguous blocks of sufficient length**

Contiguous File Allocation



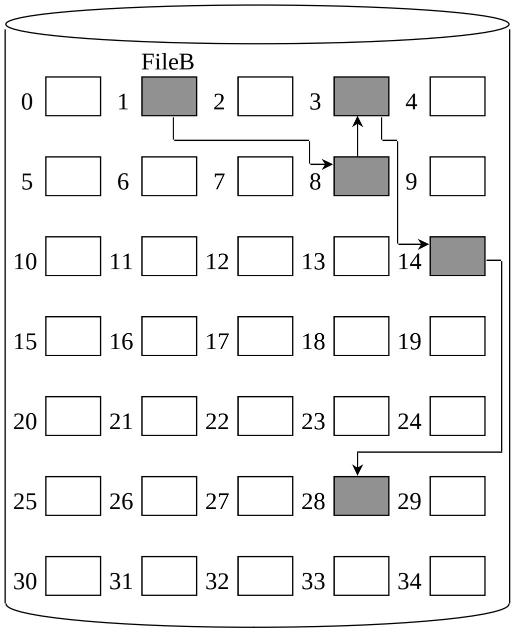
File Allocation Table

File Name	Start Block	Length
FileA	2	3
FileB	9	5
FileC	18	8
FileD	30	2
FileE	26	3

Methods of File Allocation

- **Chained allocation**
  - allocation on basis of individual block
  - each block contains a pointer to the next block in the chain
  - only single entry in the file allocation table
    - starting block and length of file
- **No fragmentation**
- **Any free block can be added to the chain**
- **No accommodation of the principle of locality**

Chained File Allocation



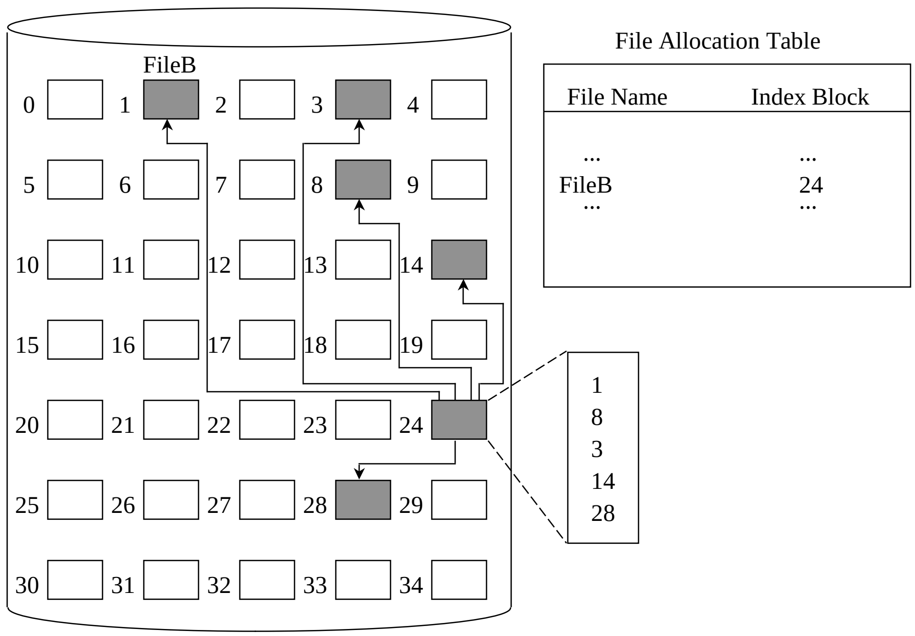
File Allocation Table

File Name	Start Block	Length
...	...	...
FileB	1	5
...	...	...

Methods of File Allocation

- **Indexed allocation**
  - file allocation table contains a separate one-level index for each file
  - the index has one entry for each portion allocated to the file
  - the file allocation table contains block number for the index

Indexed Allocation with Block Portions



Indexed Allocation - Var Length Portions