



OPERATING SYSTEMS

File Management

Introduction

File organization and Access

File Directories

File Sharing

Record Blocking

Secondary Storage Management

File System Security



Introduction

- Files are one of the most important entity b/w OS and user
- File Management System is a utility of OS that is responsible to provide several operations related with file.
- Mainly concerned with secondary storage devices
- And it allows users/applications to:
 - Create
 - Open
 - Write/Append
 - Read
 - Close
 - Delete
 - Control Access to other user's file
 - Able to restructure files
 - Move data b/w files
 - Backup and recover files
 - Access with certain symbolic identification



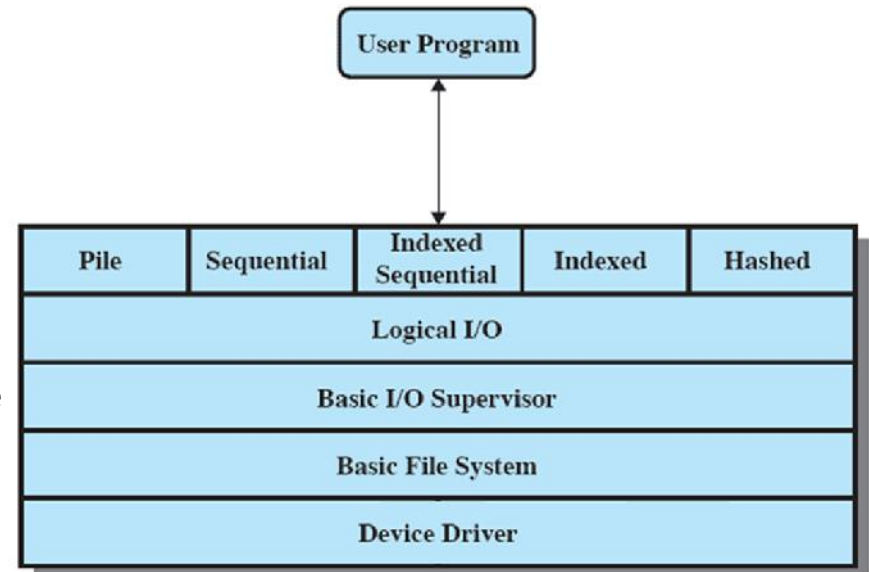
Objectives

- Users need of file management
- Data validation of file
- Performance optimization
- I/O support for various storage devices
- Minimize data loss
- Standard I/O interface functions to user programs
- I/O support for multiple users



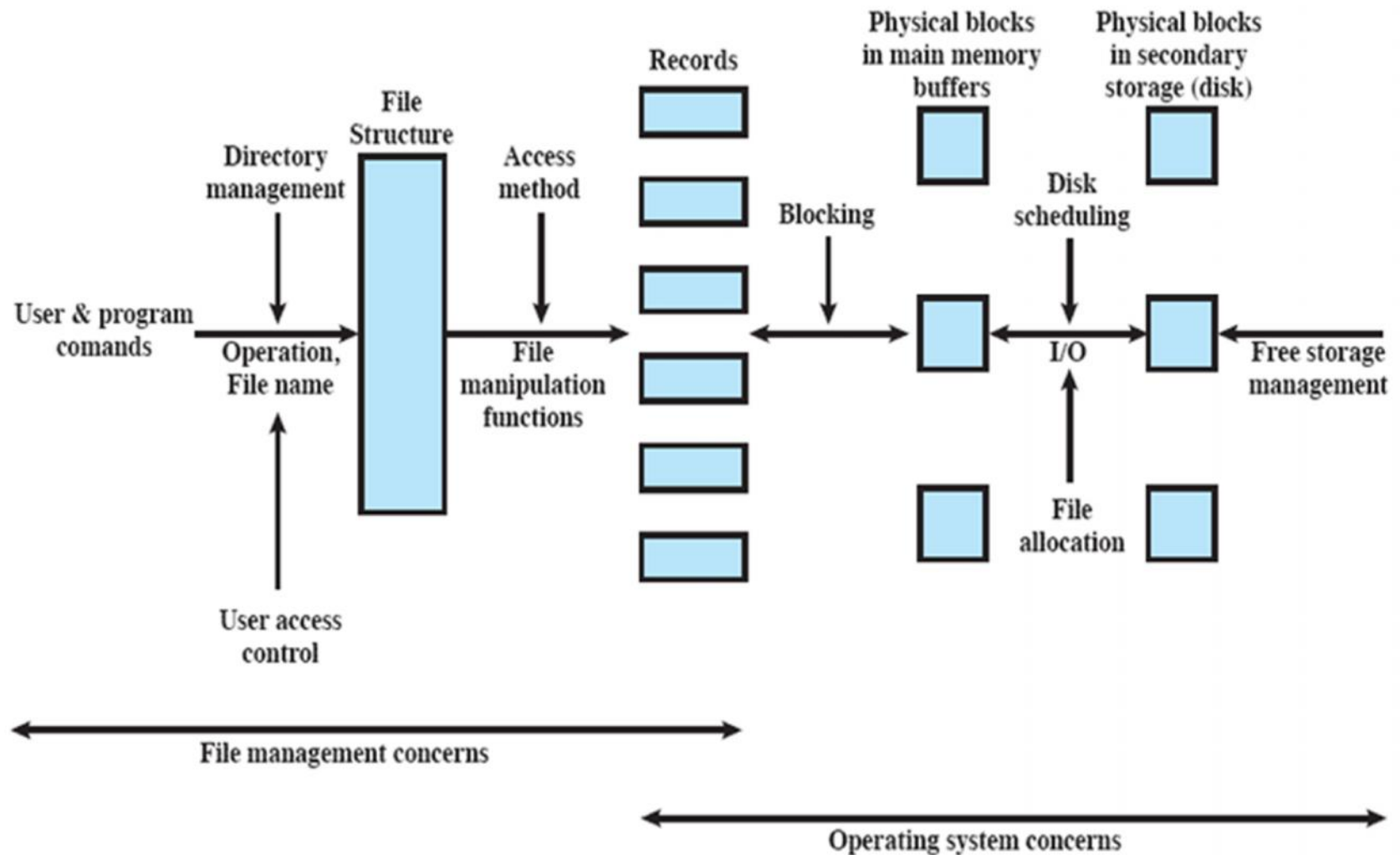
File Management System Architecture

- Device Driver communicates with device and is responsible to process I/O operation
- Basic File System or Physical I/O deals with exchange of file data (in blocks) in main memory and device
- Basic I/O Supervisor deals with file I/O initialization and termination, it deals with I/O Devices, I/O instruction Scheduling and file status
- Logic I/O provides access to users/applications over records/files
- File Organization and Access Methods represents file system structure and provides interface b/w applications and devices that hold the data





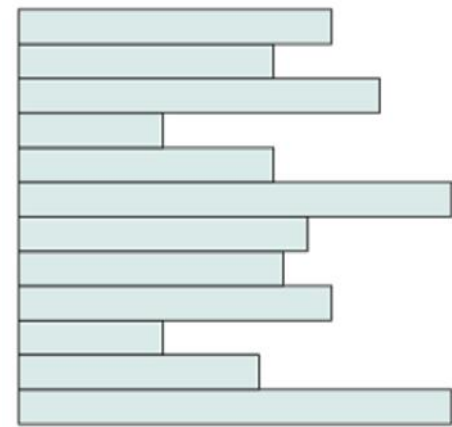
File Management Elements





File Organization & Access methods

- Technique to access and organize file logically
- Criteria
 - Short access time
 - Ease of change/update
 - Optimized storage
 - Reliable
- Techniques:
 1. Pile
 - Variable Length of records
 - Variable set of fields
 - Record is accessed by complete file search
 - Easy
 - Not optimal performance

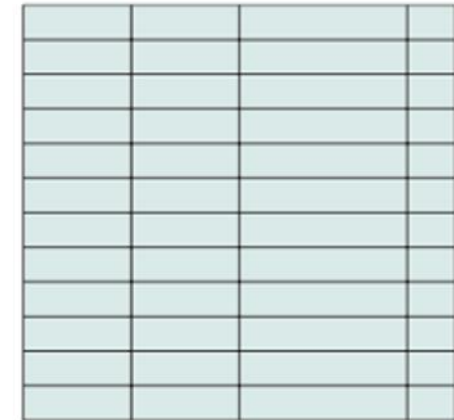




File Organization & Access methods

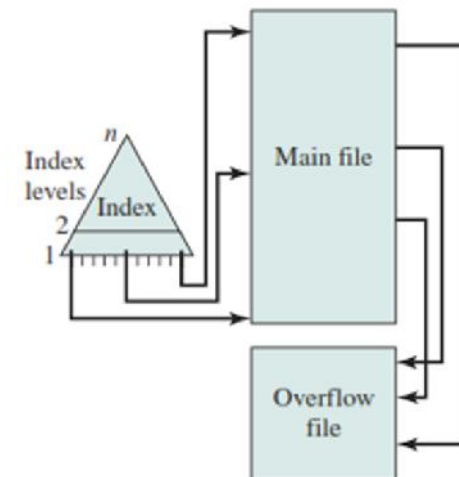
2. Sequential File

- Fixed Length of records
- Key Field is used to identify a record
- Records are stored and searched in a sequence
- Not optimal for multi-user systems and/or large files
- Easy to map with blocks
- Easy to delete a record
- Insertion of new record is done in separate file (pile) called log or transaction file to adopt sequence
- System periodically merges main file & transaction file



3. Indexed Sequential File

- Work as sequential file
- Uses indexes to locate particular record that contains key field and pointer to record
- Record can be accessed randomly
- Searching doesn't require to load several blocks
- Addition is done in overflow file and is merged with main file and index is updated periodically





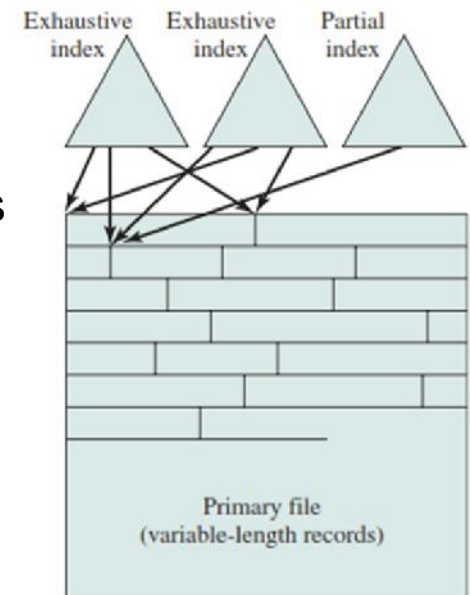
File Organization & Access methods

4. Indexed File

- Variable length records without sequence are stored
- Exhaustive index (sequential) is created that contains key attribute & point to respective record
- Partial index (sequential) is created for other attributes of interest.
- Fast search with several attributes
- No sequential or transaction file overhead

5. Direct or Hash File

- Fixed length of records
- Each record has key field
- Complete block is accessed at a time
- Hash file contains the information of key field and pointer their relative block
- Transaction file is used for new records
- which is merged and hash file is updated periodically
- Efficient when fast access to a record is required





File Directories

- Directory itself is a file of Operating System
- Used to map file names toward files
- Accessed by several file management routines
- Operations over directories:
 - Search
 - Create
 - Delete
 - List
 - Update



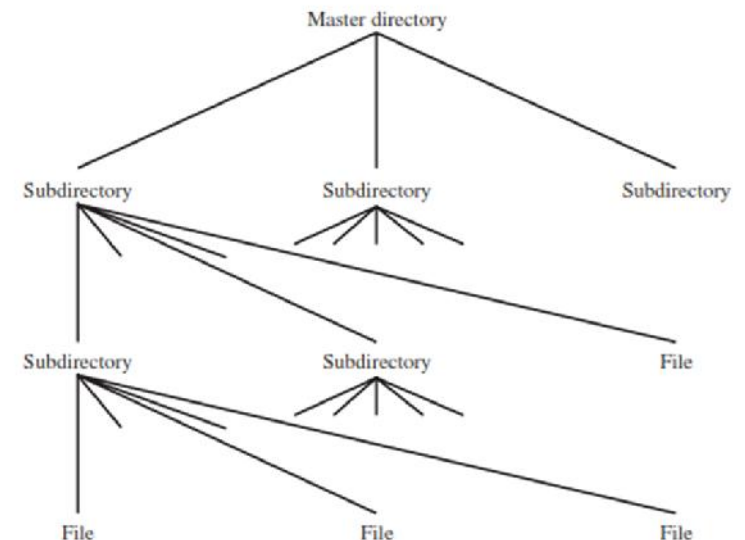
Information Elements of File Directory

Basic Information	
File Name	Name as chosen by creator (user or program). Must be unique within a specific directory
File Type	For example: text, binary, load module, etc.
File Organization	For systems that support different organizations
Address Information	
Volume	Indicates device on which file is stored
Starting Address	Starting physical address on secondary storage (e.g., cylinder, track, and block number on disk)
Size Used	Current size of the file in bytes, words, or blocks
Size Allocated	The maximum size of the file
Access Control Information	
Owner	User who is assigned control of this file. The owner may be able to grant/deny access to other users and to change these privileges.
Access Information	A simple version of this element would include the user's name and password for each authorized user.
Permitted Actions	Controls reading, writing, executing, and transmitting over a network
Usage Information	
Date Created	When file was first placed in directory
Identity of Creator	Usually but not necessarily the current owner
Date Last Read Access	Date of the last time a record was read
Identity of Last Reader	User who did the reading
Date Last Modified	Date of the last update, insertion, or deletion
Identity of Last Modifier	User who did the modifying
Date of Last Backup	Date of the last time the file was backed up on another storage medium
Current Usage	Information about current activity on the file, such as process or processes that have the file open, whether it is locked by a process, and whether the file has been updated in main memory but not yet on disk



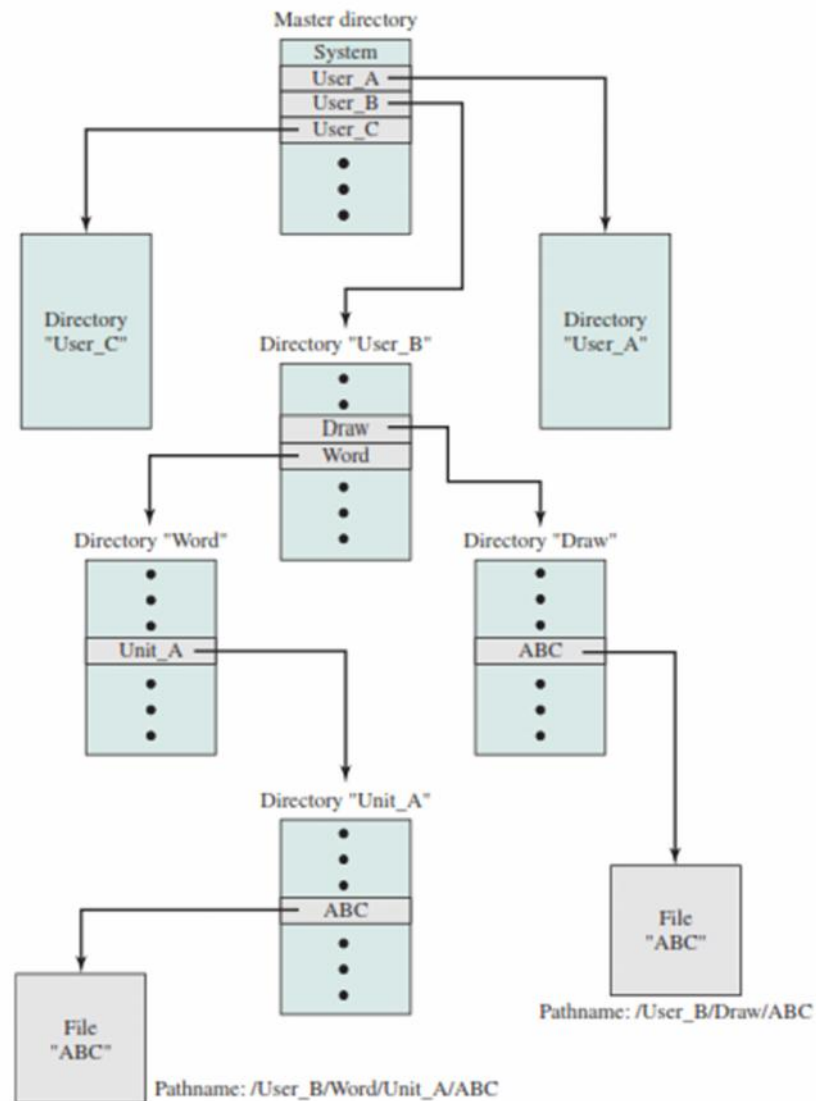
Structure of File Directory

- Simple List Structure
 - List the entries for each file
 - Sequential file
 - Forces to use unique file names
 - Not suitable for required operations
- Multi-Level Structure
 - Use Hierarchical Structure
 - A Master List contains information about subdirectory list
 - For huge number of entries, list can be created using hash file technique





Hierarchical or Tree Structured Directory





File Sharing

- File Sharing among multiple users
- Issues
 - Access Rights
 - None
 - Knowledge
 - Execution
 - Reading
 - Appending
 - Updating
 - Deletion
 - Changing Protection
 - Access rights can be defined for User Classes
 - Owner
 - Specific User
 - User Group
 - All
 - Management of simultaneous access
 - Mutual Exclusion is solution
 - File or particular record might be locked during change operation



Record Blocking

- Records are the logical unit of access of a structured file
- While, blocks are the unit for I/O with secondary storage
- Records must be organized as blocks
- Issues
 - Fixed or variable block size
 - Variable block size provides flexibility
 - Fixed block size provides easiness in organizing blocks and managing I/O and buffers
 - Larger or smaller block size
 - Larger block size is beneficial for sequential access, but larger buffer is required and is difficult to manage
 - Smaller block size increases number of I/O operations, efficient for random record access
 - Requires multiple paging technique to map blocks in small size pages



Record Blocking

- Techniques:

1. Fixed blocking

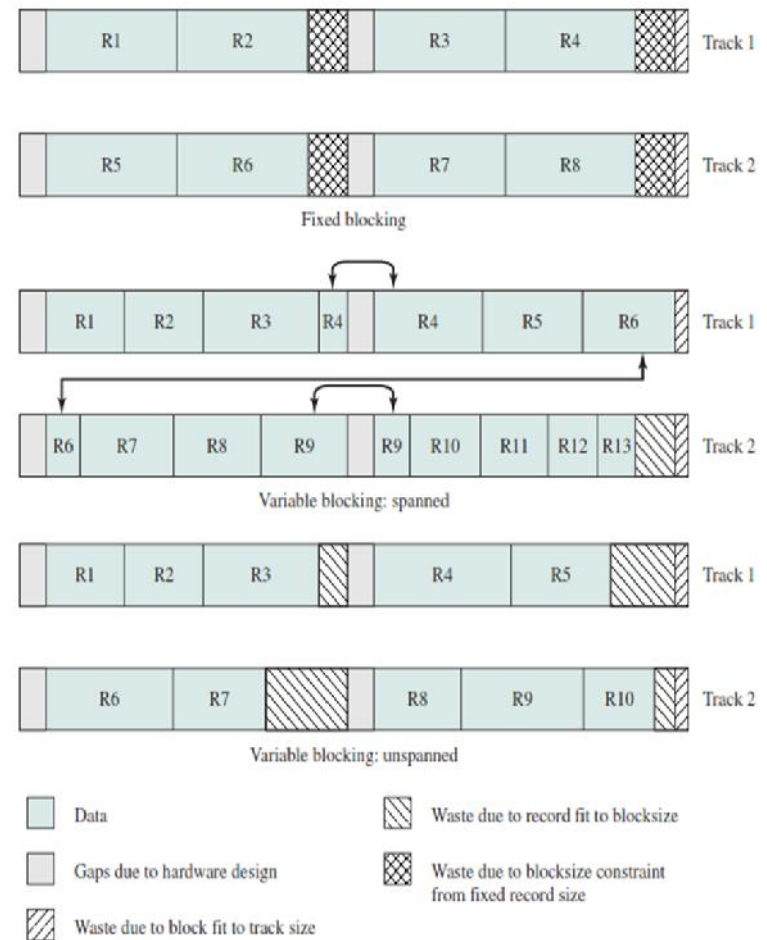
- Fixed length records are used
- Unused space at end of each block (internal fragmentation)

2. Variable length spanned blocking

- Variable length records
- Records are packed into blocks with no unused space
- Records are divided into two blocks
- Pointed by a pointer
- Difficult to perform update operation

3. Variable length unspanned blocking

- Variable length records without spanning are packed into blocks
- Unused space at end of block





Secondary Storage Management

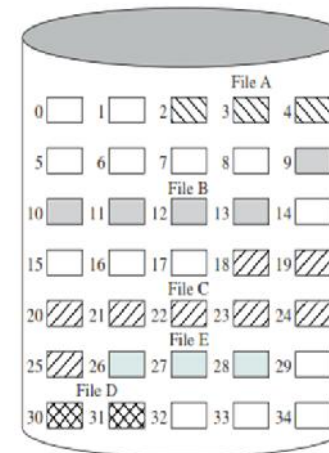
- Operating System is responsible for allocating blocks
- Issues
 - File Allocation towards secondary storage device
 - Free space management
- File Allocation
 - Size allocated when file created
 - Portion should be allocated (variable of fixed)
 - Technique
- Per-allocation Allocation technique in which max. size is allocated according to its requirement at creation
- Dynamic Allocation assign portions to file as needed



File Allocation Techniques

- **Contiguous**

- Blocks are arranged in contiguous manner
- Pre-allocation method
- Variable size portion
- Fast access to file/blocks
- Creates fragmentation which requires compaction

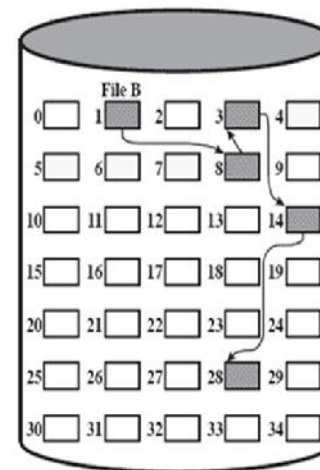


File allocation table

File name	Start block	Length
File A	2	3
File B	9	5
File C	18	4
File D	30	2
File E	26	3

- **Chained**

- Blocks are assigned in individual blocks
- Each block have pointer to next block
- Can be used for pre-allocation & dynamic
- No external Fragmentation
- Needs consolidation for faster access



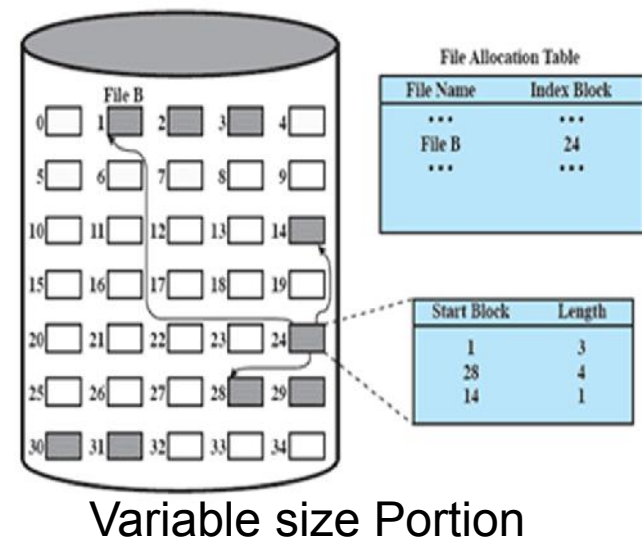
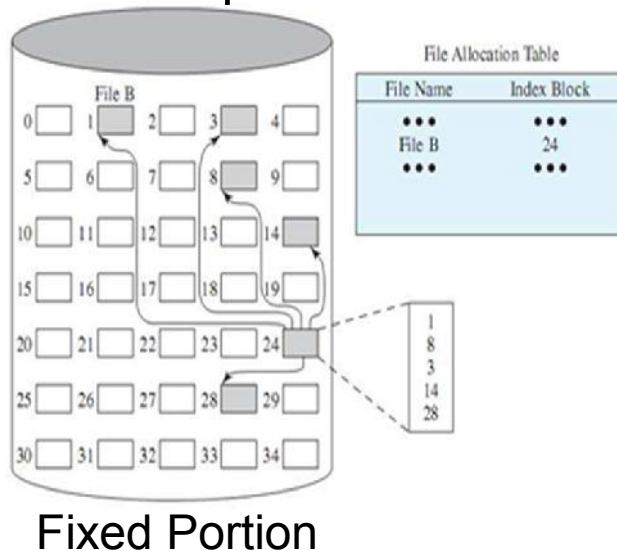
File Allocation Table

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



File Allocation Techniques

- Indexed
 - Can be used with variable size or fixed size portions
 - A block is created for each file with contains required information of file
 - File Allocation Table have entry of that respective block
 - Variable size portion provides fast access but creates fragmentation
 - Both requires consolidation





Free Space Management

- Unallocated space must be managed
- Trace the empty blocks
- Disk allocation table in addition to a file allocation table is used
- Techniques:
 1. Bit Table
 - A vector containing one bit for each block on the disk
 - Entry of free block is represented by 0 and used block by 1
 - Simple and easy
 - Feasible with all file allocation methods
 - Memory of table increases if number of blocks are increased



Free Space Management

2. Chained Free Portions

- Free portions/blocks can be chained together
- Each node contains pointer to next portion and length
- Negligible space overhead, only need to point starting node
- Feasible with all file allocation methods
- Leads to fragmentation after long time usage
- Time consuming

3. Indexing

- All free blocks are treated as a file
- Any File allocation method can be used for its management
- Single entry for all free blocks
- Feasible with all file allocation methods
- Variable size portions can be used for more efficiency



Free Space Management

4. Free Block List

- Each block is assigned a number sequentially
- List becomes too large as number of blocks are increased
- List is stored on secondary storage device and not loaded into main memory
- Free block list is accessed in FIFO or LIFO structure



Questions

