CSC 4103 - Operating Systems
Spring 2007

LECTURE - XV

# FILE SYSTEMS - I

Tevfik Koşar

Louisiana State University

March 20th, 2007

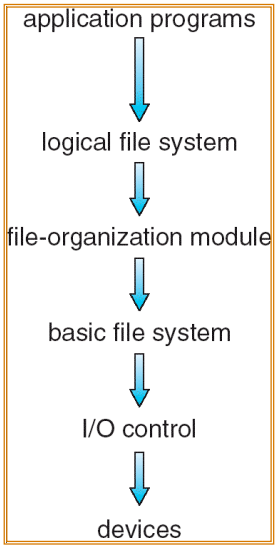
## File-System Structure

- Provides organized and efficient access to data on secondary storage, E.g.:
  - Organizing data into files and directories
  - Improve I/O efficiency between disk and memory (perform I/O in units of blocks rather than bytes)
  - Contains file structure via a File Control Block (FCB)
    - Ownership, permissions, location..

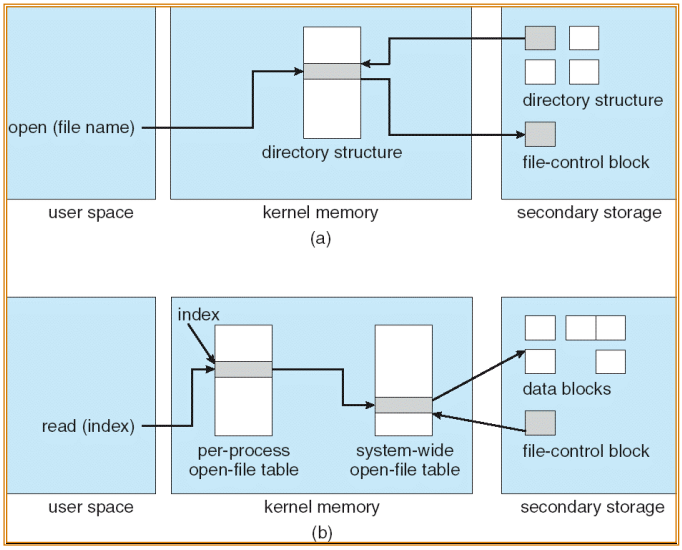
## A Typical File Control Block

| file permissions |
| --- |
| file dates (create, access, write) |
| file owner, group, ACL |
| file size |
| file data blocks or pointers to file data blocks |

## Layered File System



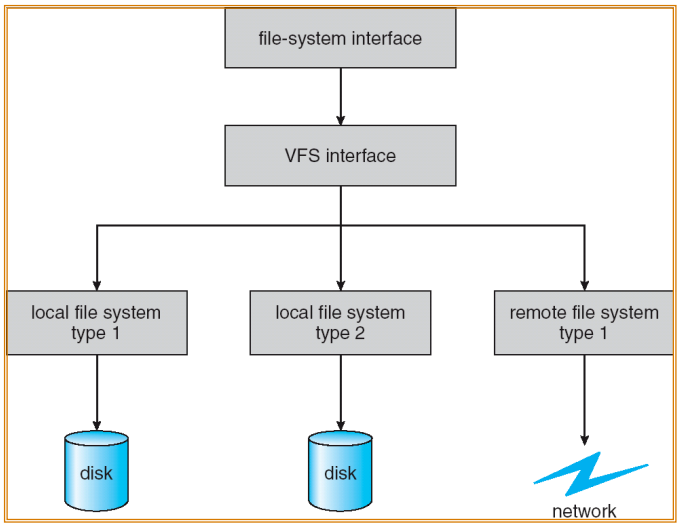
## In-Memory File System Structures




## Virtual File Systems

- Virtual File Systems (VFS) provide an object-oriented way of implementing file systems.
- VFS allows the same system call interface (the API) to be used for different types of file systems.
- The API is to the VFS interface, rather than any specific type of file system.

## Schematic View of Virtual File System




## Directory Implementation

- **Linear list** of file names with pointer to the data blocks.
  - simple to program
  - time-consuming to execute
- **Hash Table** – linear list with hash data structure.
  - decreases directory search time
  - **collisions** – situations where two file names hash to the same location
  - fixed size

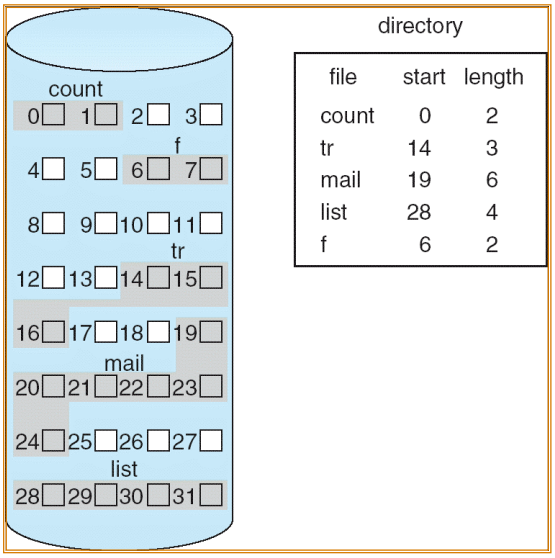
## Allocation Methods

- An allocation method refers to how disk blocks are allocated for files:
- **Contiguous allocation**
- **Linked allocation**
- **Indexed allocation**

## Contiguous Allocation

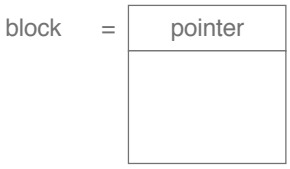
- Each file occupies a set of contiguous blocks on the disk
- Simple – only starting location (block #) and length (number of blocks) are required
- Wasteful of space (dynamic storage-allocation problem)
- Files cannot grow

# Contiguous Allocation of Disk Space



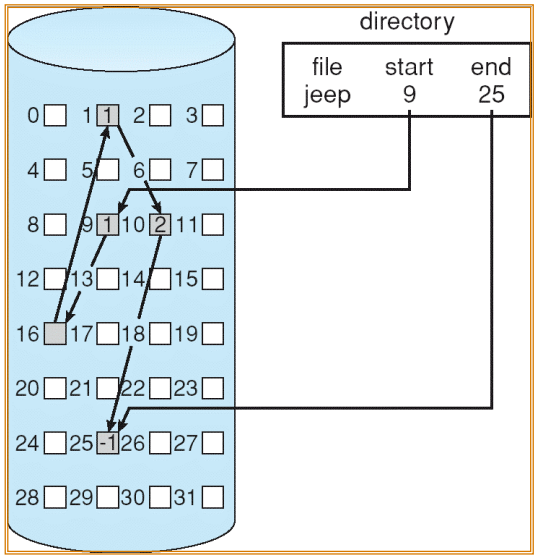

# Linked Allocation

- Each file is a linked list of disk blocks: blocks may be scattered anywhere on the disk.

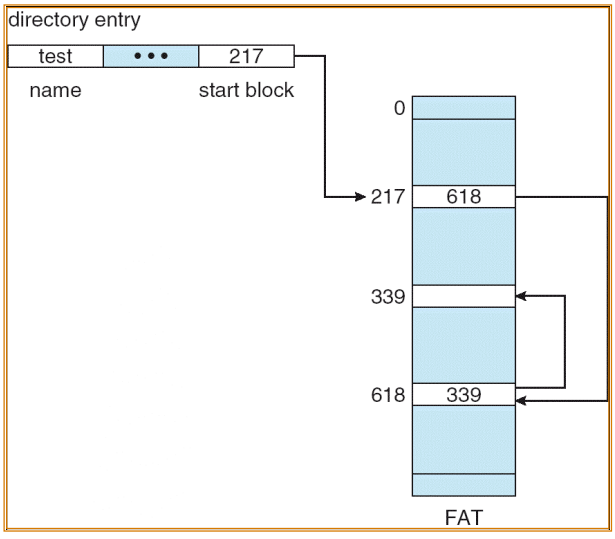



+ Simple – need only starting address
+ Free-space management system – no waste of space
- No random access
- Extra space required for pointers
- Reliability: what if a pointer gets corrupted?

# Linked Allocation

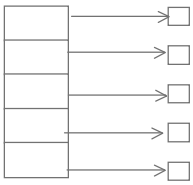



# File-Allocation Table



# Indexed Allocation

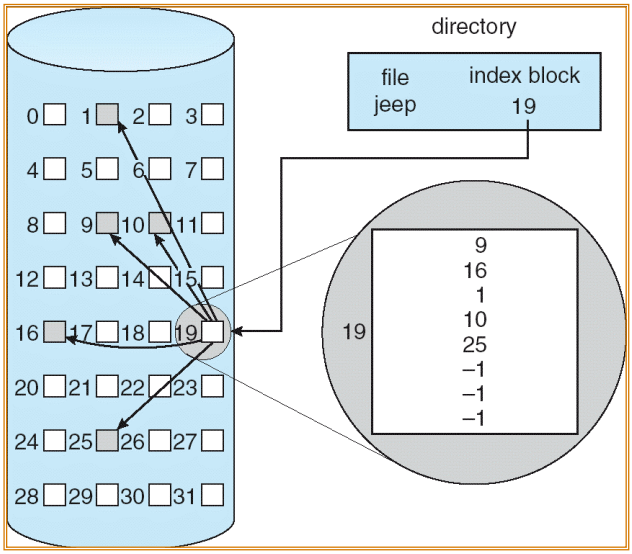
- Brings all pointers together into the *index block*, to allow random access to file blocks.
- Logical view.



index table

+ Supports direct access
+ Prevents external fragmentation
- High pointer overhead --> wasted space

# Example of Indexed Allocation



## Any Questions?




## Reading Assignment

- Read chapter 11 from Silberschatz.

# Acknowledgements


- “Operating Systems Concepts” book and supplementary material by Silberschatz, Galvin and Gagne.