File Management

Presented by

C.Namrata Mahender

File System

- File system is the most visible aspect of the OS
- File system's role is to provide on-line storage
- Contains
  - files (programs, data)
  - directory structure
  - partitions (in some systems)
- Issues include directory structure "shape" and directory/file protection

File Space

- OS abstracts from the physical property of the various secondary storage devices
  - tape, floppy disk, hard disk, optical disk, other
- Provides a uniform view for all of file space
- Files are mapped into their file space by the OS which is somewhat invisible to users

Files

- Named collection of related information recorded onto secondary storage
- Smallest allocable piece of secondary storage
  - May store:
    - binary information (executable programs, binary data
    - numeric
    - alphanumeric
    - alphabetic information
  - Info to be stored is defined by its creator and formatted (if necessary) by the application software

File Attributes

- Name
- Type
- Identifier
- Location
  - – pointer to a device and location
- Size (bytes, words or blocks)
- Protection
- Time, date, owner
  - – Time, date and owner may be kept for creation, last modification, last use

File Operations

- File is an abstract data type
  - Create a file
    - **find space in the file system, add a new file to the directory**
  - Write to a file
    - open the file
      - **search the directory for it, set a pointer to a location in the file**
      - **transfer information to storage device to write at the point of the pointer**

File Operations(cont..)

- Read a file
  - open file, transfer information from the device to memory
- Reposition pointer in a file
- Delete or Truncate (delete the data, leave the file) a file

Other Operations

- From the previous 6 operations, we can perform others:
  - Append - reposition to end of file, write
  - Copy - create a new file, read contents of old file and write to new file
  - Rename - Copy and Delete (impractical) or alter directory information

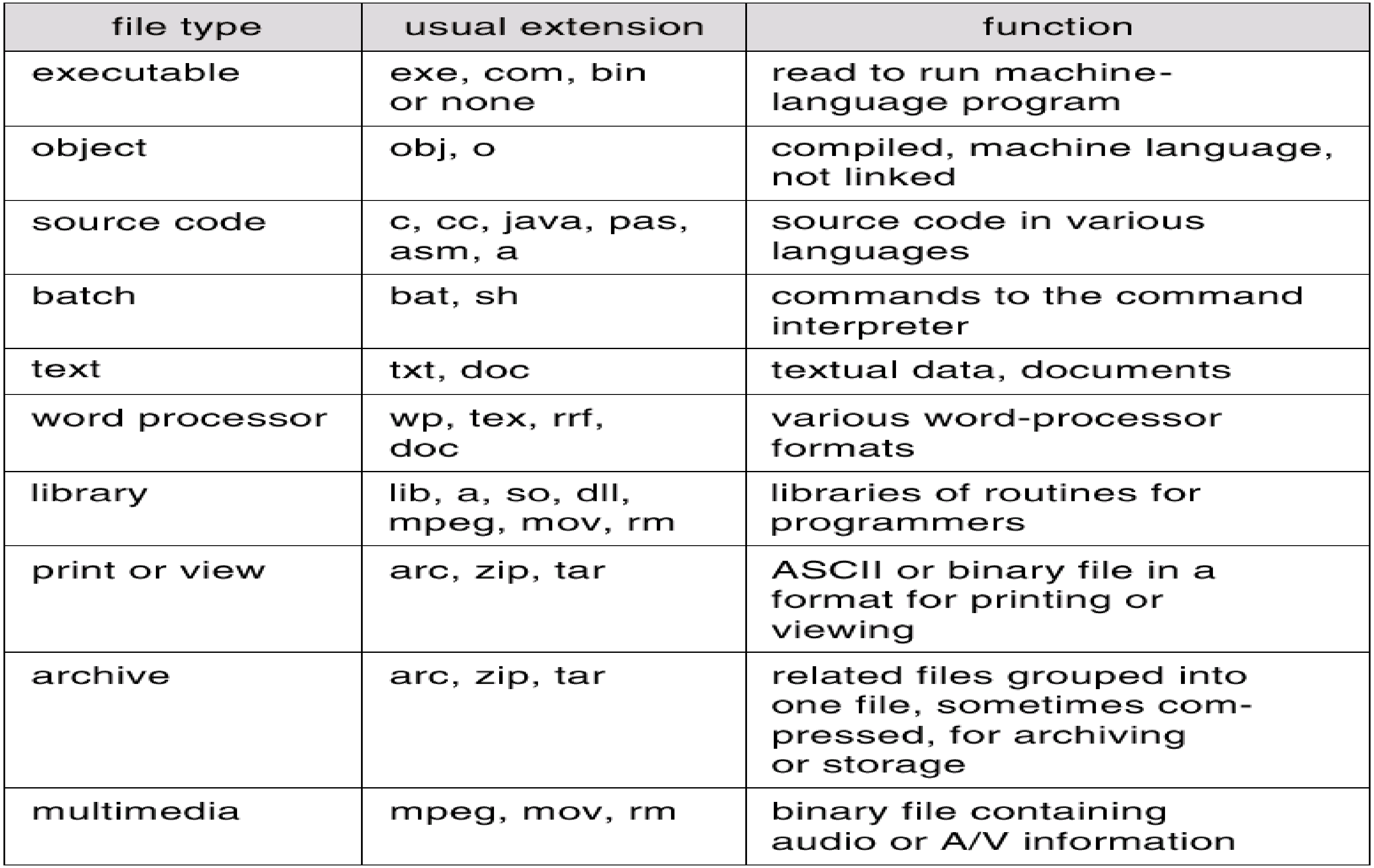
Opening and Closing files

- When a file is first referenced, the OS locates it in the file directory space and opens it by placing a pointer into a “open-file table”
  - **in memory or cache**
- Further references to the file use the table entry
- If multiple processes reference the file, an associated file count is used and increment for each open

Opening and Closing files(cont..)

- Files are closed either with an explicit close instruction or when a process terminates
  - **If a file count becomes 0, the entry is removed from the table and any necessary changes are written back to the file (such as new length, last modification,…)**

File Types – Name, Extension



file type	usual extension	function
executable	exe, com, bin or none	read to run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rrf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information

Access Methods

- Sequential Access
  - processed in order, one record after another
  - read or write and then move pointer to next record
- Direct Access
  - allows access to any record at any time
    - allows for random access
    - can also simulate sequential access by saving pointer position and incrementing/decrementing pointer

- Indexed Access
  - derive location through a key
    - such as name or code number
    - could have 1 location for each possible key or use hashing

Directory

- Directories map files onto their stored location
  – Also provide information such as names, sizes, protection, type of file, etc…
  – Directories represent logical partitions of files not physical partitions
    - a disk partition might contain several directories, 1 directory may be spread across multiple partitions

Operations Performed on Directory

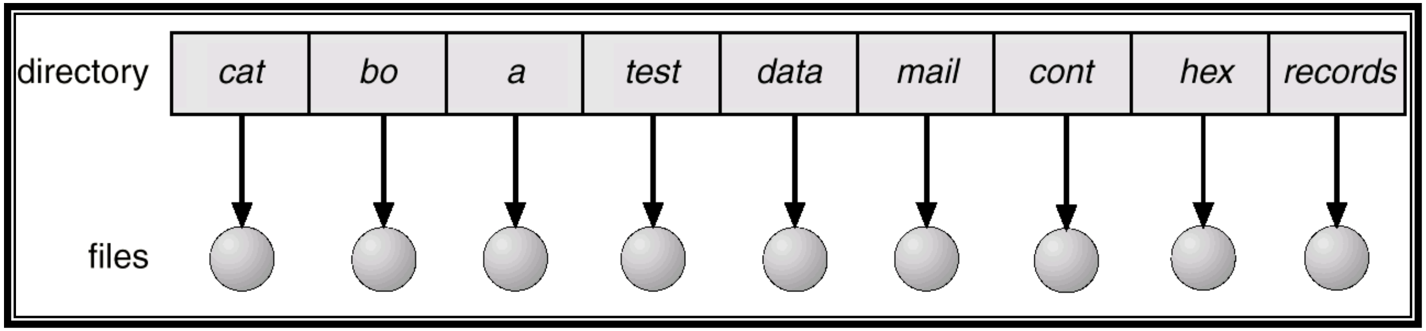
- Search for a file
- Create a file
- Delete a file
- List a directory
- Rename a file
- Traverse the file system

Organize the Directory (Logically) to Obtain

- **Efficiency** – locating a file quickly.
- **Naming** – convenient to users.
  - Two users can have same name for different files.
  - The same file can have several different names.
- **Grouping** – logical grouping of files by properties, (e.g., all Java programs, all games, …)

Single-Level Directory

- A single directory for all users.



directory

cat bo a test data mail cont hex records

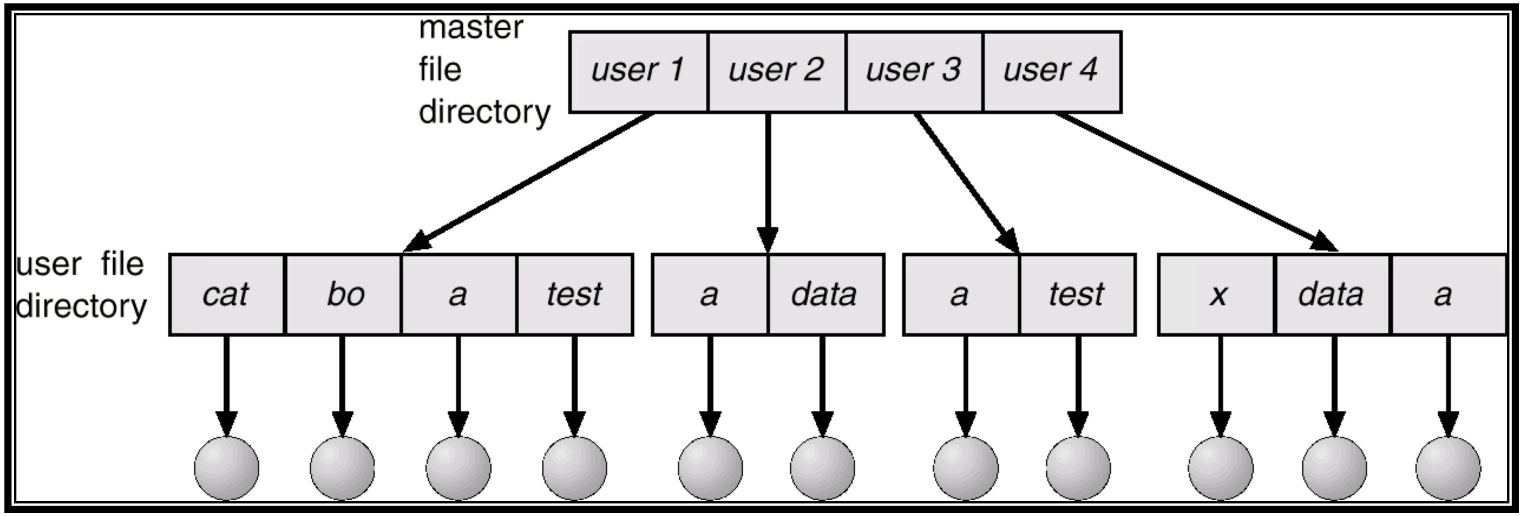
files

Naming problem

Grouping problem

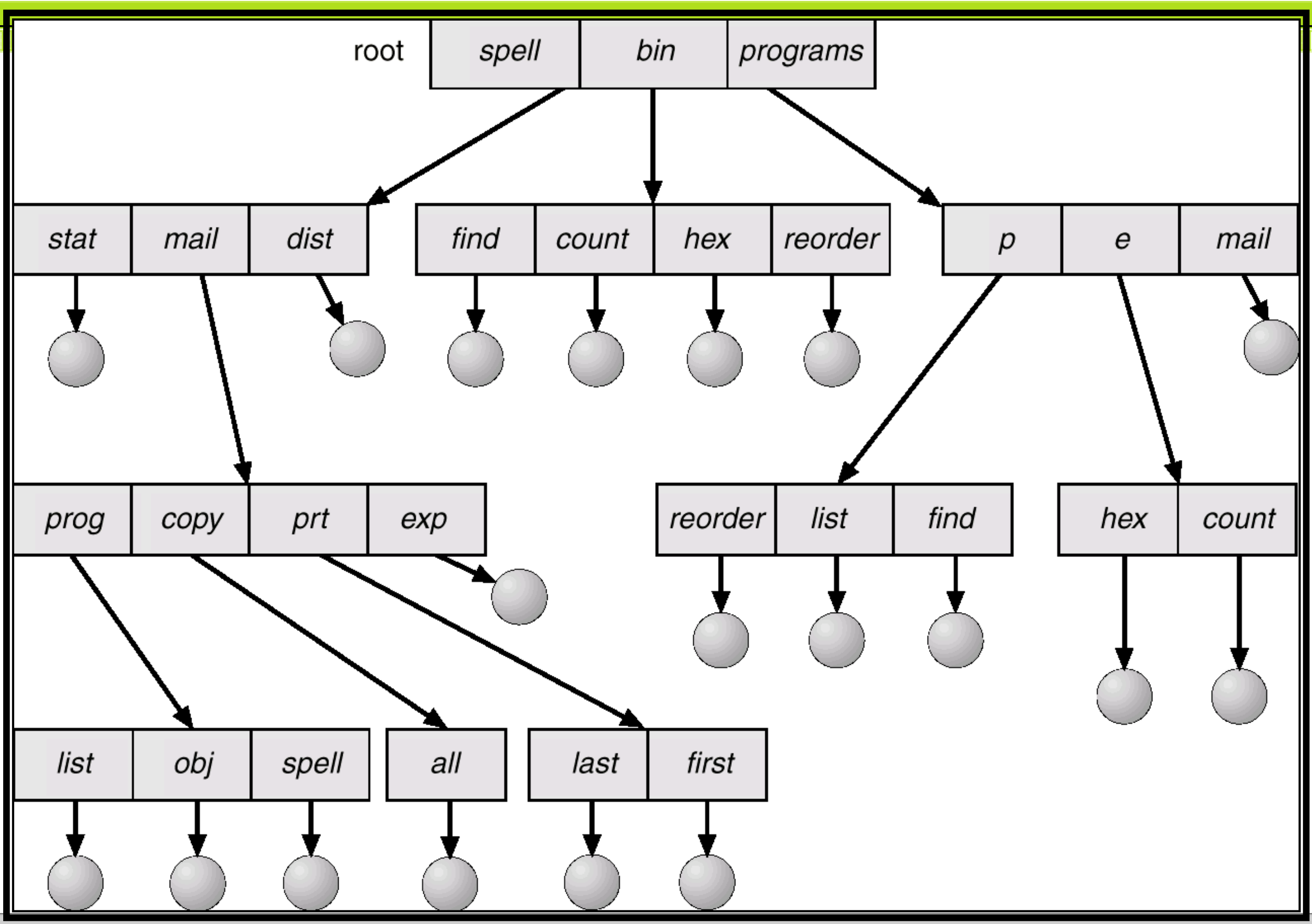
Two-Level Directory

- Separate directory for each user.



•Path name
•Can have the same file name for different user
•Efficient searching

Tree-Structured Directories



Tree-Structured Directories (Cont.)

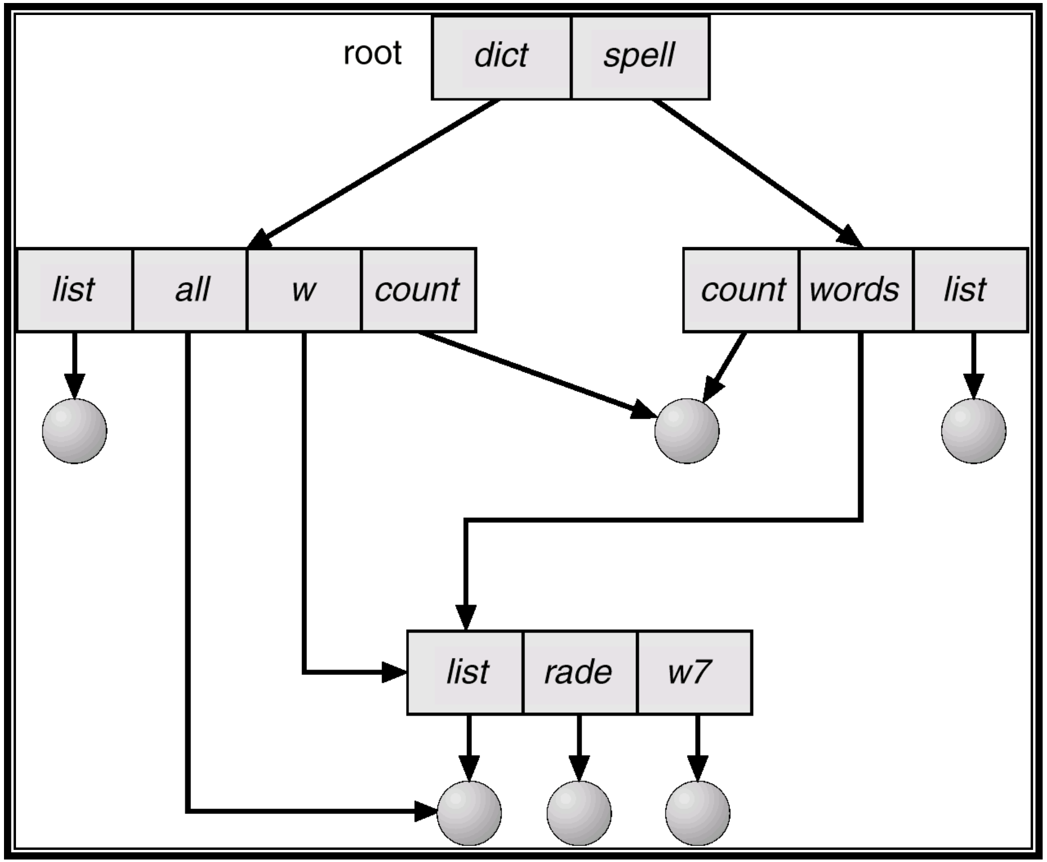
- Efficient searching

Path : 2 types

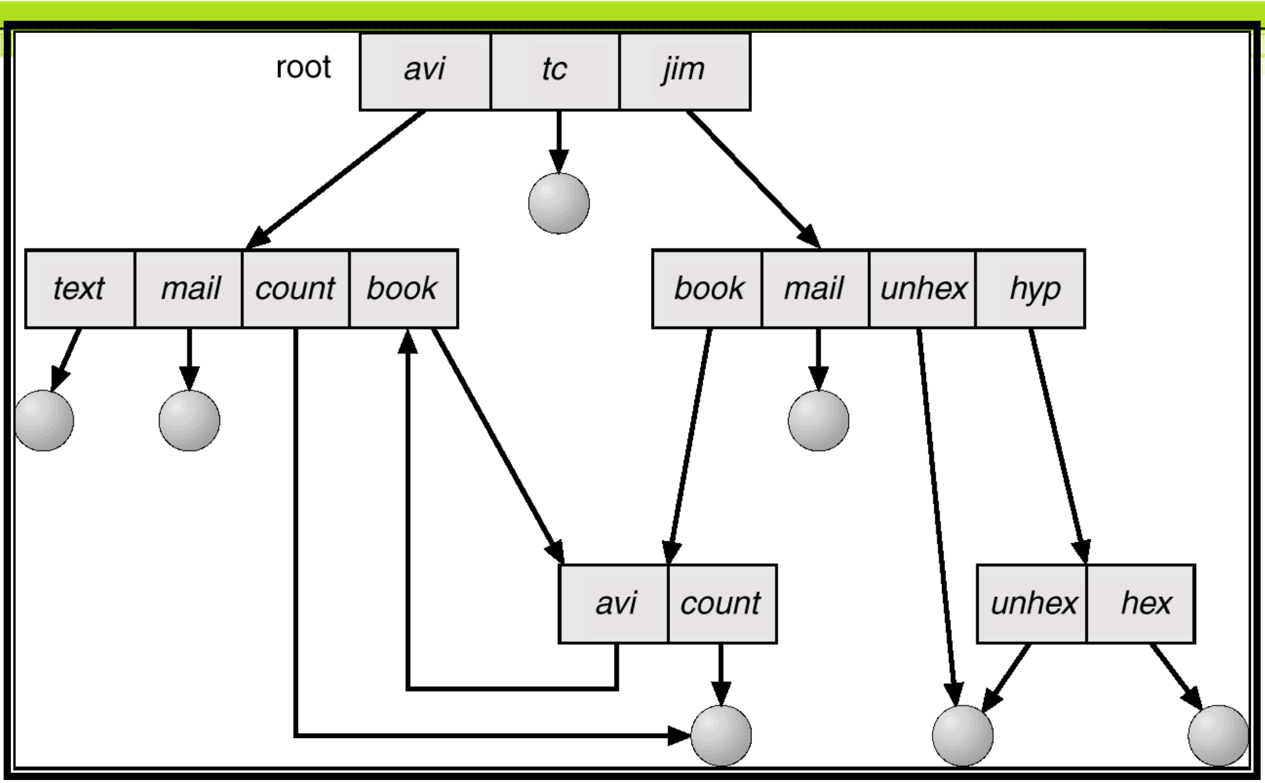
- A absolute path begins at root & follows a path down to specified file
- b. A relative path name defines a path from the current directory
- Sharing of files and directory not allowed

Acyclic-Graph Directories

- Have shared subdirectories and files.



General Graph Directory



General Graph Directory

- A garbage collector is necessary only because of possible cycles in graph
- Garbage collector determine when the reference has been deleted and the space can reallocated.
- It involves traversing of entire file system marking that can accessed or in use.
- Then a second pass collects everything that is not marked onto thelist of free space

Directory Structures

- **Single-Level**
  - all files at one level, no subdirectories
    - awkward to use and impossible if multiuser system
- **Two-Level**
  - first level are the users, second level are user files
    - awkward but possible for multiusers -- no file sharing though
- Tree-Structured
  - typical and most common method
- Acyclic-Graph
  - allows file sharing but requires additional mechanisms for file deletion
- General Graph
  - allows file sharing but requires loop detection so that traversal does not lead to infinite looping!

Protection

- File owner/creator should be able to control:
  - what can be done
  - by whom
- Types of access
  - Read
  - Write
  - Execute
  - Append
  - Delete
  - List

Access Lists

- One means for protection is to associate a list of users who have authorized access for each file
  - A file could have different lists for different types of access
    - read list, write list, etc…
- Problem
  - lists could be as long as the number of users in the system making directories very large

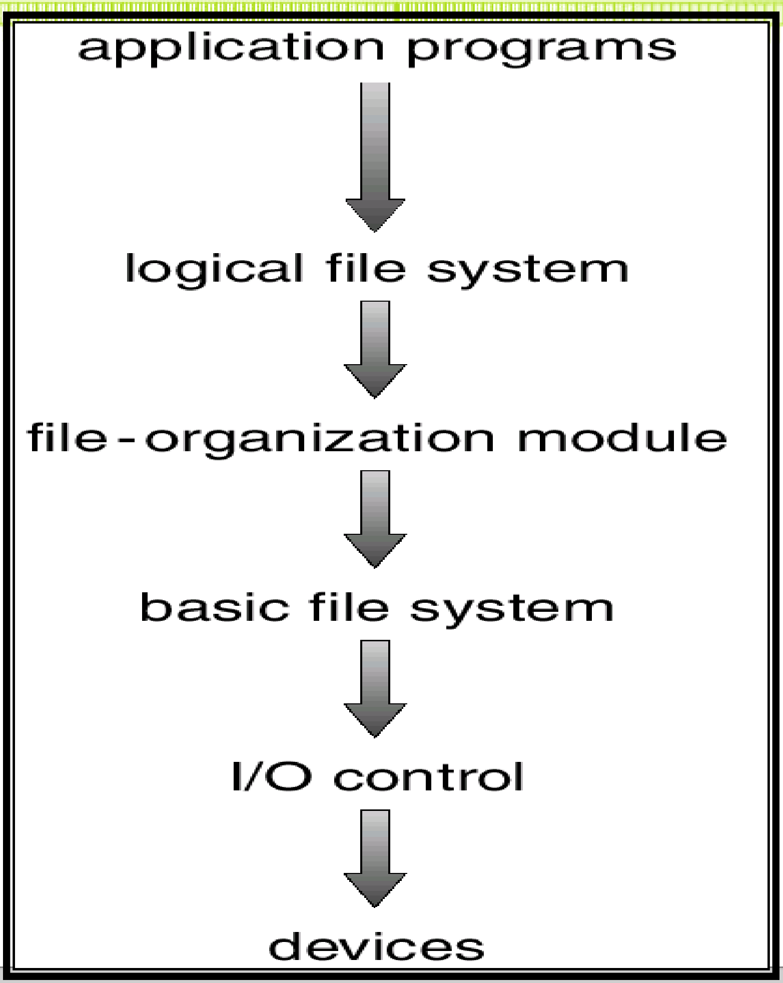
Other Protection Approaches

- Passwords
  - associate a password with each file
    - or read password, write password, execute password
- Directory Passwords
  - have a password for a directory and the password gives full access to the everything in the directory
- Disallow sharing
  - single user access to any file
    - method used by many PCs before networking became common

File-System Structure

- File structure
  - – Logical storage unit
  - – Collection of related information
- File system resides on secondary storage (disks).
- File system organized into layers.
- *File control block* – storage structure consisting of information about a file.

Layered File System



application programs

logical file system

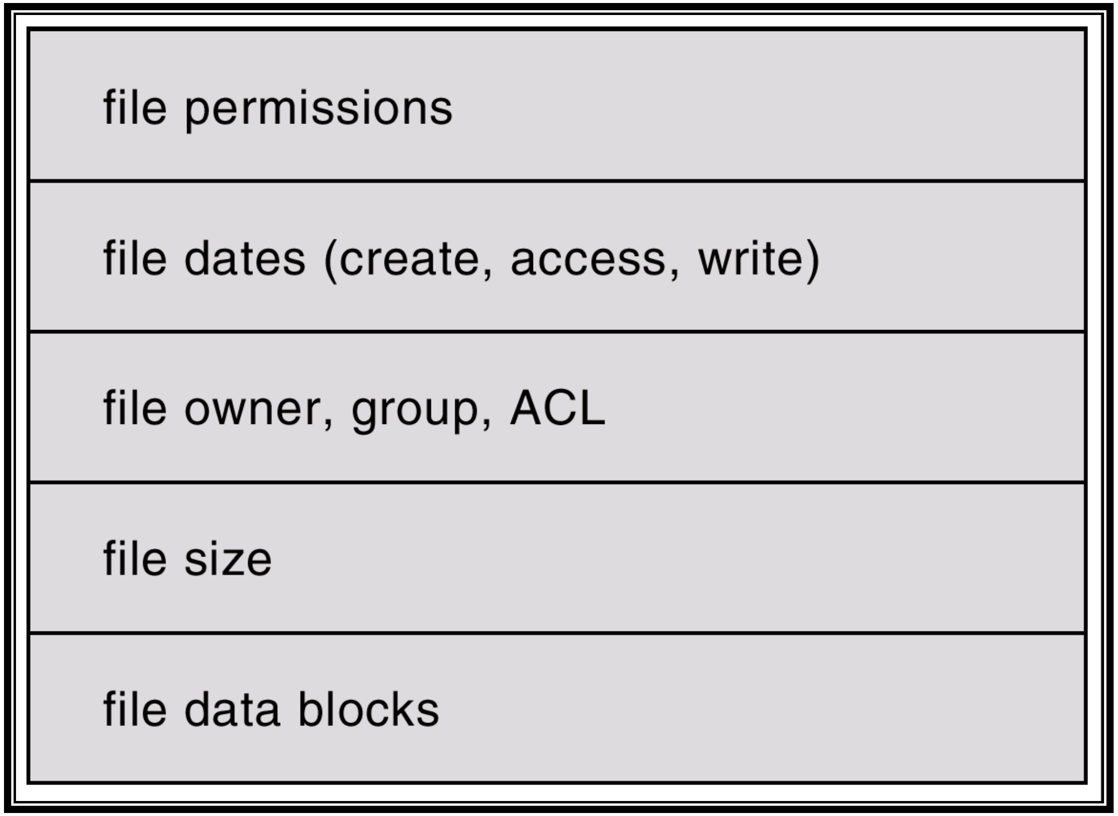
file-organization module

basic file system

I/O control

devices

A Typical File Control Block



file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks

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

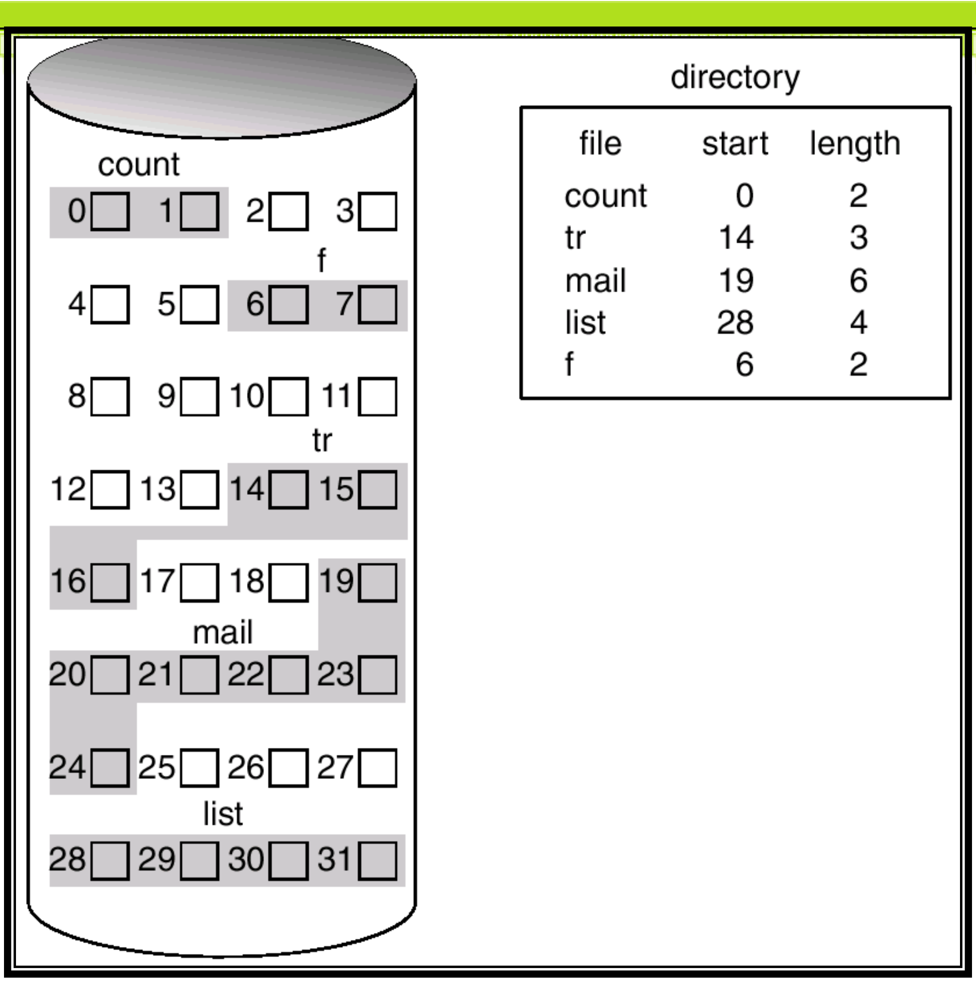
Contiguous Allocation

- Each file occupies a set of contiguous blocks on the disk.
- Simple – only starting location (block #) and length (number of blocks) are required.
- Random access.
- Wasteful of space (dynamic storage-allocation problem).

Example if file is n blocks & starts at b

Then b,b+1…..b+n-1

Contiguous Allocation of Disk Space

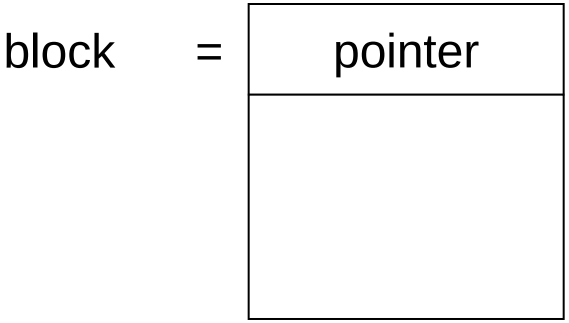


directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

Linked Allocation

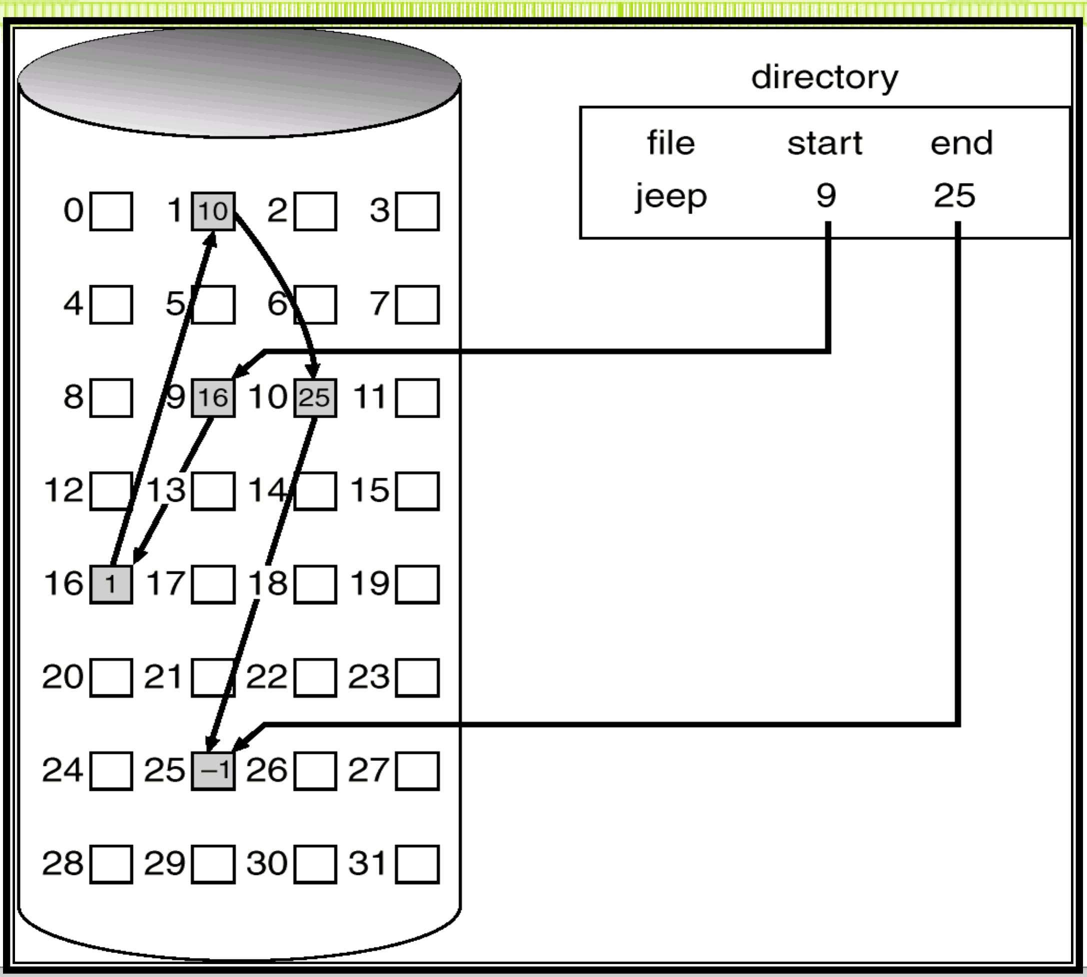
- Each file is a linked list of disk blocks: blocks may be scattered anywhere on the disk.



Linked Allocation (Cont.)

- Simple – need only starting address
- Free-space management system – no waste of space
- No random access
- Mapping

Linked Allocation (Cont.)



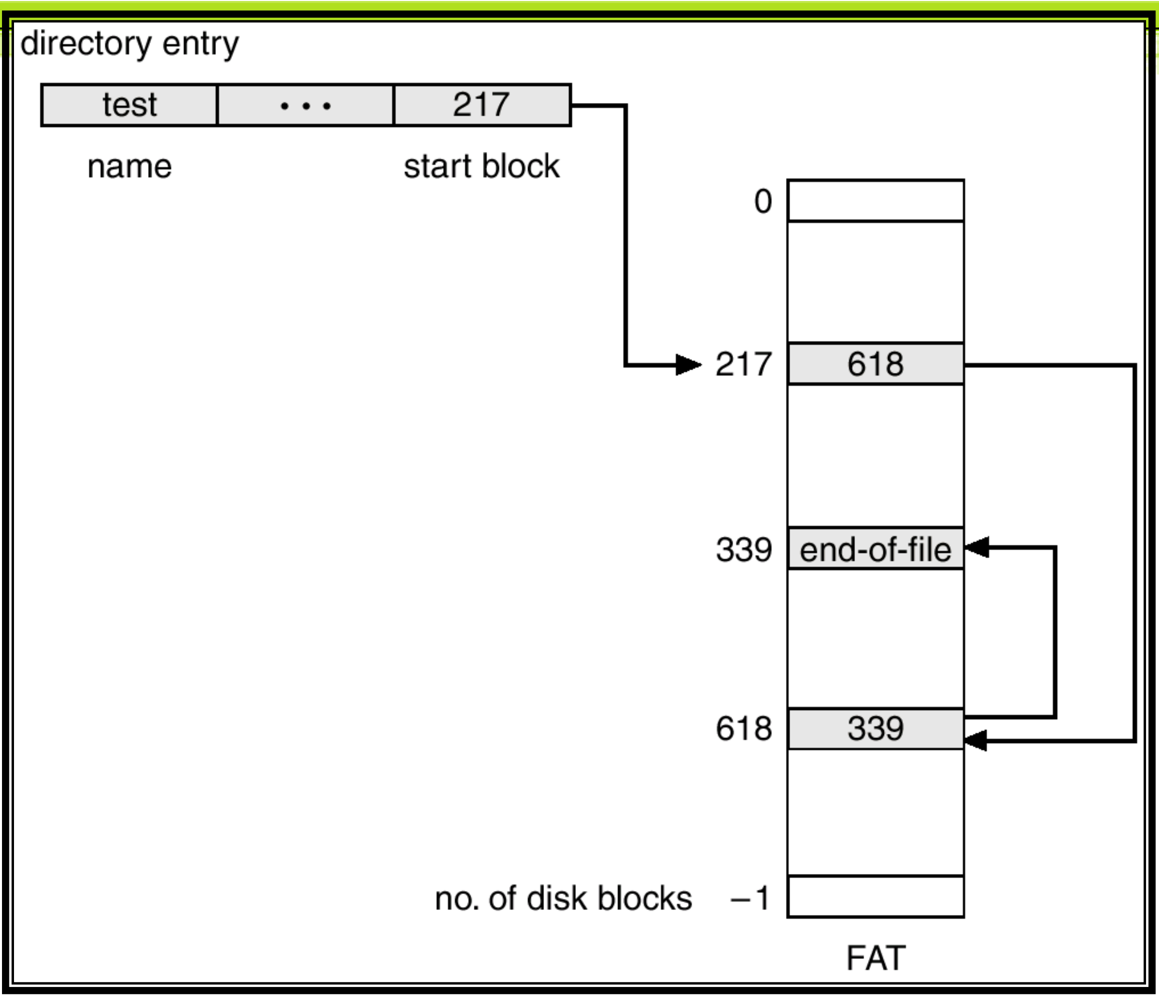
directory

file	start	end
jeep	9	25

File-Allocation Table

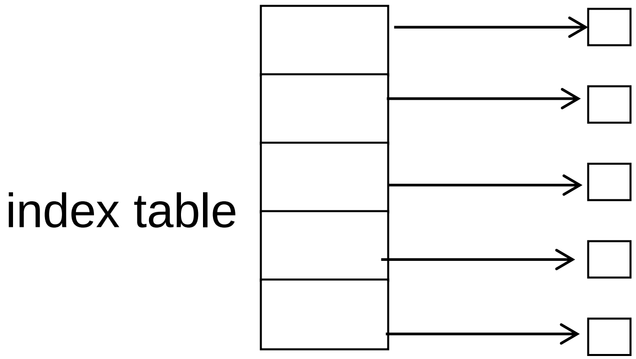
- It is a variation of linked list
- The directory entry contains the block number of the first block of the file.
- The table entry index by the block number then contains the number of next block in the file
- The last block has an end of file value
- Free block are shown by zero value in table

File-Allocation Table



Indexed Allocation

- Brings all pointers together into the ***index block***.
- Logical view.



Recovery

- In a system crash, some files may have been open and partially altered
  - a consistency checker is often used to determine files where this has occurred and tries to fix them
- If the disk fails, a backup can be used to restore the information
  - backups usually stored on magnetic tape
    - A useful backup strategy is required!

Thank you