COMP 3511
Operating Systems

Lab 08

Outline

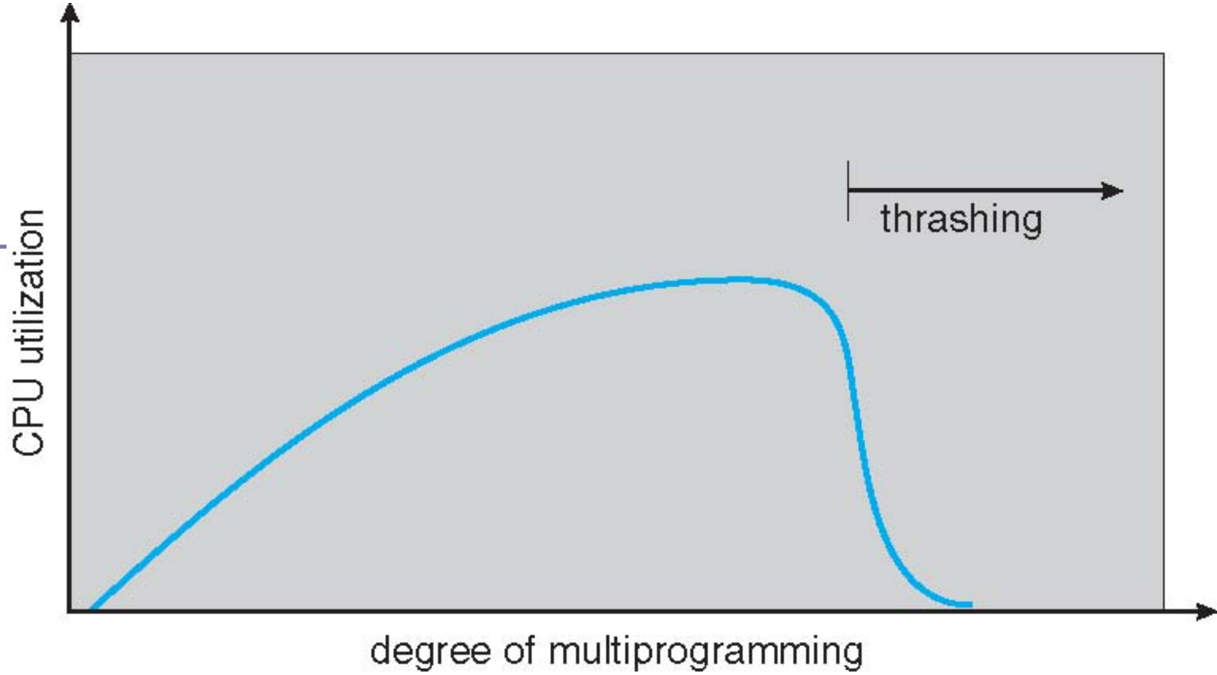
- **Thrashing**
- File Systems
  - File
  - Directory Organization
- File System Implementation
  - File System Structure
  - Allocation Methods
  - Free Space Management

Thrashing

- **Thrashing** → a process is busy swapping pages in and out
  - If a process does not have “enough” pages, the page fault rate is very high
  - very quickly need replaced frame back

- This leads to
  - Low CPU utilization, much time in I/O
  - Operating system thinking that it needs to increase the degree of multiprogramming
  - Another process added to the system

Thrashing



- How to detect?

  Level of

  CPU utilization

  Vs.

  Level of multiprogramming

- How to eliminate?
  - reduce the level of multiprogramming
  - use local replacement algorithm.
    - the number of frames allocated to a process is fixed

Outline

- Thrashing
- File Systems
  - **File**
  - Directory Organization
- File System Implementation
  - File System Structure
  - Allocation Methods
  - Free Space Management

File – Logical Storage Unit

- File Properties
  - Name, identifier, type, location, size, protection
  - Time, date, user identification
  - Kept in directory structure, maintained in disk
- File Operation
  - Create, Delete
  - Write, Read, Open, Close, etc.

File Open

- to manage open files:
  - **Open-file table**: tracks open files
  - **File pointer**: pointer to last read/write location, per process that has the file open
  - **File-open count**: counter of number of times a file is open – to allow removal of data from open-file table when last process closes it
  - **Disk location of the file**: cache of data access information
  - **Access rights**: per-process access mode information

Access Methods

- **Sequential Access**

```
read next
write next
reset
```

- **Direct Access**
  - file is made up of fixed length logical records

```
read n
write n
position to n
     read next
     write next
rewrite n
```

n = relative block number

Outline

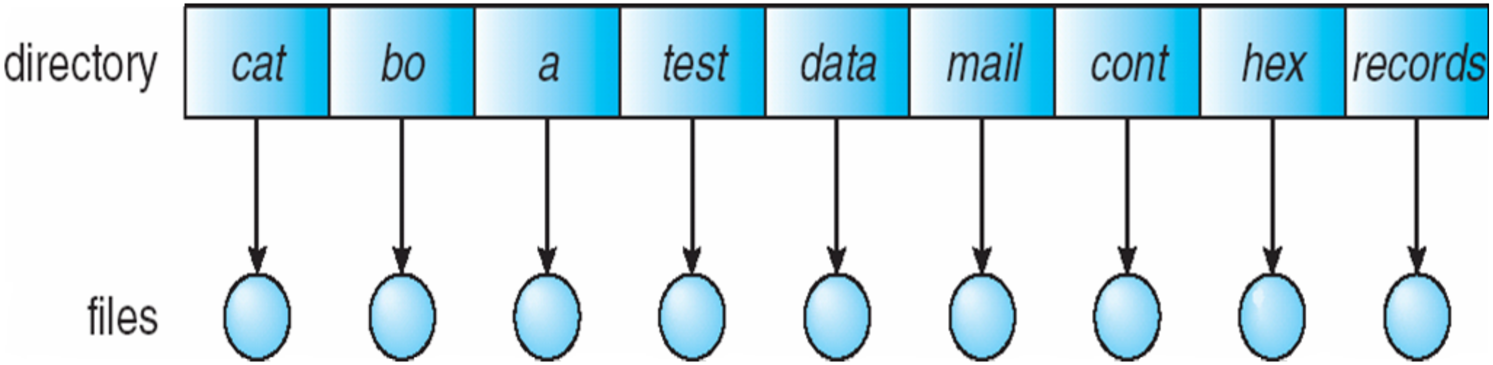
- Thrashing
- File Systems
  - File
  - **Directory Organization**
- File System Implementation
  - File System Structure
  - Allocation Methods
  - Free Space Management

Goals for directory organization

- 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

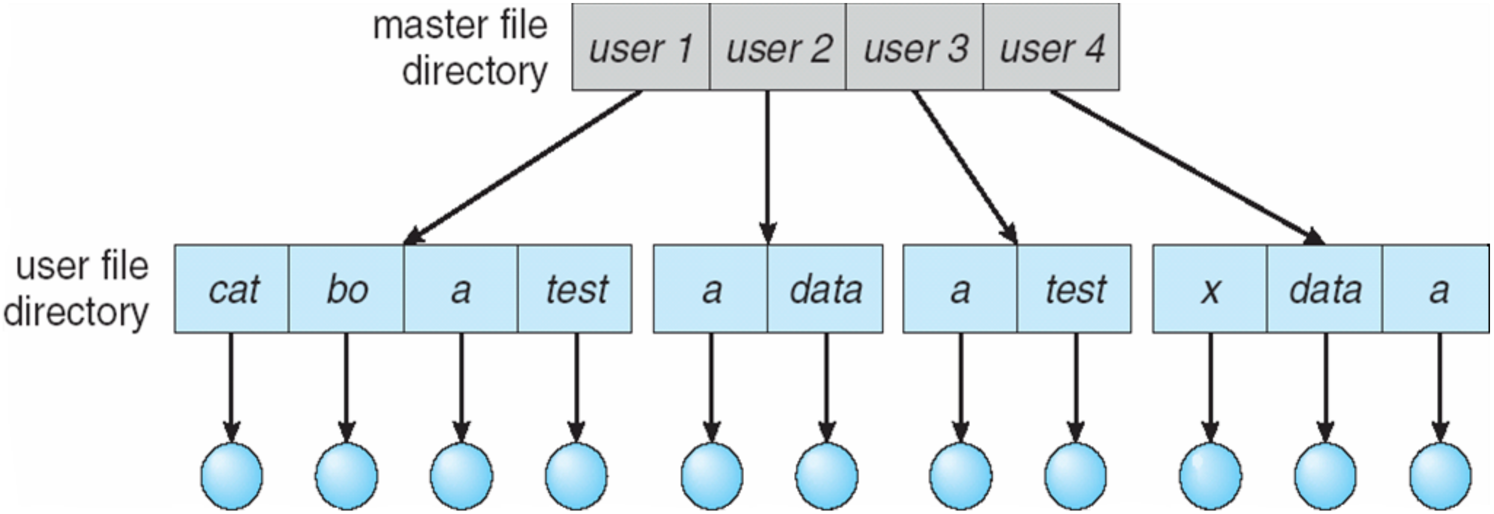
All files are kept in the same directory.



- Naming problem
- grouping problem

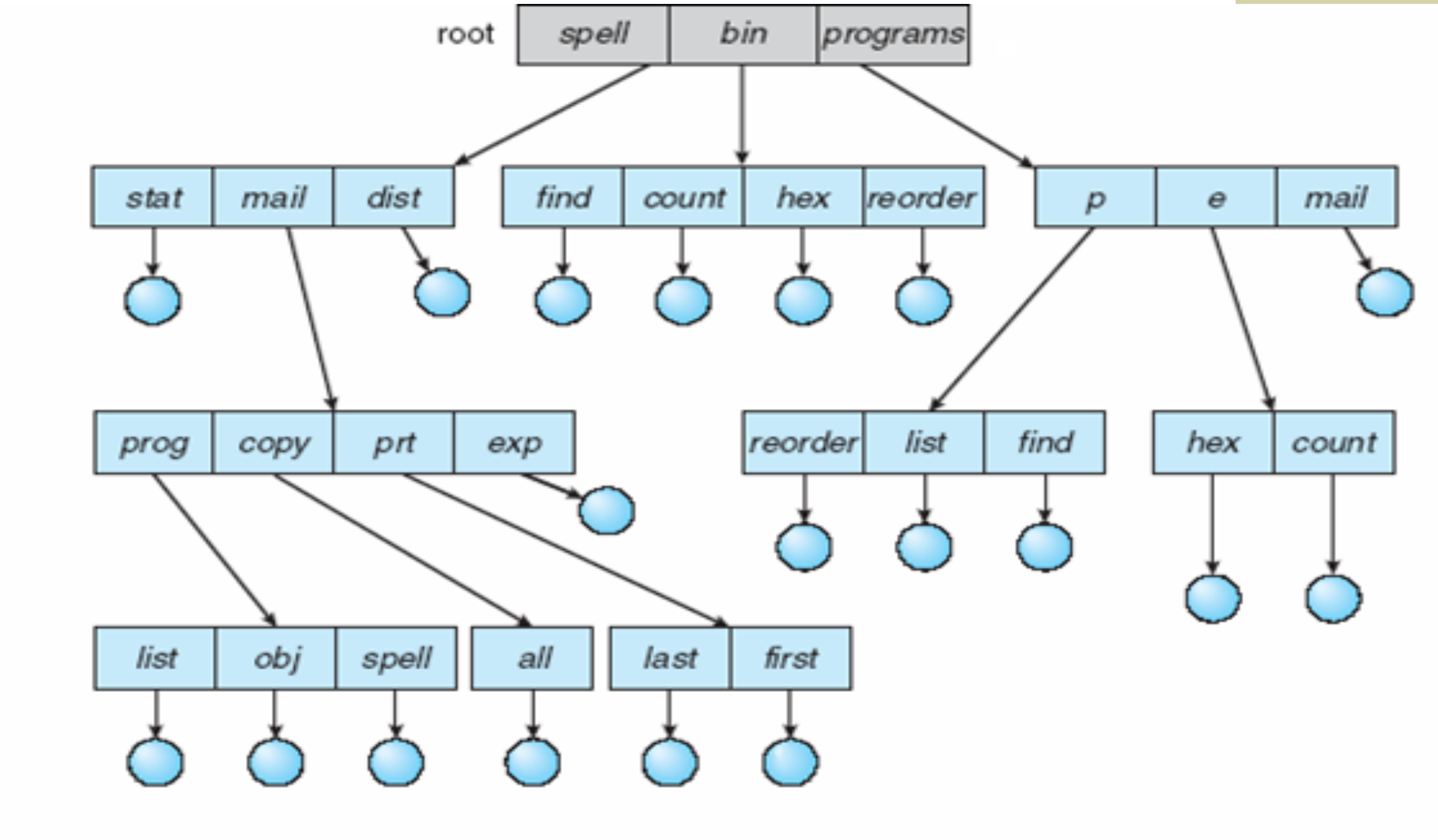
Two-level Directory

- Separate directory for each user



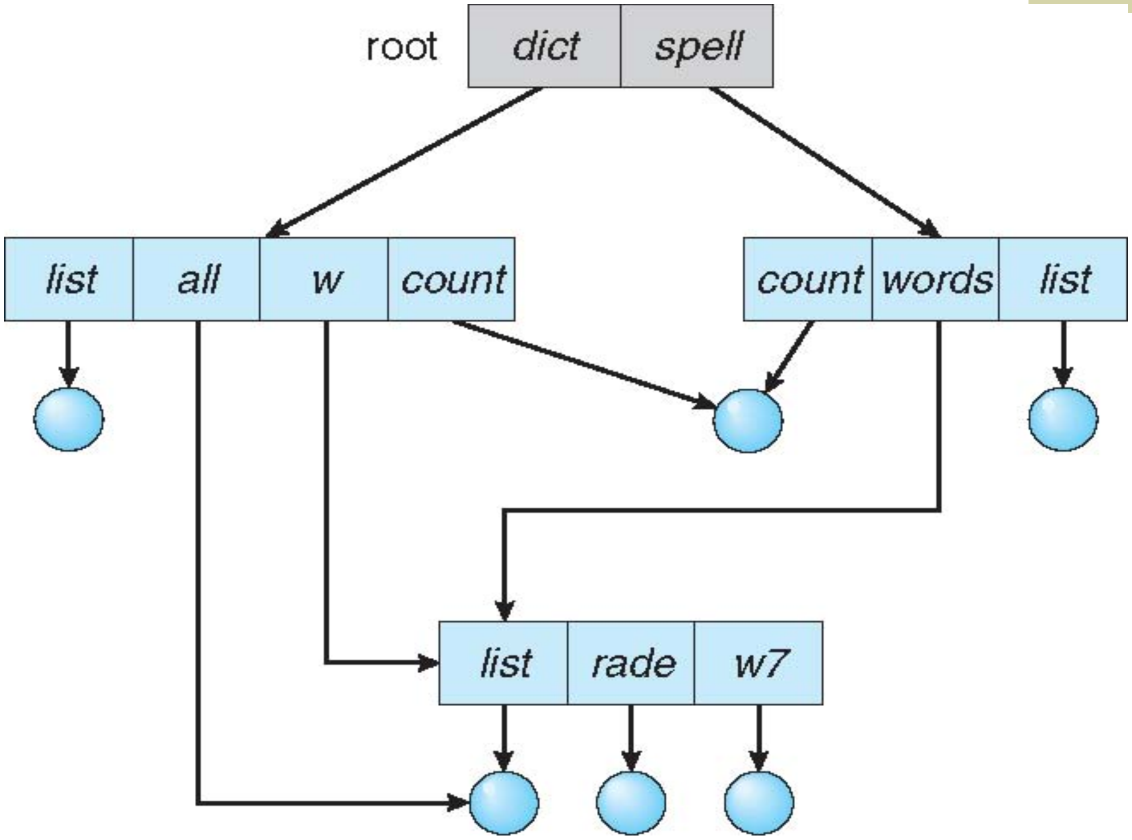
- Enable naming
- Efficient searching
- No grouping capability

Tree-structured Directory



- grouping capability
- No sharing ability

Acyclic–Graph Directory

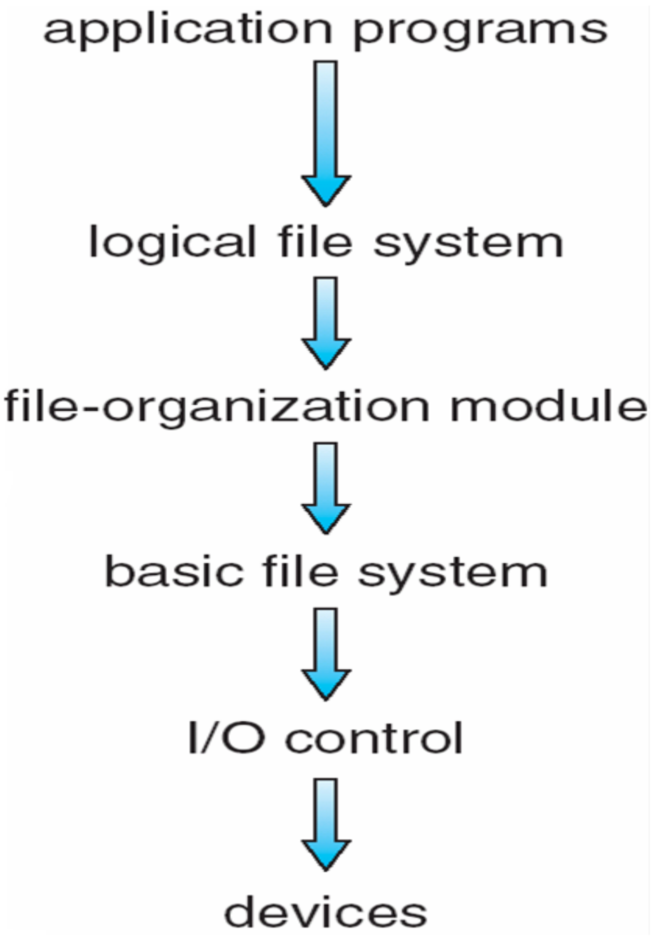


- File sharing enabled

Outline

- Thrashing
- File Systems
  - File
  - Directory Organization
- File System Implementation
  - **File System Structure**
  - Allocation Methods
  - Free Space Management

Layered File System



- Provide file names

- Transfer file names into logical address

- Transfer logical address to physical blocks

- Transfer physical blocks to general I/O control command

- Transfer I/O control command to hardware specific commands

File system implementation

- **Boot control block**

  - contains info needed by system to boot OS from that volume

  - Needed if volume contains OS, usually first block of volume
- **Volume control block** (**superblock, master file table**)

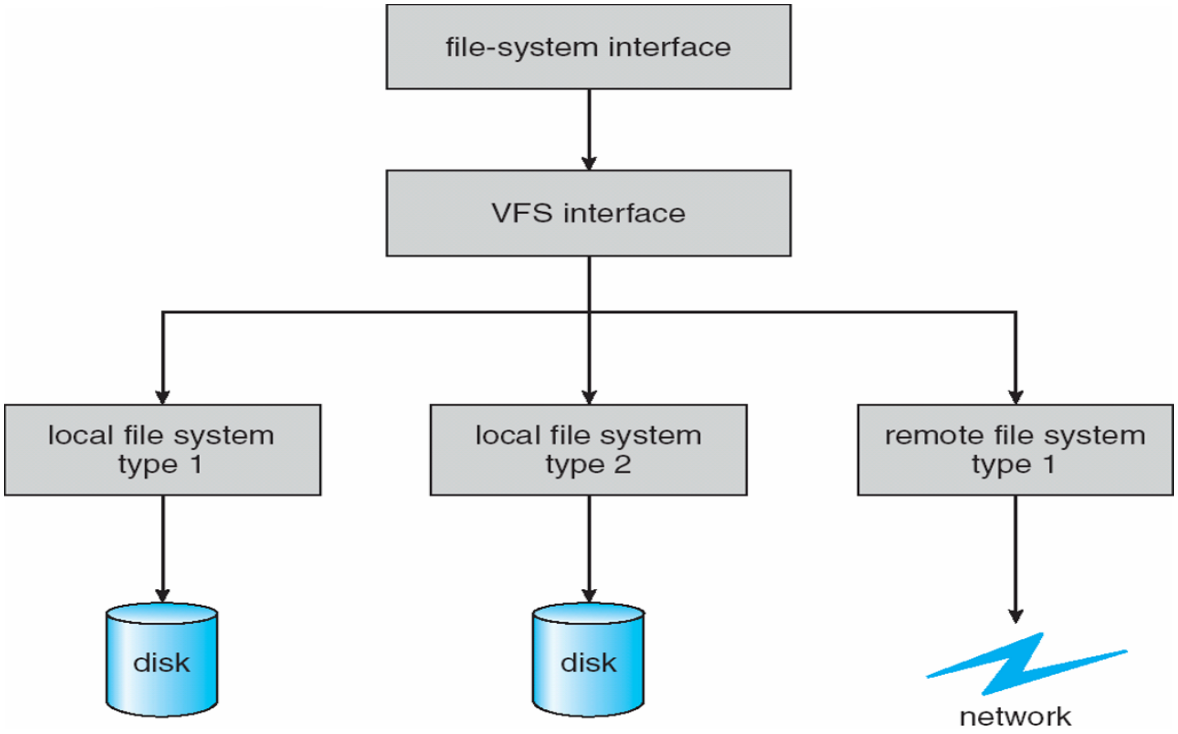
  - contains volume details

  - Total # of blocks, # of free blocks, block size, free block pointers or array
- **File Control Block (FCB)**

  - contains many details about the file

  - owner, permissions, size, dates
  - File data blocks or pointers to file data blocks

Virtual file systems

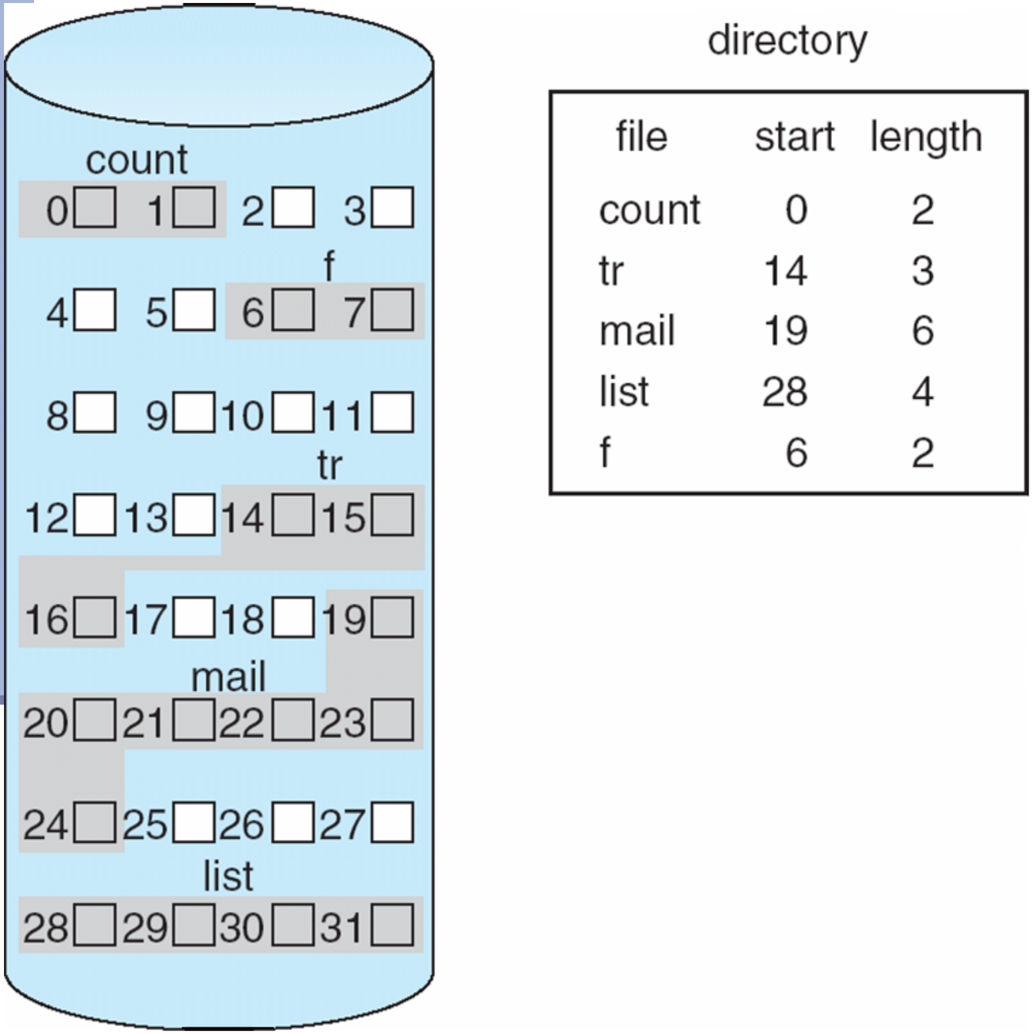


- Separate file-system generic operations from implementation details

Outline

- Thrashing
- File Systems
  - File
  - Directory Organization
- File System Implementation
  - File System Structure
  - **Allocation Methods**
  - Free Space Management

Allocation methods - Contiguous

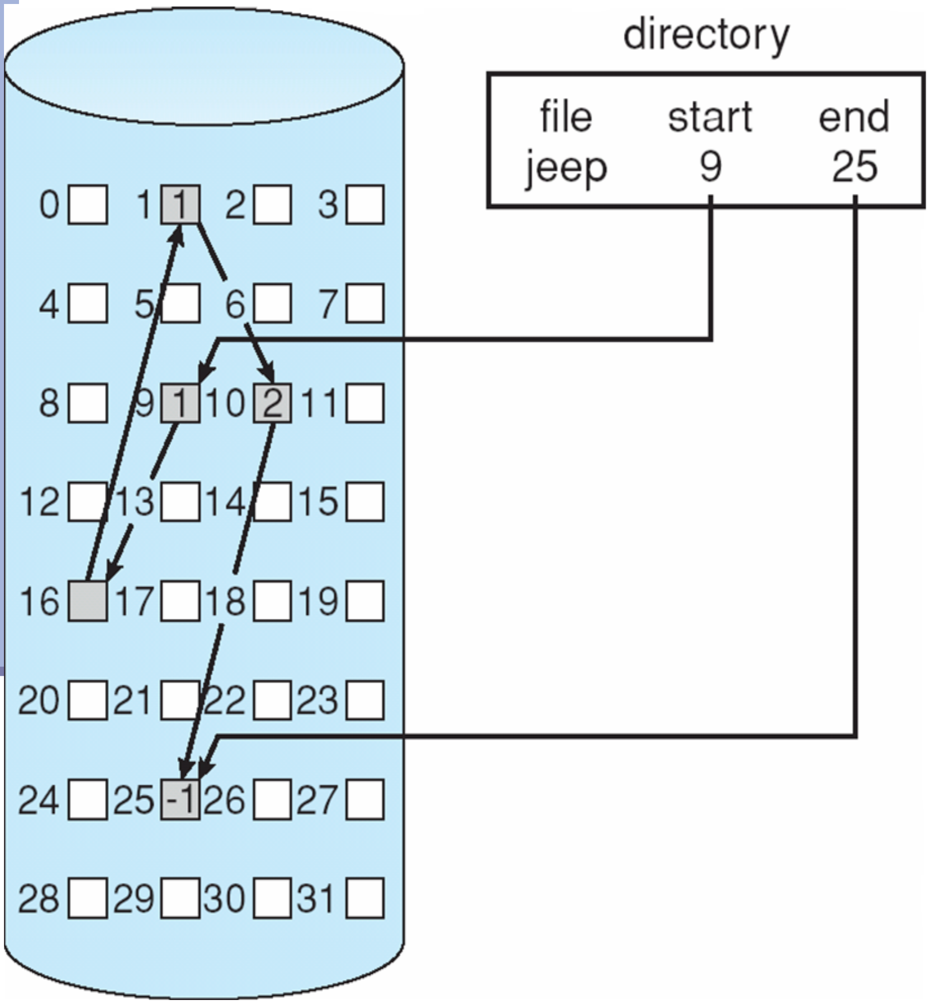


directory

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

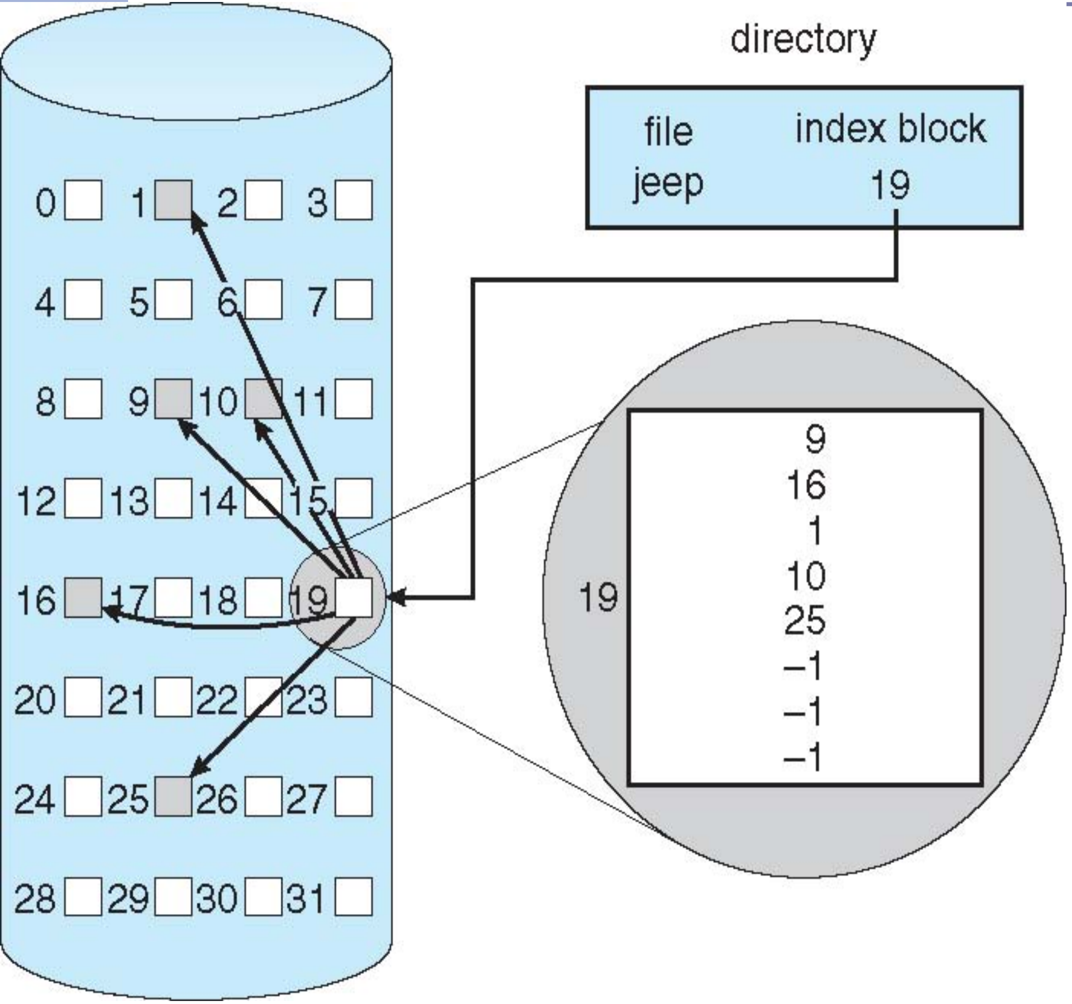
- Each file - a set of contiguous blocks
- Simple
  - Starting block
  - Number of blocks
- Problem
  - External fragmentation

Allocation methods - Linked



- Each file – a linked list of blocks
- benefit
  - No external fragmentation
- Problem
  - Poor reliability
  - complex

Allocation methods - Indexed



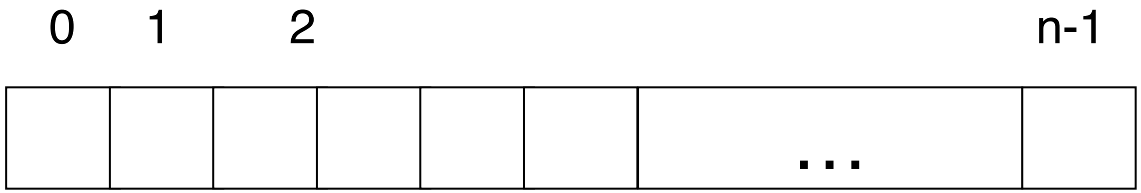
- Each file has its own block(s) of pointers to its data blocks
- benefit
  - No external fragmentation
  - More reliability
- Problem
  - Waste space

Outline

- Thrashing
- File Systems
  - File
  - Directory Organization
- File System Implementation
  - File System Structure
  - Allocation Methods
  - **Free Space Management**

Free-Space Management – Bit Vector

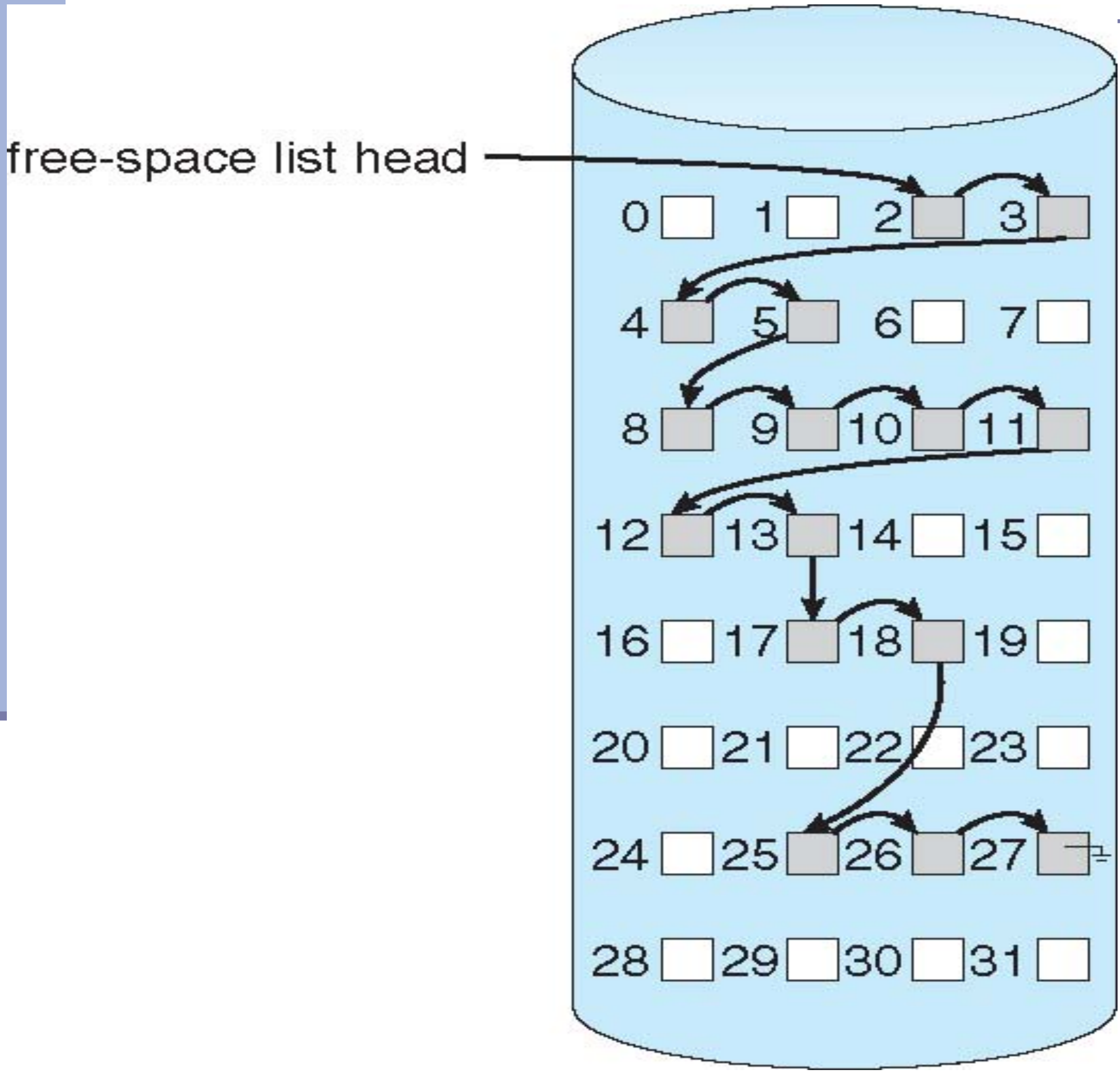
- **free-space list** to track available blocks/clusters
- **Bit vector** or **bit map** (*n* blocks)



$$\text{bit}[i] = \begin{cases} 1 \rightarrow \text{block}[i] \text{ free} \\ 0 \rightarrow \text{block}[i] \text{ occupied} \end{cases}$$

- Pros:
  - Simple
  - Efficient (in find the first continuous n free blocks)
- Cons:
  - Extra space needed

Free-Space Management – Linked List



- link together all the free disk blocks
- Keep a pointer to the first free block
- Benefit:
  - Space saving
- problem
  - Not efficient