

CS 4410
Operating Systems

File System Interface

Summer 2016
Cornell University

Today

- Managing data in persistent storage.
- File system
 - Organization
 - Interface

Storing data

- Computers must be able to reliably store data.
 - From smart phones to data centers.
- Varied amount of data.
 - From a few GBs to “big data”!
- Varied formats of data:
 - Photos, music, texts, programs and OS code, ...
- The operating system provides appropriate abstractions for managing heterogeneous stored data.



Nonvolatile storage

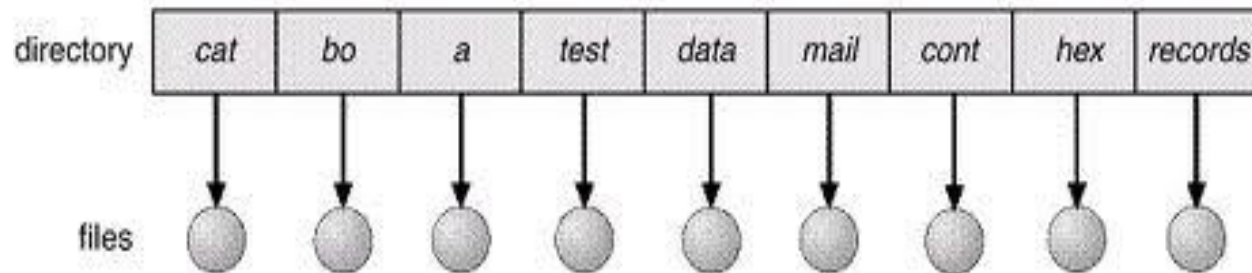
- Data in DRAM memory can be lost when a crash or a power outage happens.
- But user's data needs to be persistently stored.
- Nonvolatile storage is persistent, with
 - higher capacity and lower cost than DRAM.
- Examples: magnetic disk, flash storage. But:
 - only coarse-grained units can be accessed (not individual bytes), and
 - each storage access is slower than a memory access.
- OS manages persistent storage differently than memory.

File system

- A file system is a common abstraction to allow applications to access persistent storage.
- Key parts of a file system abstraction:
 - File: named collection of data
 - Directory: list of files and directories
 - Recursive structure.
- The same abstraction for all storage devices.
 - Different implementation.

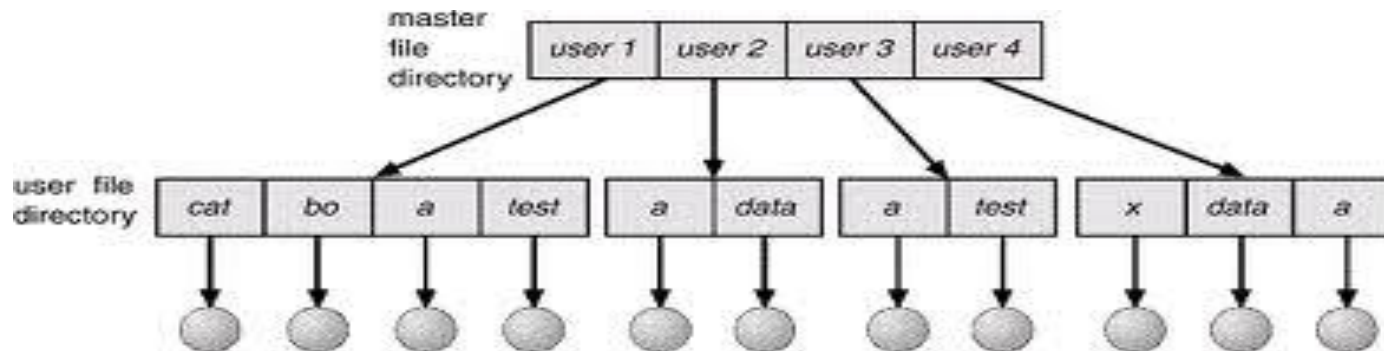
Organization of files

- Single level



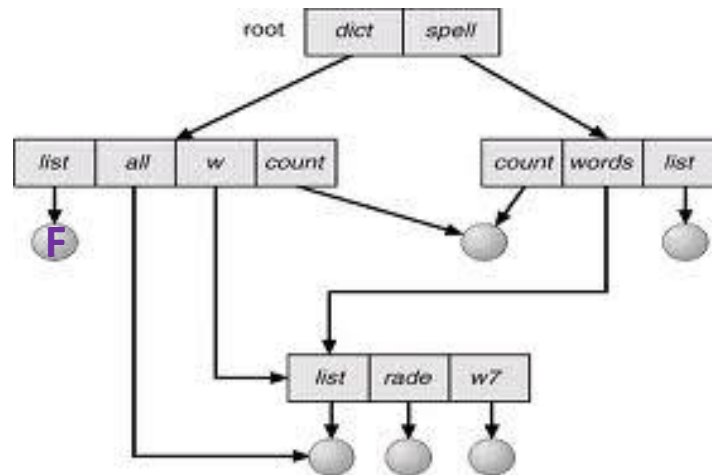
Organization of files

- Two-level



Organization of files

- Directed acyclic graph (DAG)

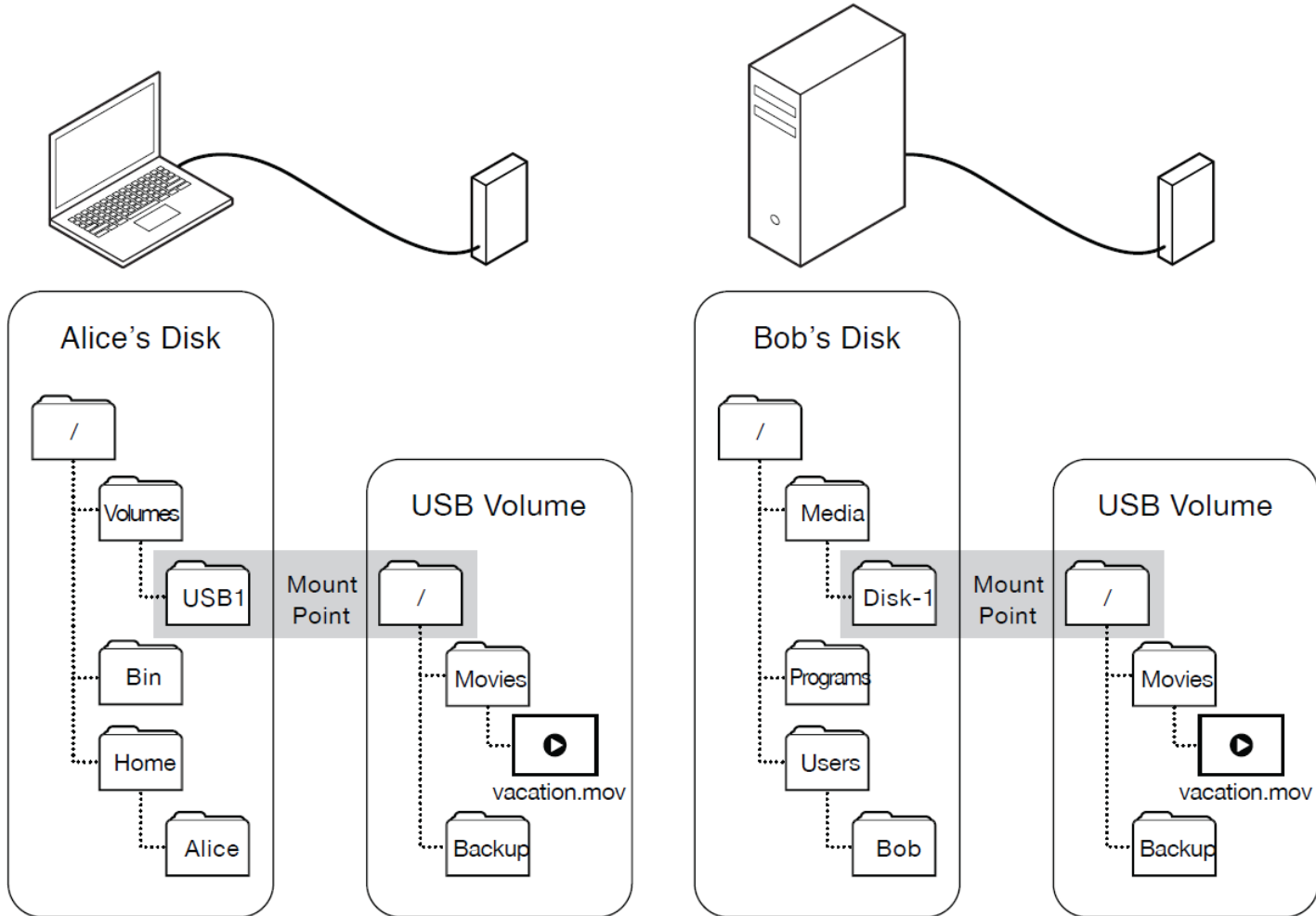


- The absolute path of file F is: /dict/list/F
- The organization of files should not contain cycles.

Managing multiple file systems

- A single computer can make use of multiple file systems stored on multiple volumes.
 - A volume is a collection of physical storage resources than form a logical storage device.
- Each volume can be mounted in a single logical hierarchy.
- Mounting a volume on an existing file system creates a mapping from some path in the existing file system to the root directory of the mounted volume's file system.

Mounting a USB volume



File system API

- Create and delete files.
- Open and close.
- Read and write.

Open file

- Operating systems require processes to explicitly open files before accessing them.
- Thus, searching the file and getting information about the file happens only once, rather than every time the file is accessed.
 - File information: location, size, access rights.
- When a process opens a file, the OS creates a data structure with
 - the file's information and
 - a pointer to the process' current position within the file,
- and returns a reference to that data structure.
 - File descriptor.
- Each subsequent access to that file is accompanied by the file descriptor.

Today

- Managing data in persistent storage.
- File system
 - Organization
 - Interface

Coming up...

- Next lecture: storage devices
- HW4: released today