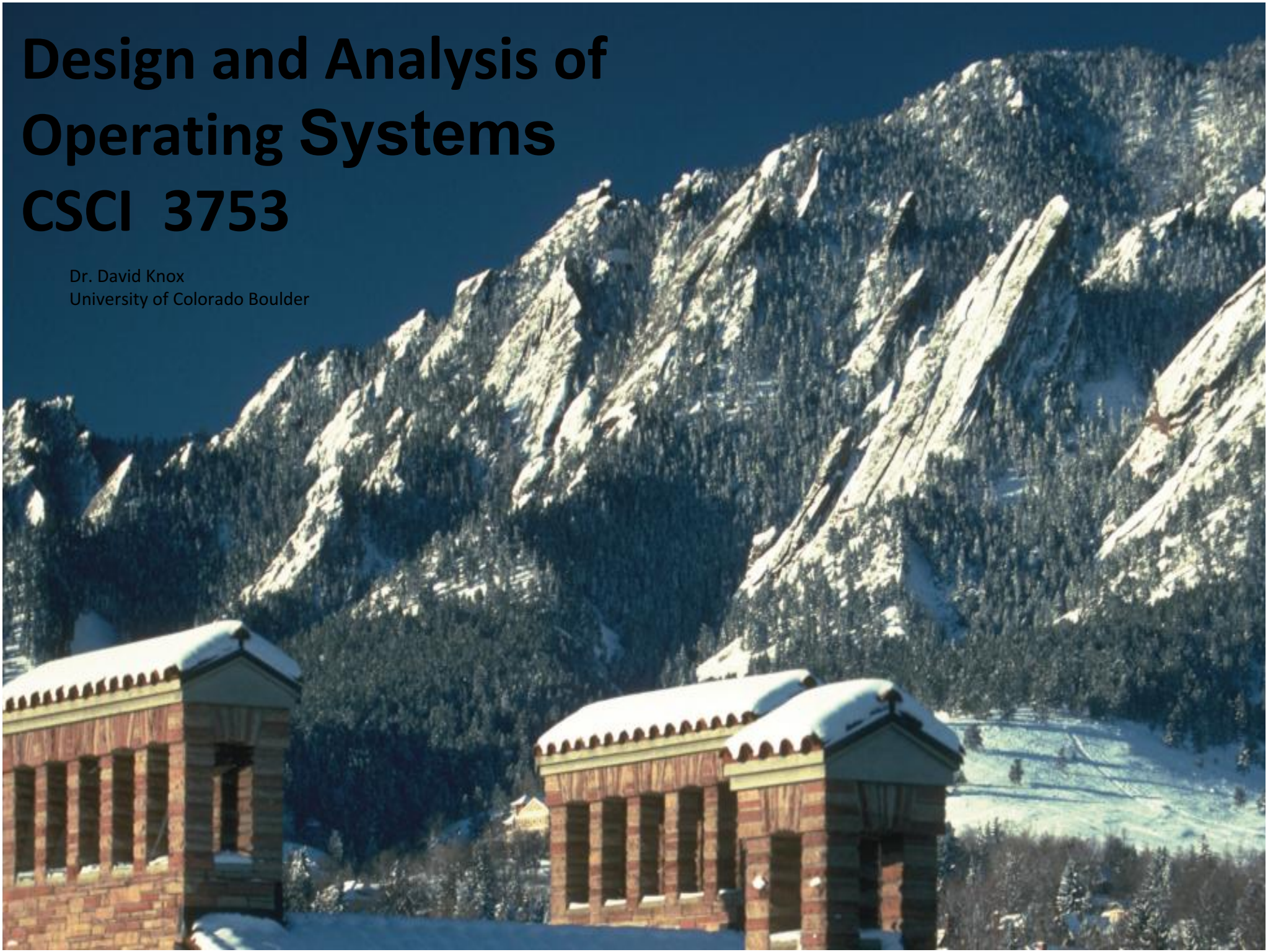
Design and Analysis of Operating Systems CSCI 3753

Dr. David Knox
University of Colorado Boulder



Department of Computer Science
UNIVERSITY OF COLORADO **BOULDER**

Design and Analysis of Operating Systems CSCI 3753

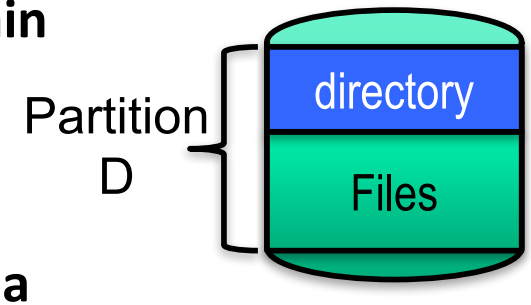
Virtual File System

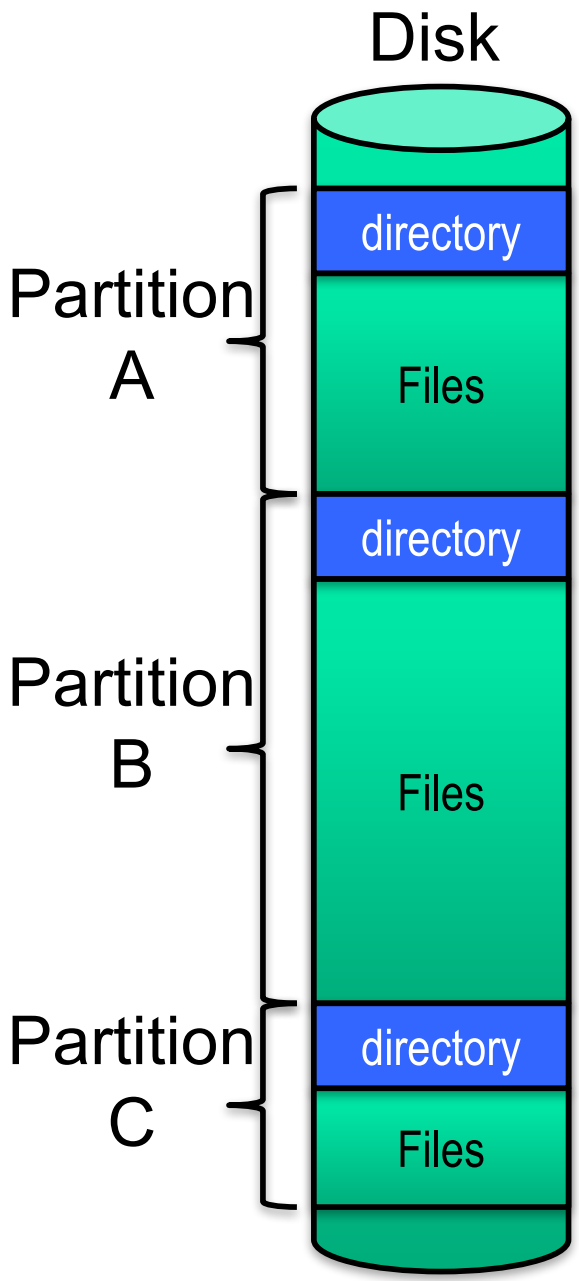
Dr. David Knox
University of
Colorado Boulder

Virtual File System

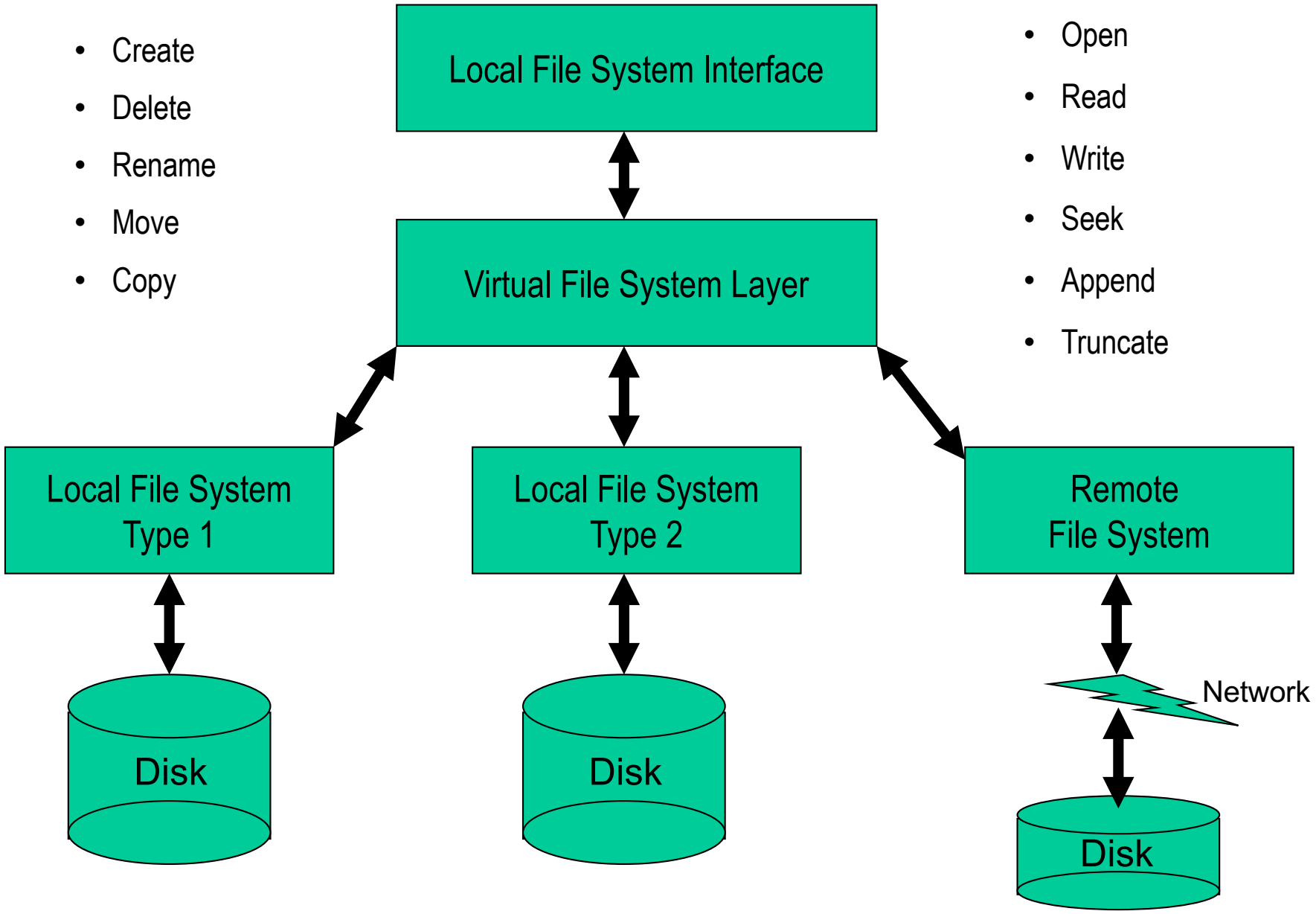
Multiple File Systems

- **A typical disk may have multiple partitions**
  - Each partition may contain a separate file system, and possibly OS as well
  - Each file system will have its own directory structure to keep track of its files
  - A file system may also span multiple disks (e.g. RAID, not shown)
- **Other I/O devices may contain their own file systems**
  - e.g. a USB flash drive
- **Want to share these files on a single computer's file system**



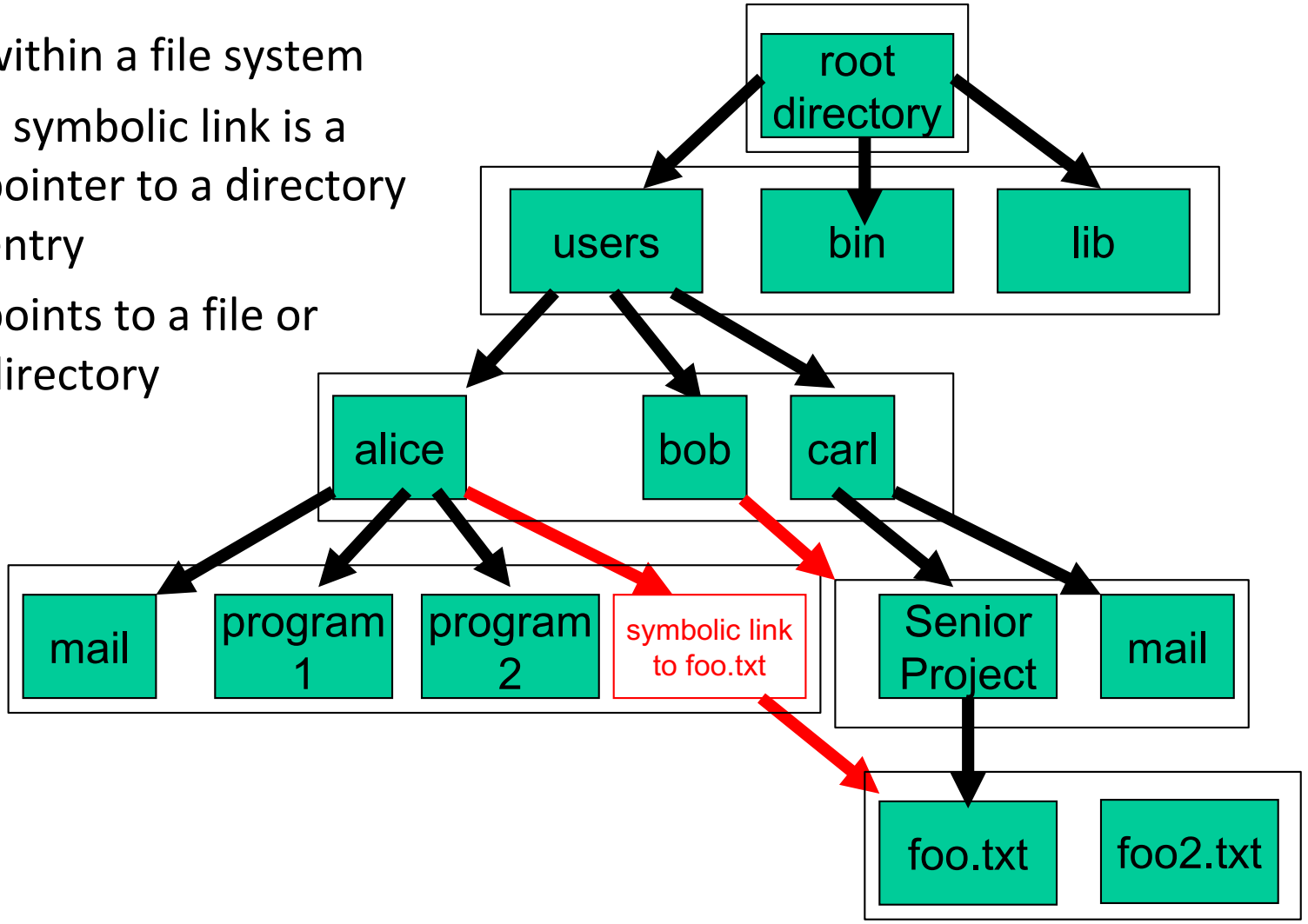


Virtual File Systems



Sharing Directories & Files

- ***symbolic links***
  - within a file system
  - a symbolic link is a pointer to a directory entry
  - points to a file or directory



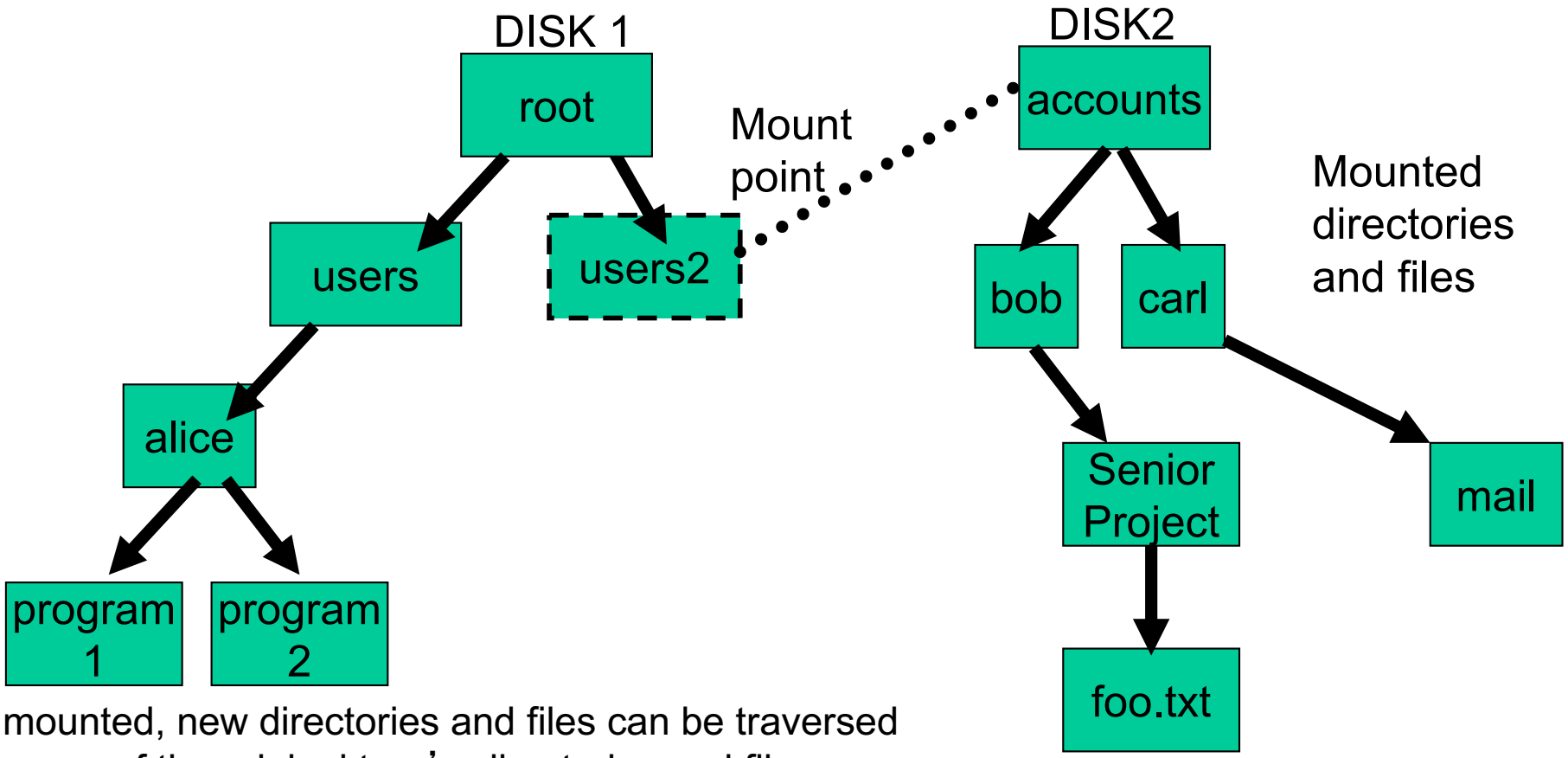
Sharing Directories & Files

- ***symbolic links***
  - within a file system
  - a symbolic link is a pointer to a directory entry
  - points to a file or directory

- **Files may be stored on different disks**
  - or different partitions within a disk
  - or on removable media
  - need access to the share files
  - should be within the same directory structure

Mounting File Systems

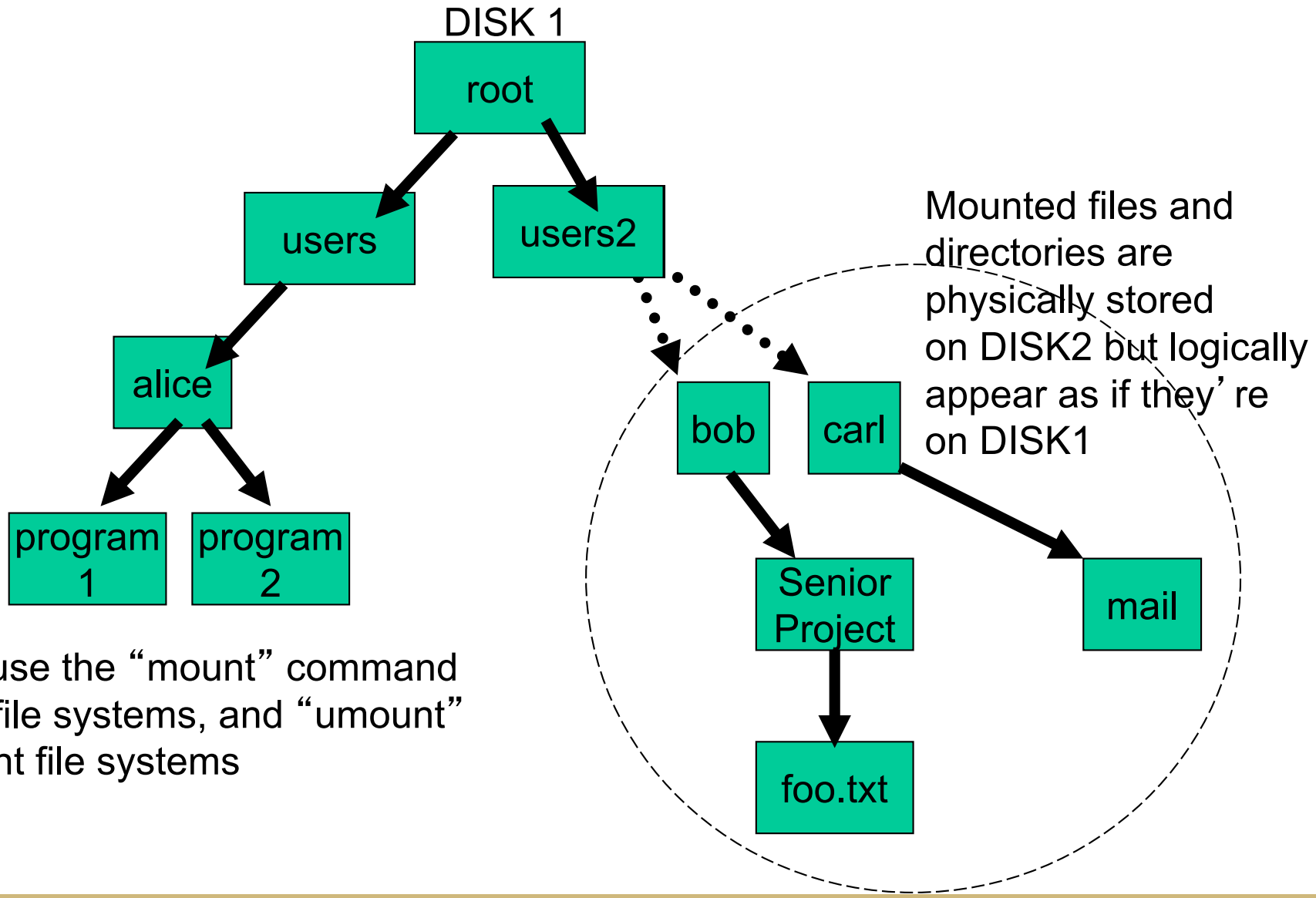
- **Want to share files within the same directory structure though some files may be stored on different disks, or different partitions within a disk**
  - *Mount* these new file systems so they appear within your current directory structure



Once mounted, new directories and files can be traversed just like any of the original tree's directories and files

Mounting File Systems

Final result of file mounting



In Linux, use the "mount" command to mount file systems, and "umount" to unmount file systems

Mounting File Systems

- **Example: to mount a remote directory**
  - mount /xfs filesystem at home.colorado.EDU
  - as a local directory /xfs
  - type:

```
mkdir  /xfs
/bin/mount  home.colorado.edu:/vol/xfs  /xfs
```

- **When a file system is no longer needed**
  - unmount the file system

Mounting File Systems

- **Ideally, you can mount the new file system (e.g. USB stick) anywhere within the current directory tree**
  - Unix follows this flexible approach.
    - The Unix file manager keeps track of what file systems are mounted in which directory by setting a flag in the in-memory copy of the inode for that directory. The flag indicates that the directory is a mount point.
    - A field then points to an entry in the mount table, indicating which device is mounted there.
    - The mount table entry contains a pointer to the disk location of the mounted file system.

Mounting File Systems

- **Ideally, you can mount the new file system (e.g. USB stick) anywhere within the current directory tree (cont.)**
  - Windows mounts a new device containing a file system
    - At the top level, e.g. D:\ or F:\
    - Later versions also allow mounting anywhere
  - Mac OS mounts a new device with a file system At the root level
    - Adds a folder icon on the screen

Virtual File Systems

- **Mounted file system could be of a different type than the current OS file system**
- **How does the OS manage this heterogeneity?**
  - Implement a *Virtual File System* (VFS) layer that abstracts file representation and manipulation
  - VFS layer specifies an abstract model of a file and directory and abstract operations on files and directories
- **The VFS translates abstract operations to/from the specific language of the mounted file system**
- **Note, the mounted file system need not be local**
  - Distributed file systems allow file systems to be across networks

Department of Computer Science
UNIVERSITY OF COLORADO **BOULDER**

Design and Analysis of Operating Systems CSCI 3753

Dr. David Knox
University of
Colorado Boulder