



Overview of Files and I/O Systems

- Concept and Vocabulary review





File Concept

- Contiguous **logical** address space
- Types:
 - Data
 - numeric
 - character
 - binary
 - Program





File Structure

- None - sequence of words, bytes
- Simple **record** structure
 - Lines
 - Fixed length
 - Variable length
- Complex Structures
 - Formatted document
 - HTML, XML, ...
 - Relocatable load file
 - Relational DB tables, schema
- Who decides:
 - Operating system
 - Program





File Attributes

- Name – only information kept in human-readable form.
 - different OS use different naming schemes
- Type – needed for systems that support different types.
 - how do you find the type of a file?
- Location – pointer to file location on device.
- Size – current file size.
- Protection – controls who can do reading, writing, executing.
- Time, date, and user identification – data for protection, security, and usage monitoring.
- Information about files are kept in the directory structure, which is maintained on the disk...called **metadata**
 - where can/should metadata be located?





File Operations

- Create
- Write
- Read
- Reposition within file – file **seek**
- Delete
- Truncate
- $\text{Open}(F_i)$ – search the directory structure on disk for entry F_i , and move the content of entry to memory.
- $\text{Close}(F_i)$ – move the content of entry F_i in memory to directory structure on disk.





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 com- pressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information





Access Methods

- Sequential Access

-
-
-
-
-
-

- read next
- write next
- reset
- no read after last write
(rewrite)

- Direct Access

-
-
-
-
-
-
-
-

- read n
- write n
- position to n
- read next
- write next
- rewrite n

n = relative block number





Information in a Device Directory

- Name
- Type
- Address
- Current length
- Maximum length
- Date last accessed (for archival)
- Date last updated (for dump)
- Owner ID (who pays)
- Protection information (ACLs, TSol Labels)





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

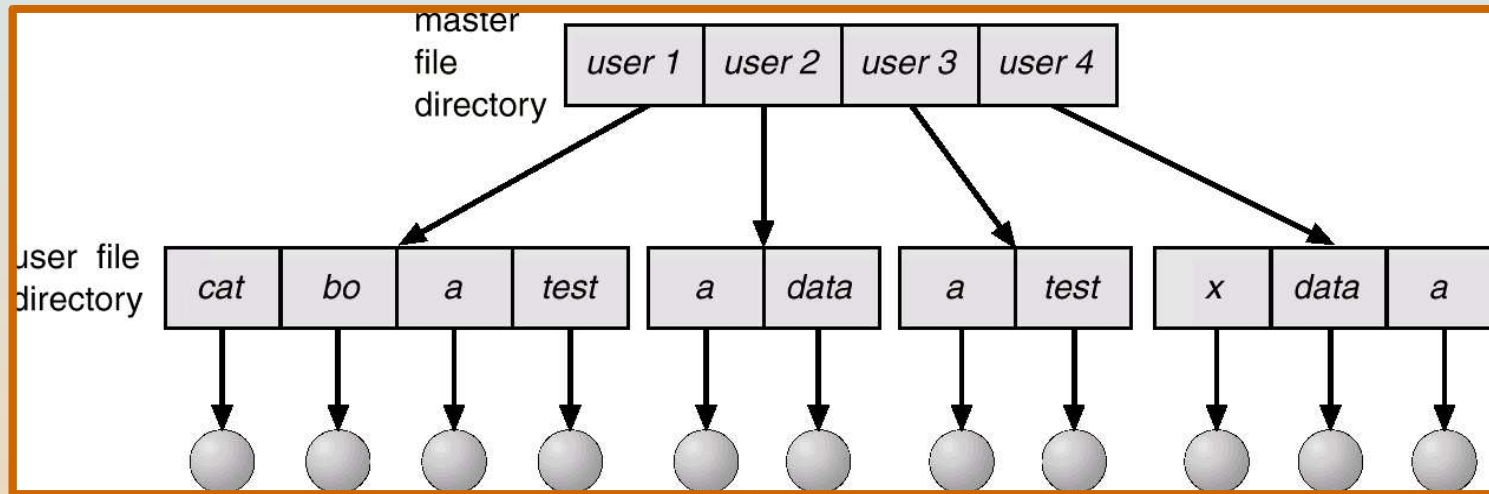
- Efficiency – locating a file quickly.
- Naming – convenient to users.
 - Two users can have same name for different files.
 - The same file can have several different names.
 - using links and symbolic links
- Grouping – logical grouping of files by properties, (e.g., all Java programs, all games,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





Two-Level Directory

- Separate directory for each user.

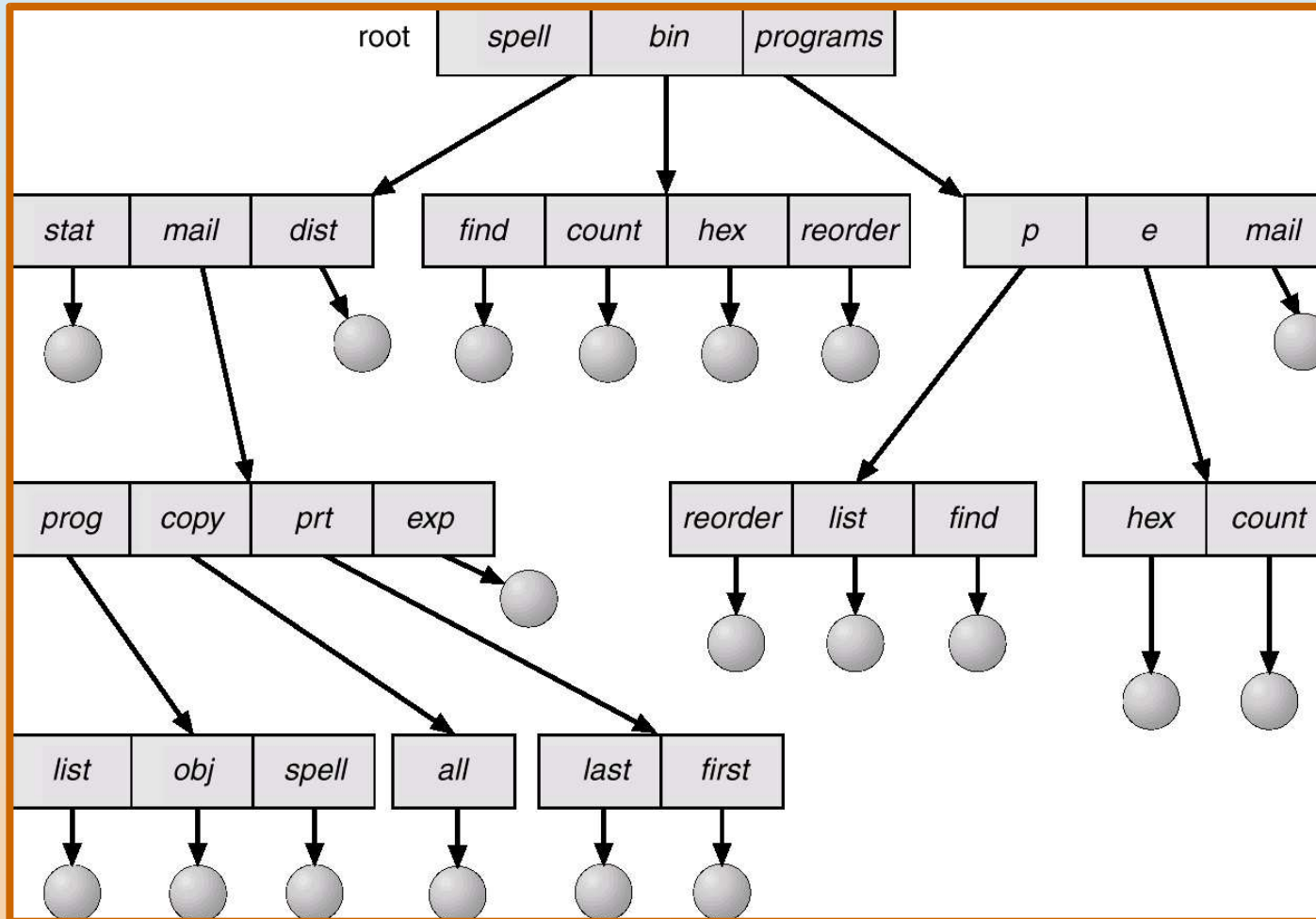


- **Path** name
- Can have the same file name for different user
- Efficient searching
- No grouping capability





Tree-Structured Directories





Tree-Structured Directories (Cont.)

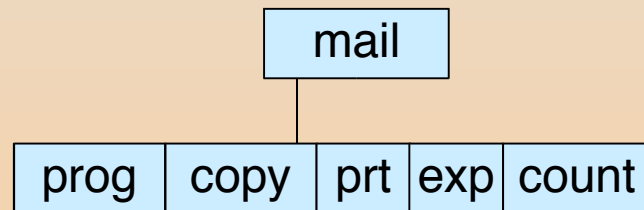
- Efficient searching
- Grouping Capability
- Current directory (working directory)
 - `cd /spell/mail/prog`





Tree-Structured Directories (Cont.)

- **Absolute** or **relative** path name
- Creating a new file is done in current directory.
- Delete a file
 - `rm <file-name>`
- Creating a new subdirectory is done in current directory.
 - `mkdir <dir-name>`
 - Example: if in current directory `/mail`
 - `mkdir count`



Deleting “mail” = deleting the entire subtree rooted by “mail”.





File Sharing

- Sharing of files on multi-user systems is desirable.
- Sharing may be done through a protection scheme.
- On **distributed** systems, files may be shared across a network.
- Network File System (**NFS**) is a common distributed file-sharing method.
 - **invented by...?**





Protection

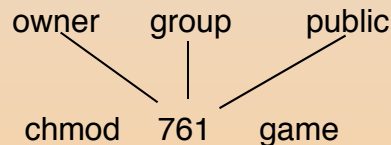
- File owner/creator should be able to control:
 - **what** can be done
 - by **whom**
- Types of access
 - Read
 - Write
 - Execute
 - Append
 - Delete
 - List





Access Lists and Groups

- Mode of access: **read**, **write**, **execute (cd)**
- Three classes of users
- | | | | |
|------------------|---|---|-------|
| | | | RWX |
| a) owner access | 7 | □ | 1 1 1 |
| | | | RWX |
| b) group access | 6 | □ | 1 1 0 |
| | | | RWX |
| c) public access | 1 | □ | 0 0 1 |
- Ask manager to create a group (unique name), say G, and add some users to the group.
- For a particular file (say game) or subdirectory, define an appropriate access.

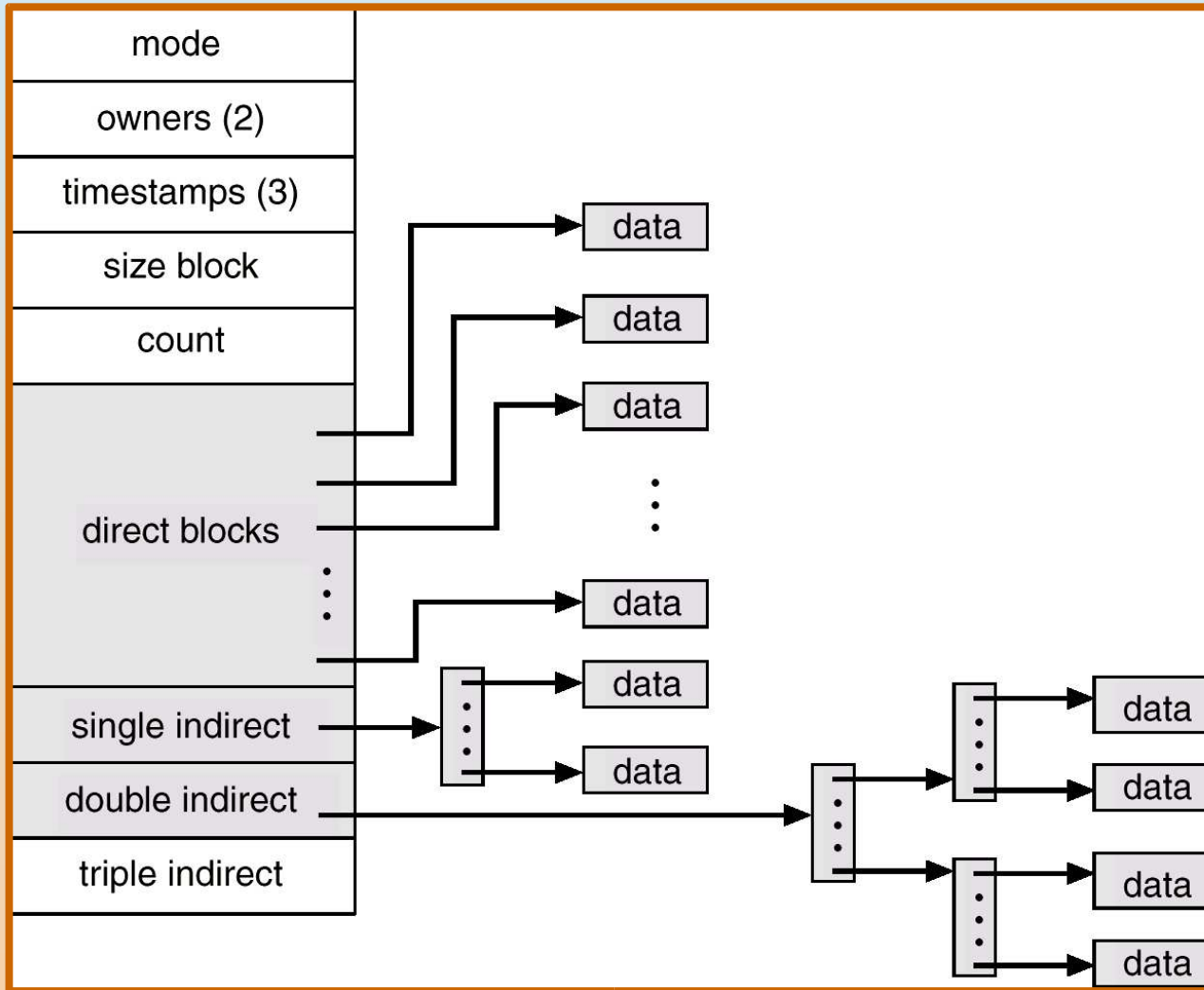


Attach a group to a file

chgrp G game

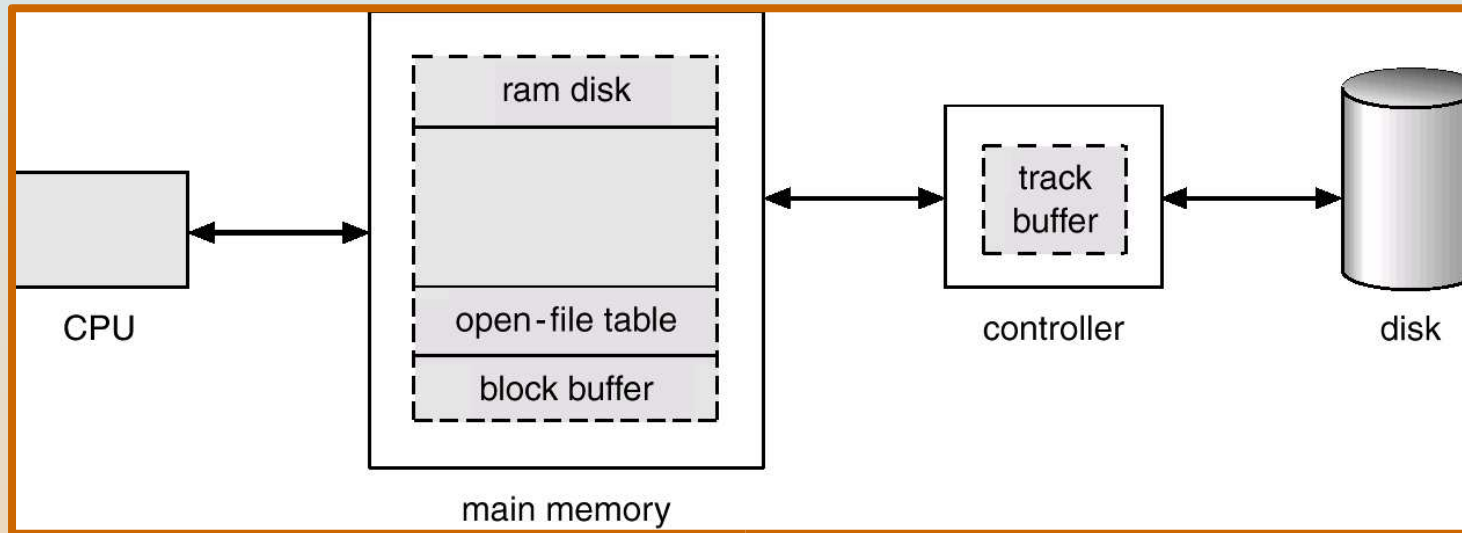


Combined Scheme: UNIX (4K bytes per block)





Various Disk-Caching Locations





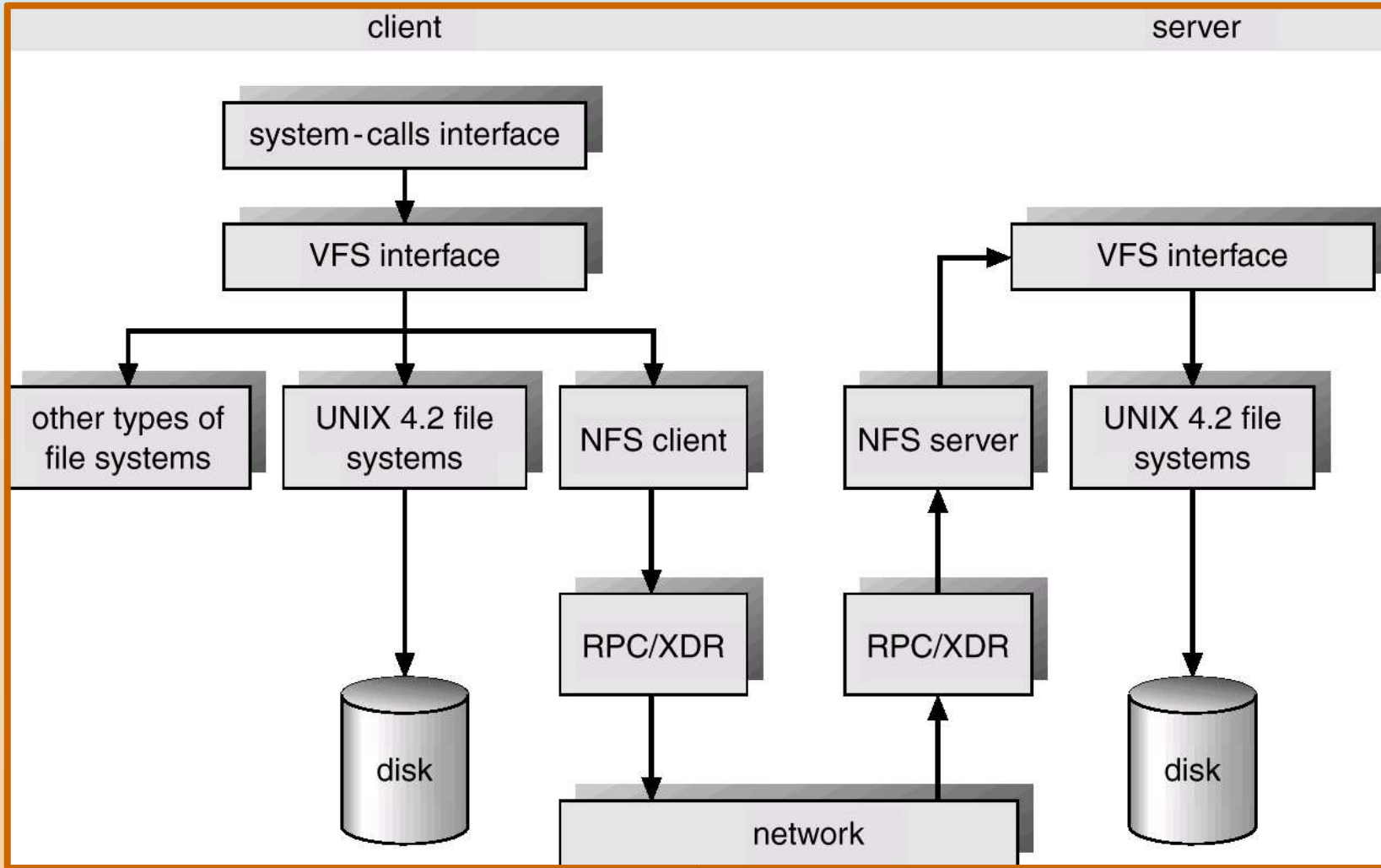
Log Structured File Systems

- **Log** structured (or **journaling**) file systems record each update to the file system as a transaction.
-
- All transactions are written to a log. A transaction is considered committed once it is written to the log. However, the file system may not yet be updated.
-
- The transactions in the log are asynchronously written to the file system. When the file system is modified, the transaction is removed from the log.
-
- If the file system crashes, all remaining transactions in the log must still be performed.





Schematic View of NFS Architecture





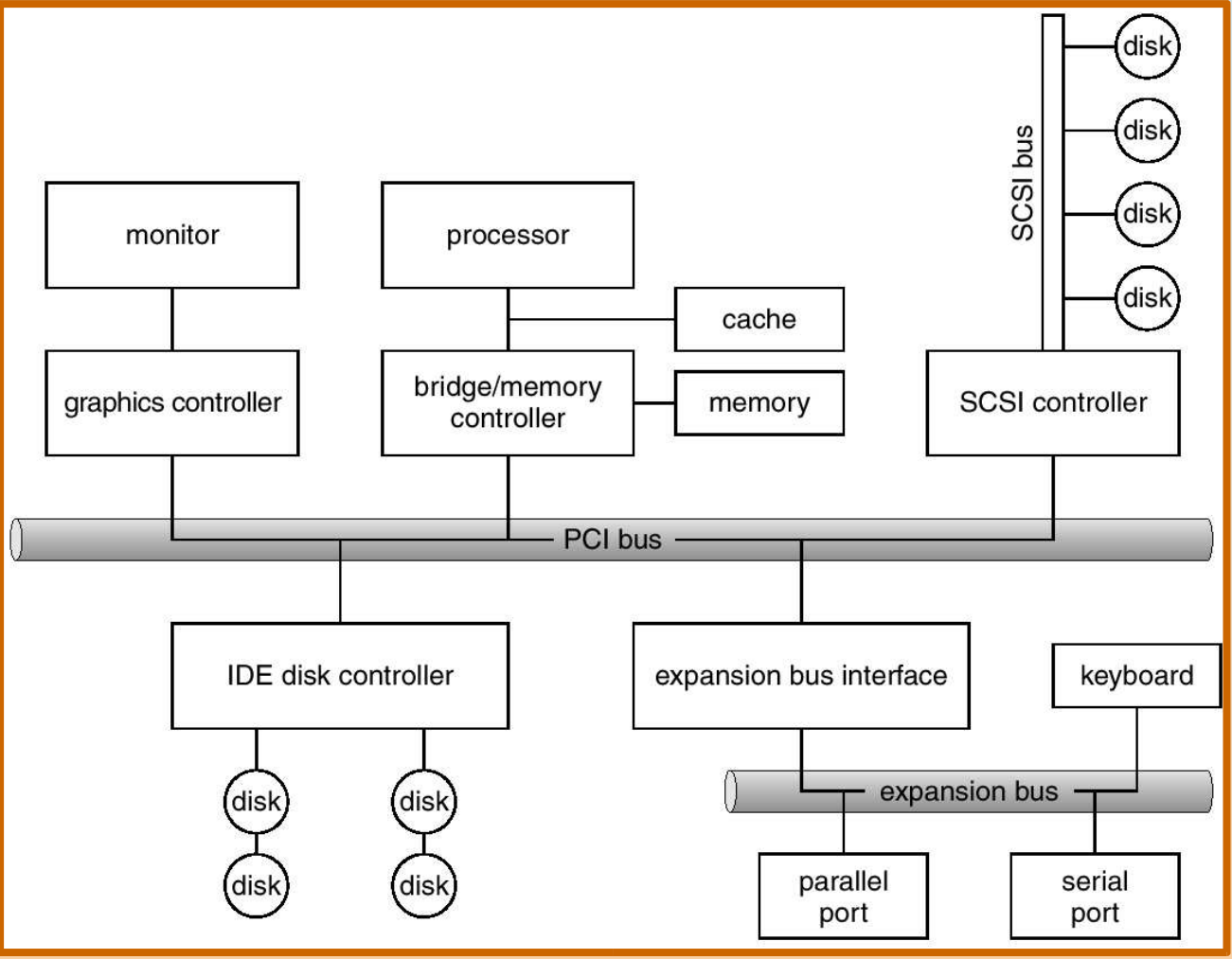
I/O Hardware

- Incredible variety of I/O devices
- Common concepts
 - Port
 - Bus (daisy chain or shared direct access)
 - Controller (host adapter)
 - **what kinds of HBAs are there?**
- I/O instructions control devices
- Devices have addresses, used by
 - Direct I/O instructions
 - Memory-mapped I/O





A Typical PC Bus Structure





Characteristics of I/O Devices

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read&write	CD-ROM graphics controller disk





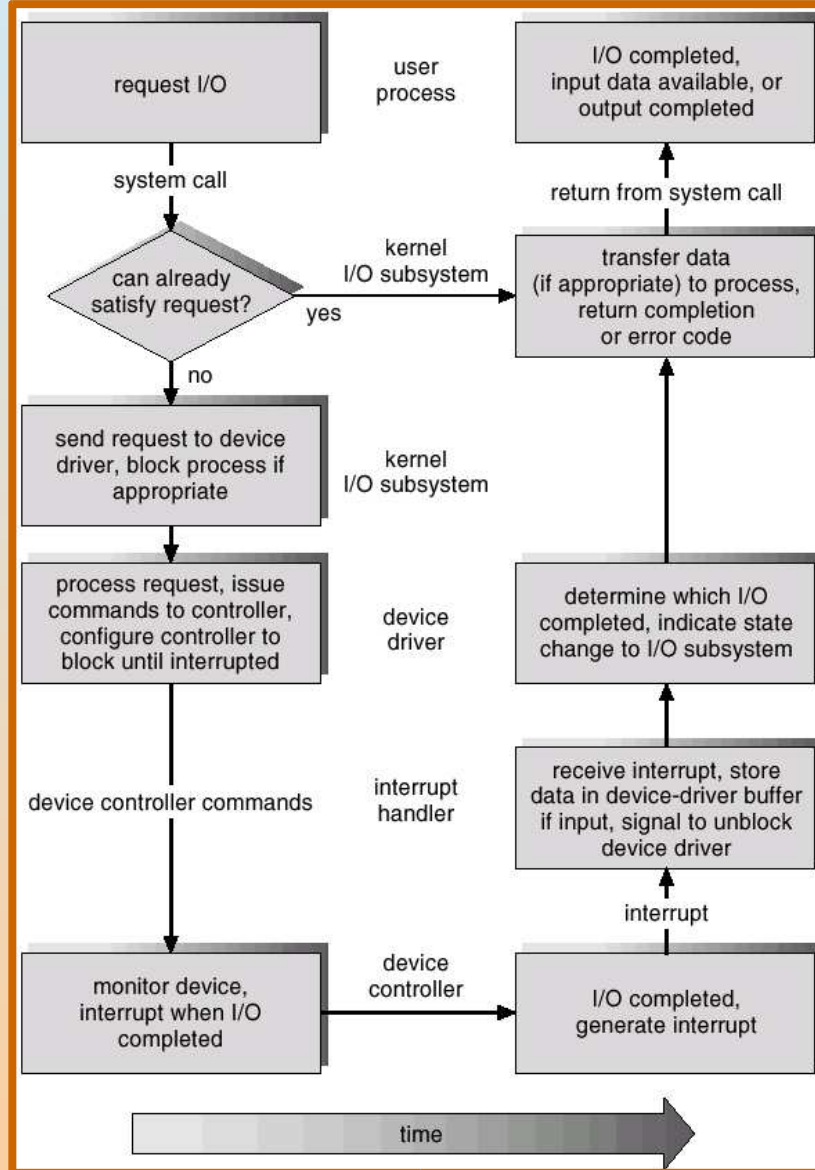
Blocking and Nonblocking I/O

- Blocking - process **suspended** until I/O completed
 - Easy to use and understand
 - Insufficient for some needs
- Nonblocking - I/O call returns as much as available
 - User interface, data copy (buffered I/O)
 - Implemented via **multi-threading**
 - Returns quickly with count of bytes read or written
- Asynchronous - process runs while I/O executes
 - Difficult to use
 - **WHY?**
 - I/O subsystem signals process when I/O completed





Life Cycle of An I/O Request





Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of **logical blocks**, where the logical block is the smallest unit of transfer.
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially.
 - Sector 0 is the first sector of the first track on the outermost cylinder.
 - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost.
- **What does disk structure look like?**
 - **what are important performance characteristics?**





Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of **logical blocks**, where the logical block is the smallest unit of transfer.
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially.
 - Sector 0 is the first sector of the first track on the outermost cylinder.
 - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost.
- **What does disk structure look like?**
 - **what are important performance characteristics?**
 - RPM
 - Seek time, settle time
 - rotational delay
 - “smart controller”
 - = transfer rate (MB/s)





RAID Structure

- **RAID** – multiple disk drives provides reliability via redundancy.
 - **Redundant Array of Inexpensive Disks**
- RAID is arranged into six different levels.





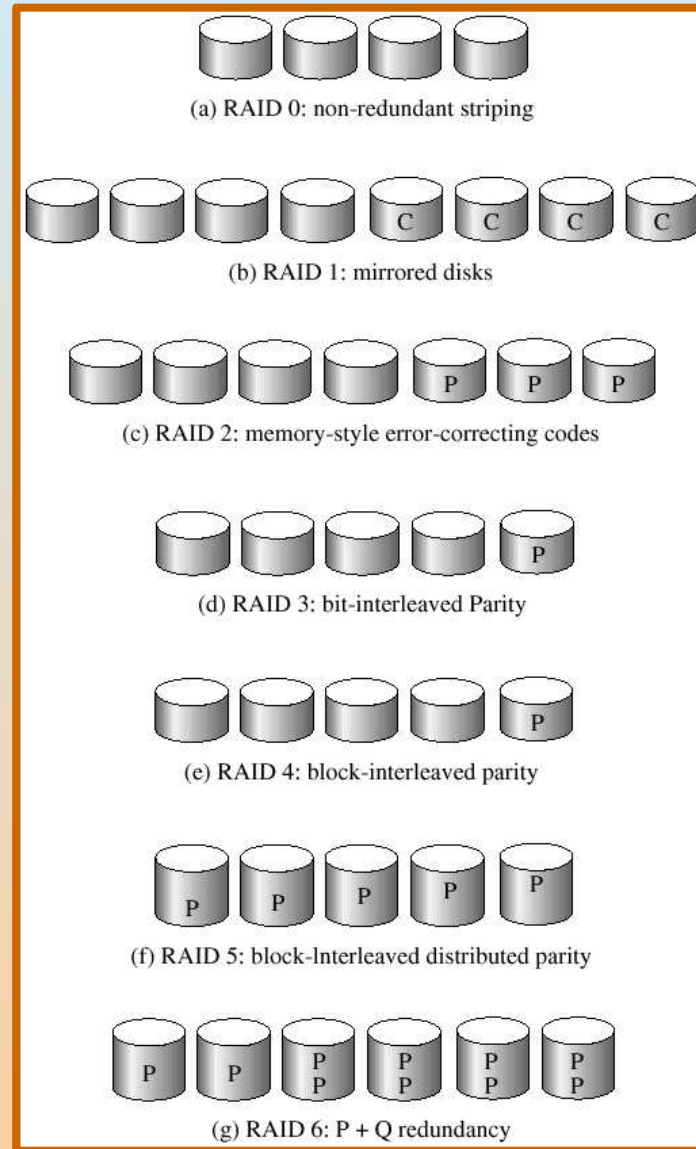
RAID (cont)

- Several improvements in disk-use techniques involve the use of **multiple disks working cooperatively**.
- Disk **striping** uses a group of disks as one storage unit.
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data.
 - **Mirroring** or shadowing keeps duplicate of each disk.
 - Block interleaved **parity** uses much less redundancy.



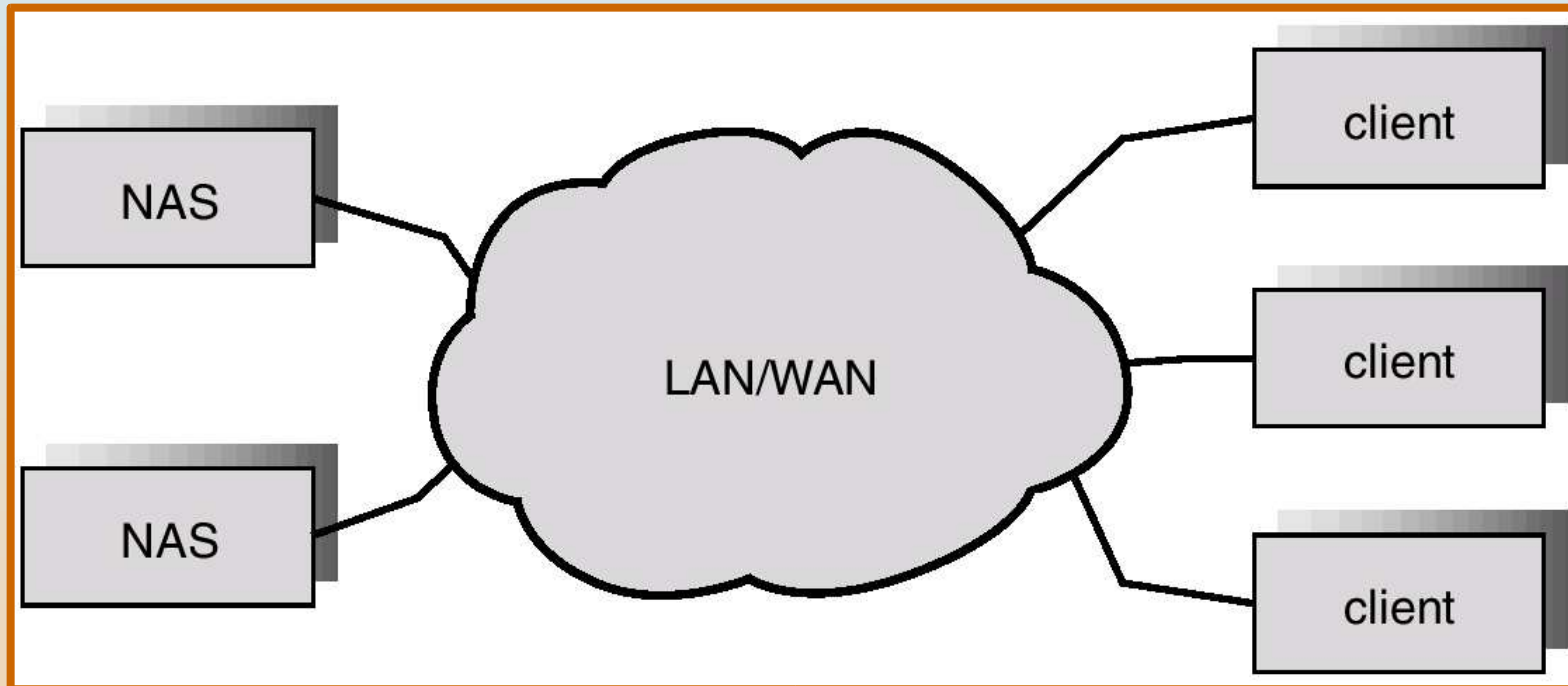


RAID Levels



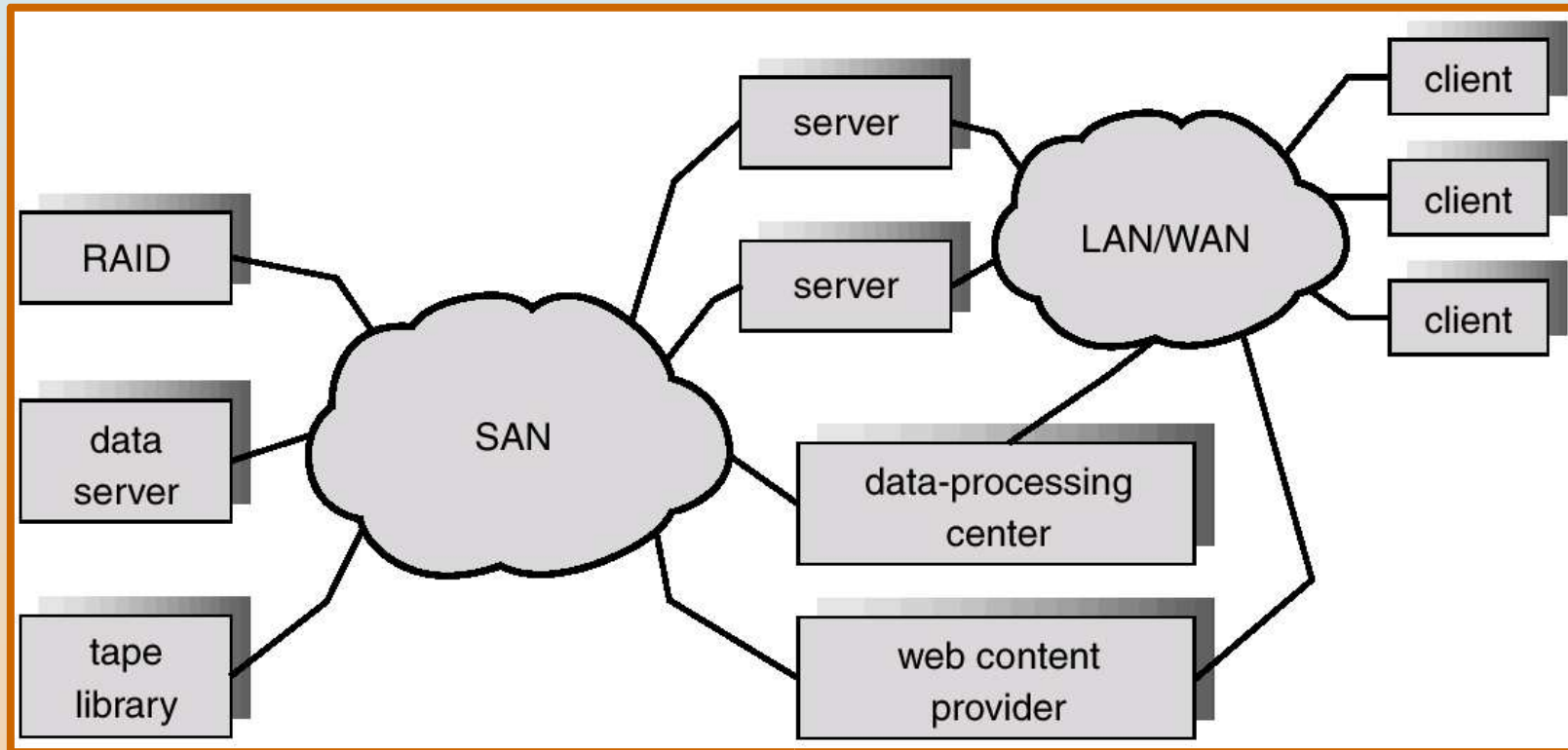


Network-Attached Storage





Storage-Area Network





Operating System Issues

- Major OS jobs are to manage physical devices and to present a virtual machine abstraction to applications
- For hard disks, the OS provides two abstraction:
 - **Raw** device – an array of data blocks.
 - **File system** – the OS queues and schedules the interleaved requests from several applications.





Hierarchical Storage Management (HSM)

- A **hierarchical storage system** extends the storage hierarchy beyond primary memory and secondary storage to incorporate tertiary storage — usually implemented as a jukebox of tapes or removable disks.
- Usually incorporate tertiary storage by extending the file system.
 - Small and frequently used files remain on disk.
 - Large, old, inactive files are archived to the jukebox.
- HSM is usually found in supercomputing centers and other large installations that have enormous volumes of data.





Price per Megabyte of Magnetic Hard Disk, From 1981 to 2000

