CS 3305A

File Systems
Network File Systems (NFS)

Lecture 21 & 22

Files

- Collection of related information recorded on secondary storage
  - e.g., helloworld.c, resume.doc
- Can contain programs (source, binary) or data
- Files have attributes:
  - Name, type, location, size, protection, creation time etc

File Naming

- Files are named
- Even though files are just a sequence of bytes, programs can impose structure on them
  - Files with a certain standard structure imposed can be identified using an extension to their name
  - Application programs may look for specific file extensions to indicate the file's type
  - But as far as the operating system is concerned its just a sequence of bytes

File Types – Name, Extension

file type	usual extension	function
executable	exe, com, bin or none	ready-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, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	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, mp3, avi	binary file containing audio or A/V information

File Attributes

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

Possible file attributes

Basic File Operations

- Operations
  - Create
  - Write
  - Read
  - Delete
  - Others
    - A reposition within the file, append, rename
- For write/read operations, the operating system needs to keep a file position pointer for each process

Directories

- OS File systems use directories to keep track of files

- Directory operations
  - File search
  - File creation
  - File deletion
  - Directory listing
  - File renaming

File System

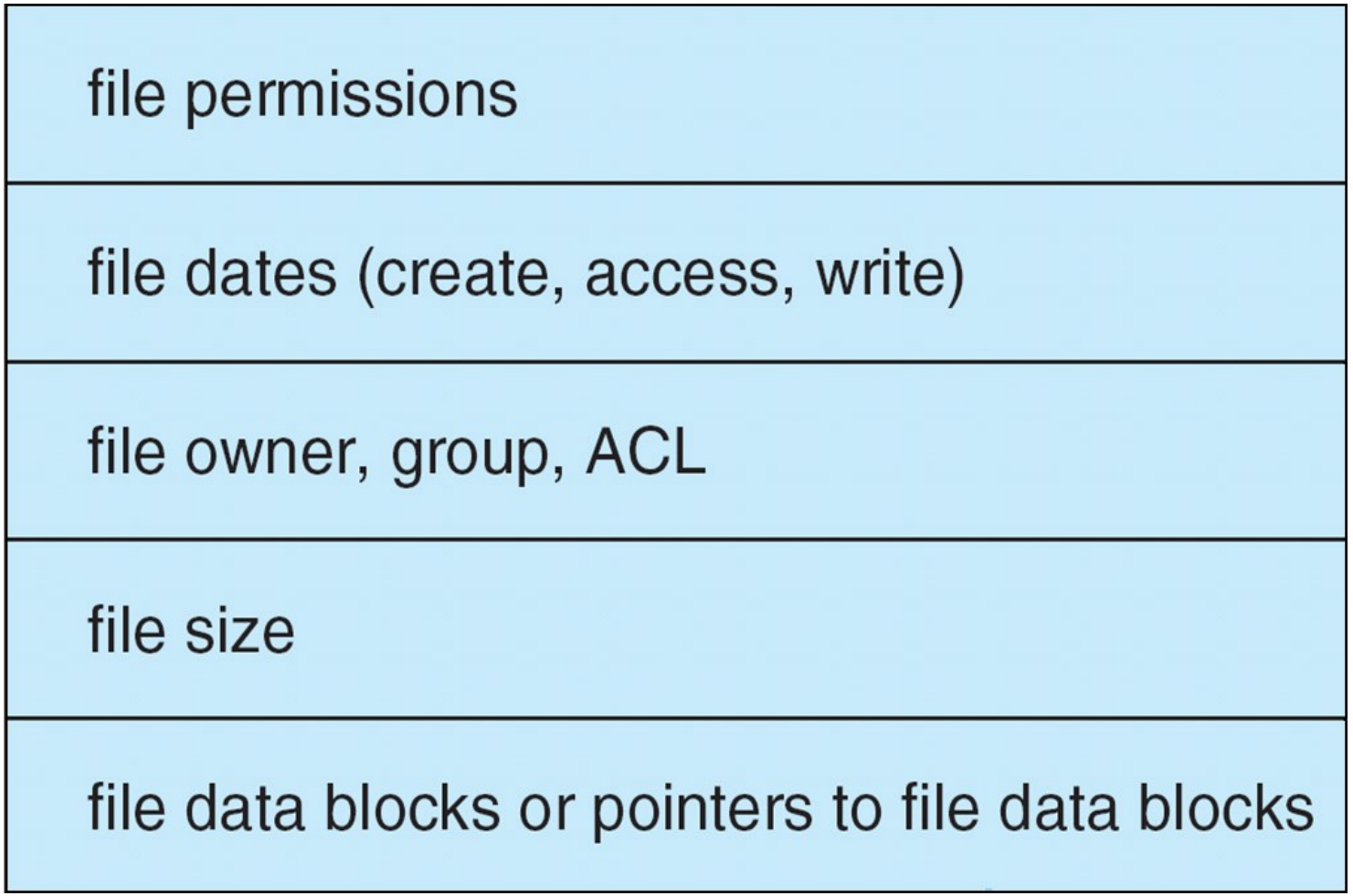
- There are many file systems
- Unix uses the UNIX file system (UFS)
- Windows NT File System FAT, FAT32 and NTFS
- Linux file system is known as the extended file system with versions denoted by ext2 and ext3
- Google created its own file system to meet its needs.

File System Structures

File Control Block

- Information about a file may be maintained in File control block (FCB):
- One FCB per file
- A FCB is associated with a unique identifier
- Consists of details about a file

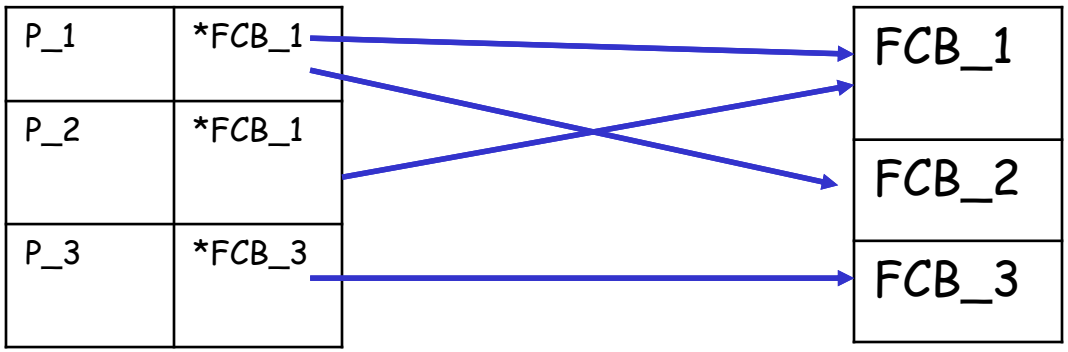
A Typical File Control Block



File System Structures

- System-wide open file table: Contains a copy of the FCB of each open file

- Per-process open-file table: Contains a pointer to the appropriate entry in the system-wide open-file table



Per-process open file Table

System-wide open file Table

File Creation

- To create a new file, an application uses the file system
- Creation requires the allocation of a new File Control Block (FCB)

File Opening

- What happens when an open(filename) is called in an application program?

- The OS searches the system-wide open-file table to see if the file is in use by another process

File Opening

- If the file is in use by another process?
  - A per-process open-file table entry is created pointing to the existing system-wide open-file table entry
- If the file is not in use by another process?
  - Search the directory structure for the given file name
  - When found, file is opened, and the FCB is copied into a system-wide open-table
  - A per-process open-file table entry is created pointing to the existing system-wide open-file table entry

File Opening

- When a process closes the file, the per-process table entry is removed
- A count associated with the system-wide entry's open count is decremented
- Now let us look a file-system layout

Allocation Methods

File System Layout

- File systems usually are stored on disks
- Most disks can be divided up into partitions
  - Independent file systems (for different operating systems) on each partition
- Sector 0 of the disk is the Master Boot Record (MBR)
- The end of the MBR contains the partition table
  - Gives the start and end of each partition

Allocation Methods

- Many files are stored on the same disk
- The main problem is how to allocate space so that
  - Disk space is used effectively and
  - Files can be access quickly

Allocation Methods

- Files are sequences of bytes
- Disks are arrays of sectors (512 bytes)
  - Granularity of disk I/O is sectors
  - File data must be stored in sectors

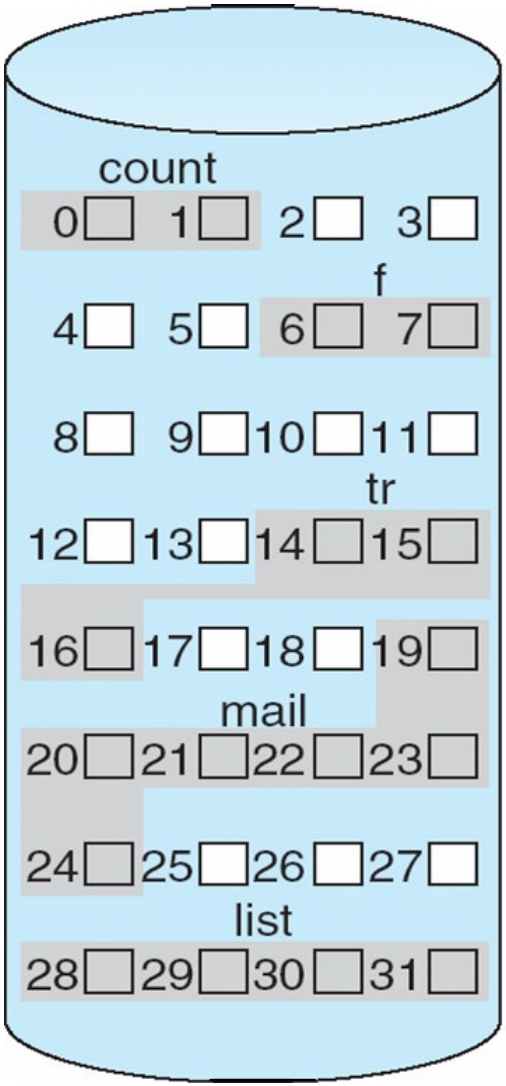
Allocation Methods

- File systems may also define a block size
  - Block size usually consists of a number of sectors
  - Contiguous sectors are allocated to a block
- File systems view the disk as an array of blocks
  - Must allocate blocks to file
  - Must manage free space on disk

Allocation Methods

- Approaches to allocating blocks to a file
  - Contiguous Allocation
  - Linked List Allocation
  - Indexed Allocation

Contiguous Allocation



directory

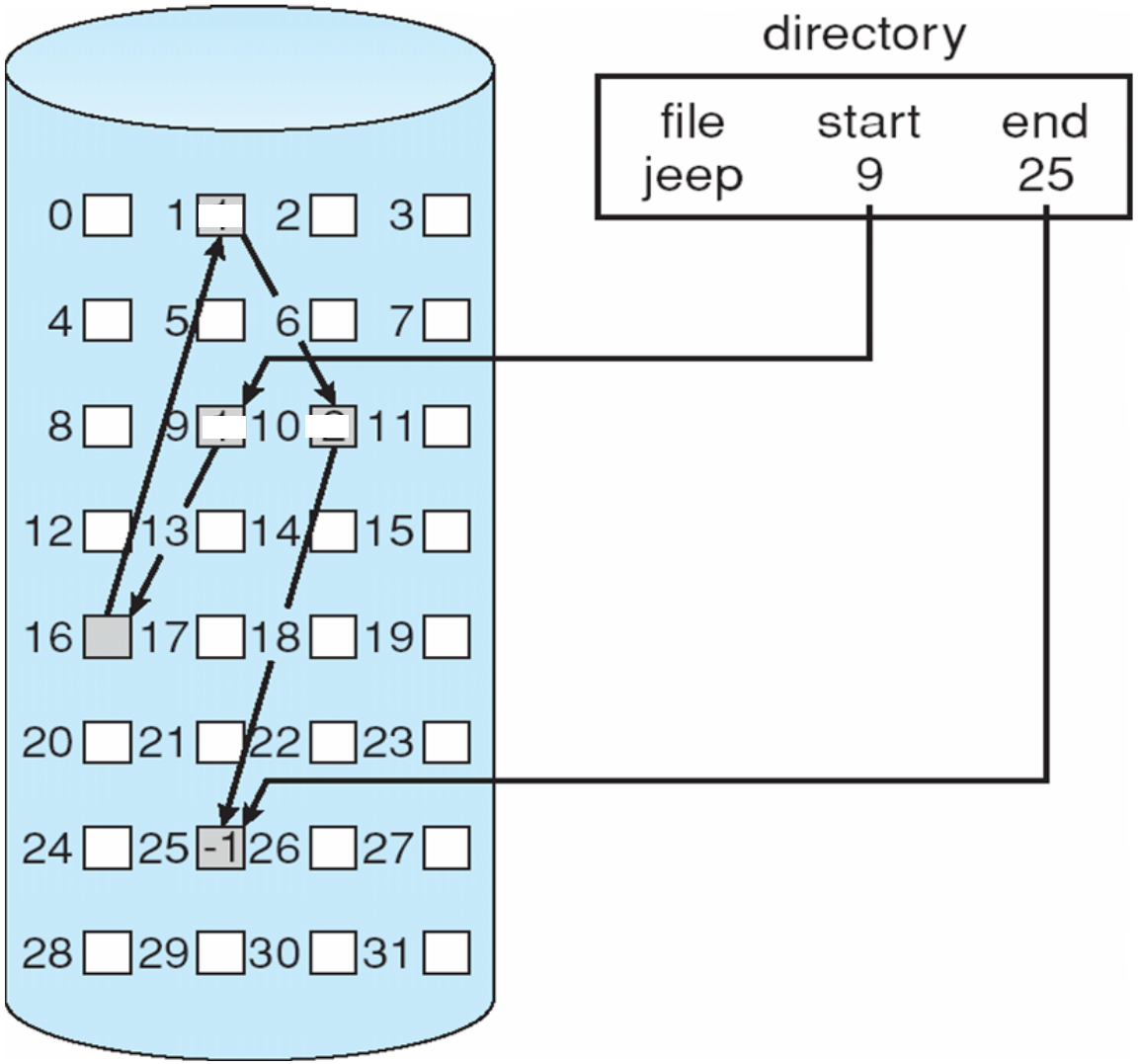
file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

Start and length are stored in the FCB

Contiguous Allocation

- Advantages:
  - Easy to implement
  - Read performance is excellent
- Disadvantages
  - Fragmentation
  - Will need periodic compaction (time-consuming)

Linked List Allocation



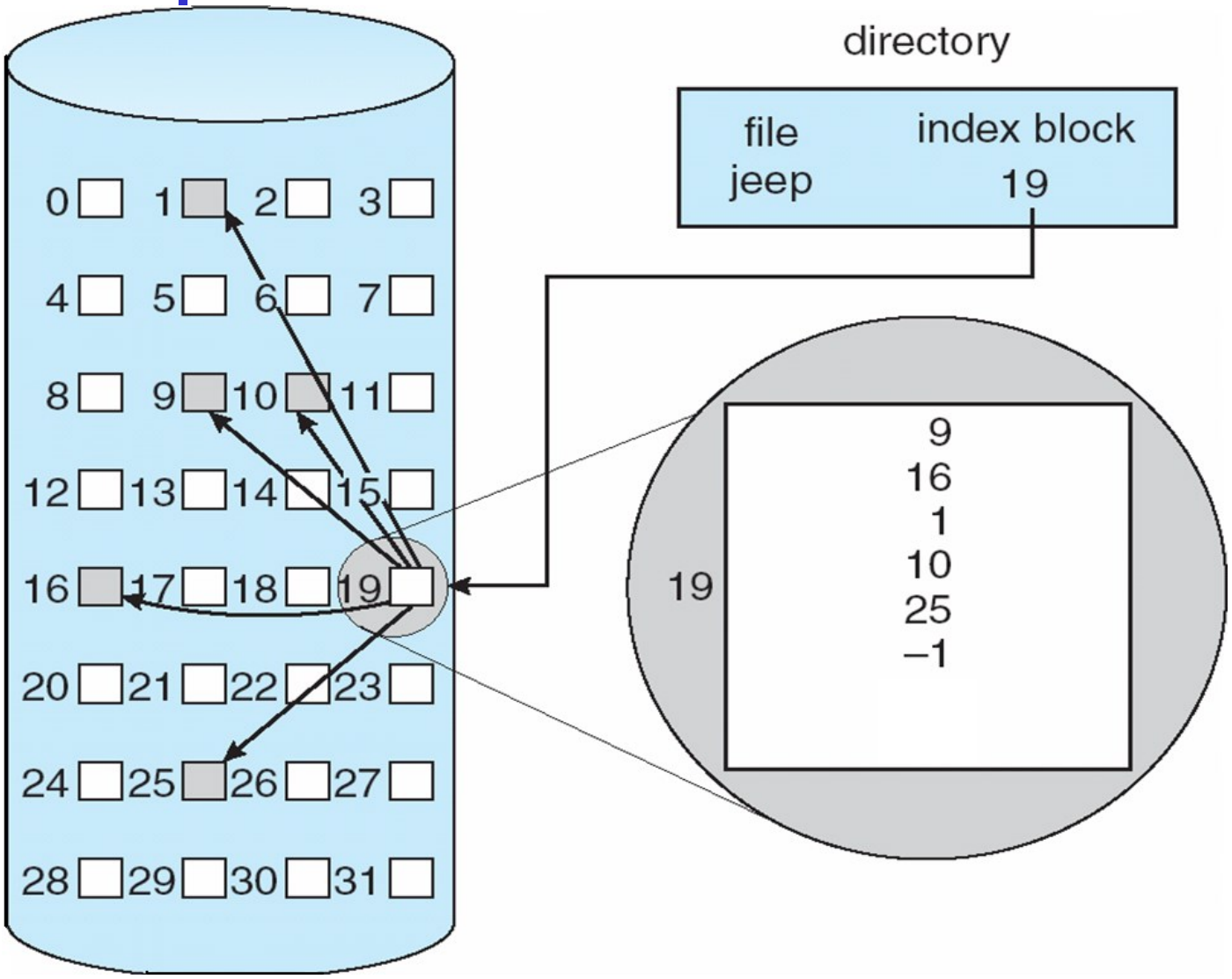
Linked List Allocation

- Advantage
  - No fragmentation (except internal fragmentation)
- Disadvantage
  - Random access is slow

Indexed Allocation

- Indexed allocation brings all pointers together into an index block

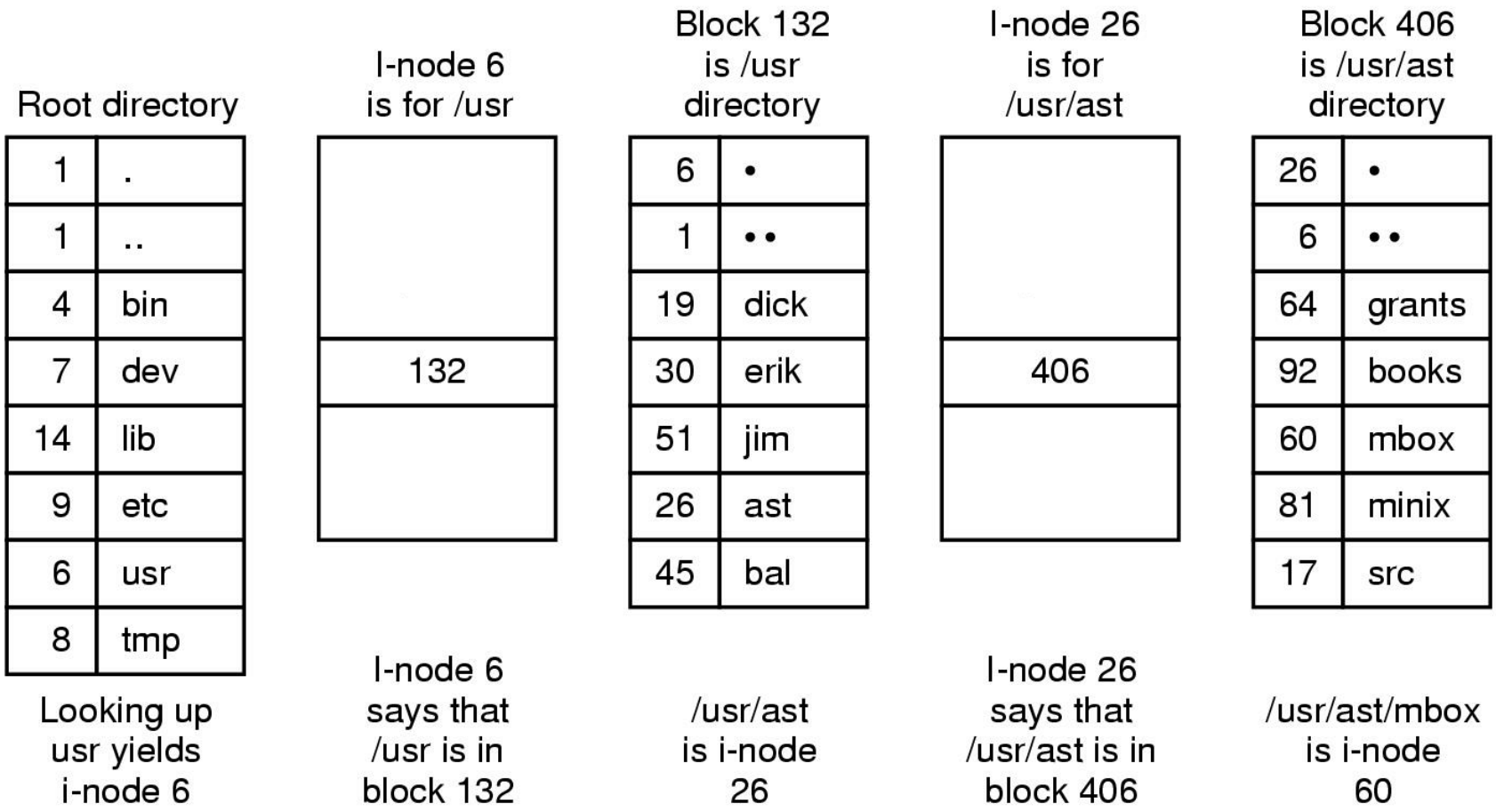
Example of Indexed Allocation



Entry Lookup

- This discussion focuses on Unix-related file systems
- Let's see how this is done for the path name */usr/ast/mbox*

Looking up for an entry



The steps in looking up /usr/ast/mbox

Entry Lookup

- First the system locates the root directory
  - There is information for each file and directory within the root directory
  - A directory entry consists of the file name and i-node number
- The file system looks up the first component of the path, *usr*, to find the i-node for */usr* which is i-node 6
- From this i-node the system locates the directory for */usr/* which is in block 132

Entry Lookup

- The system then searches for *ast* within the */usr* directory which is block 132
- The entry gives the i-node for */usr/ast/*
- From this i-node the system can find the directory itself and lookup *mbox*
  - The i-node for this file is then read into memory and kept there until the file is closed

Network File Systems (NFS)

Network File Systems

- I'm on a Linux machine
- I go to another Linux machine
  - I see the same files in my home directory
  - Why?
- This is because you are using a networked file system (NFS)
- NFS also provides the protocols needed for an integrated view of directories

NFS

- An NFS server runs on a machine which may have large disks and hosting the data / files / applications

- An NFS client runs on client machines and access the files on machines that run NFS servers.

NFS

- Files / Directories can be mounted which means that they can appear in the tree structure
  - The files / directories can be on remote machines
  - The remote machines may use a different OS than the client machine

NFS Structures

- Export List
  - Maintained by server (NFS Server)
  - Information:
    - List of local directories that server exports for mounting
    - Names of machines that are permitted to mount them

NFS Mount Protocol

- Establishes initial logical connection between NFS server and NFS client
- Client sends a mount request message to a server
  - Request includes name of remote directory to be mounted and name of server machine storing it

NFS Mount Protocol

- Following a mount request that conforms to its export list, the server returns a file handle—a key for further accesses
- The mount operation changes only the user's view and does not affect the server side

NFS

- When an application program calls open() to obtain access to a file it is making a call to the NFS client
- If the path refers to a local file, the system uses the computer's standard file system software to access the file
- If the path refers to a remote file, the system uses NFS client software to access the remote file through the NFS server

NFS

- The NFS server handles incoming client requests
- These requests are mapped to local file system operations