File Systems

Operating Systems

1

Long-term Information Storage

1. Must store large amounts of data
2. Information stored must survive the termination of the process using it
3. Multiple processes must be able to access the information concurrently

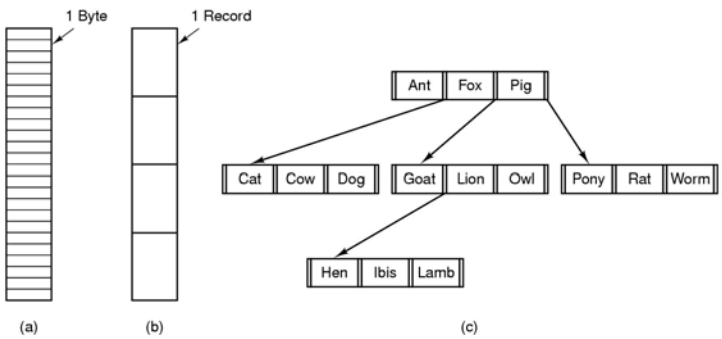
2

File Naming

Extension	Meaning
file.bak	Backup file
file.c	C source program
file.gif	Compuserve Graphical Interchange Format image
file.hlp	Help file
file.html	World Wide Web HyperText Markup Language document
file.jpg	Still picture encoded with the JPEG standard
file.mp3	Music encoded in MPEG layer 3 audio format
file.mpg	Movie encoded with the MPEG standard
file.o	Object file (compiler output, not yet linked)
file.pdf	Portable Document Format file
file.ps	PostScript file
file.tex	Input for the TEX formatting program
file.txt	General text file
file.zip	Compressed archive

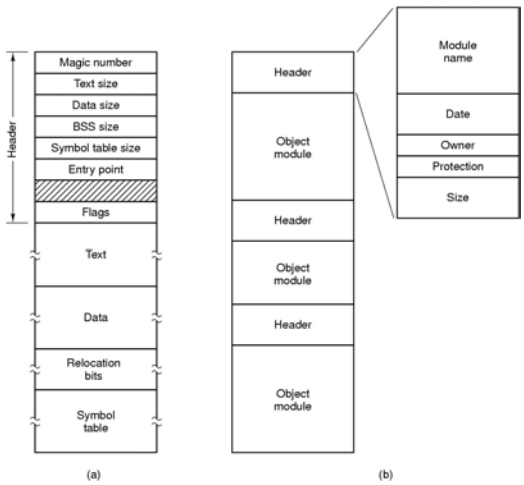
Typical file extensions.

File Structure



- Three kinds of files
  - byte sequence
  - record sequence
  - tree

File Types



(a) An executable file (b) An archive

5

File Access

- Sequential access
  - read all bytes/records from the beginning
  - cannot jump around, could rewind or back up
  - convenient when medium was mag tape
- Random access
  - bytes/records read in any order
  - essential for data base systems
  - read can be …
    - move file marker (seek), then read or …
    - read and then move file marker

6

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

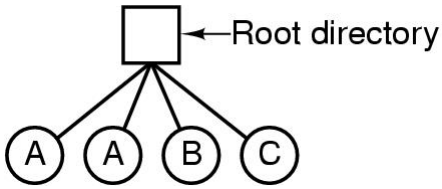
7

File Operations

1. Create
2. Delete
3. Open
4. Close
5. Read
6. Write
7. Append
8. Seek
9. Get attributes
10. Set Attributes
11. Rename

8

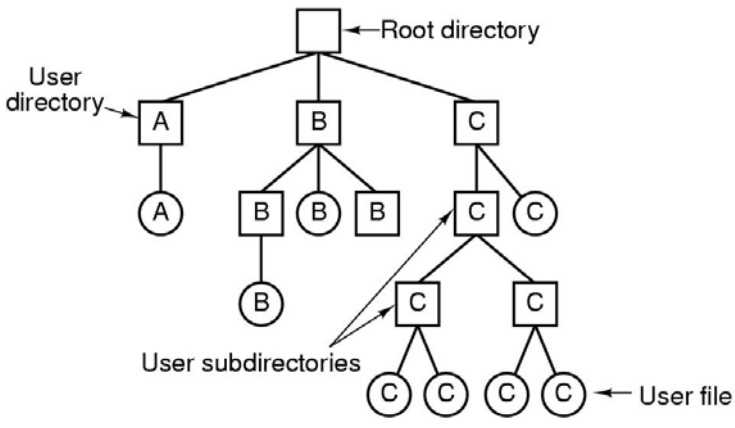
Directories: Single-Level Directory Systems



- A single level directory system
  - contains 4 files
  - owned by 3 different people, A, B, and C

9

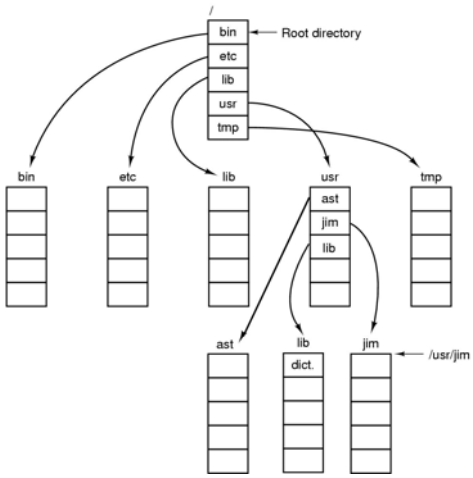
Hierarchical Directory Systems



A hierarchical directory system

10

Path Names



A UNIX directory tree

Directory Operations

1. Create
2. Delete
3. Opendir
4. Closedir
5. Readdir
6. Rename
7. Link
8. Unlink

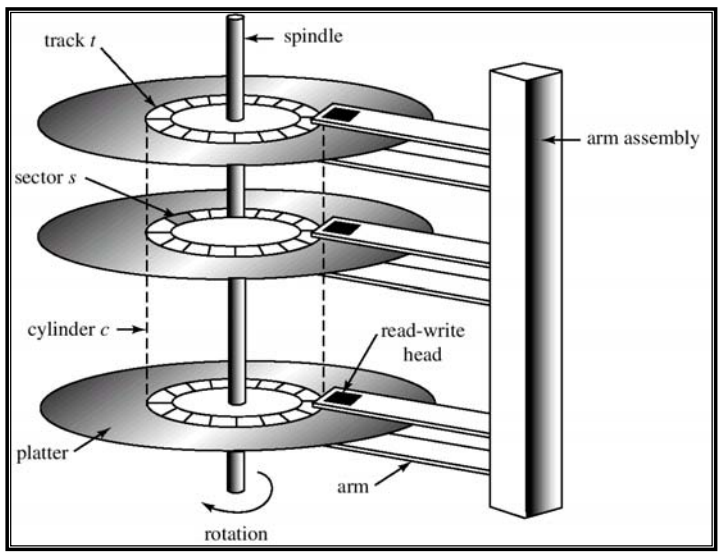
Disks: Disk Hardware

Parameter	IBM 360-KB floppy disk	WD 18300 hard disk
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (avg)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360 KB	18.3 GB
Seek time (adjacent cylinders)	6 msec	0.8 msec
Seek time (average case)	77 msec	6.9 msec
Rotation time	200 msec	8.33 msec
Motor stop/start time	250 msec	20 sec
Time to transfer 1 sector	22 msec	17 µsec

Disk parameters for the original IBM PC floppy disk and a Western Digital WD 18300 hard disk

13

Moving-Head Disk Mechanism



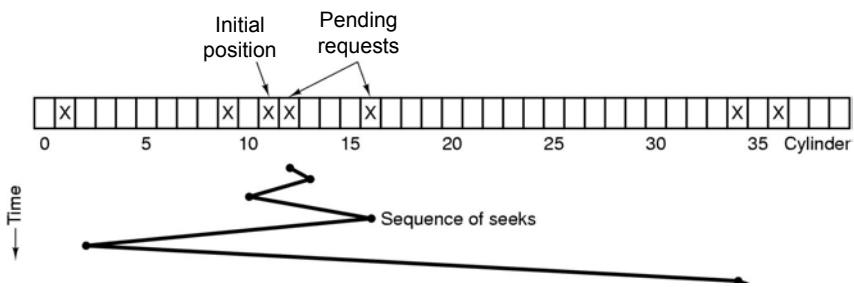
14

Disk Arm Scheduling Algorithms (1)

- Time required to read or write a disk block determined by 3 factors
  1. Seek time
  2. Rotational delay
  3. Actual transfer time
- Seek time and rotational delay dominate
  - Average rotational delay = half the time for one rotation
- Disk scheduling algorithms attempt to minimize average seek time
- Layout of files on disk has a big impact on file system performance

15

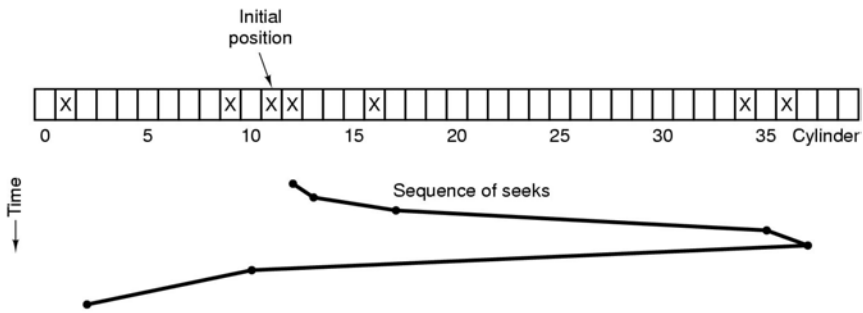
Disk Arm Scheduling Algorithms (2)



Shortest Seek First (SSF) disk scheduling algorithm

16

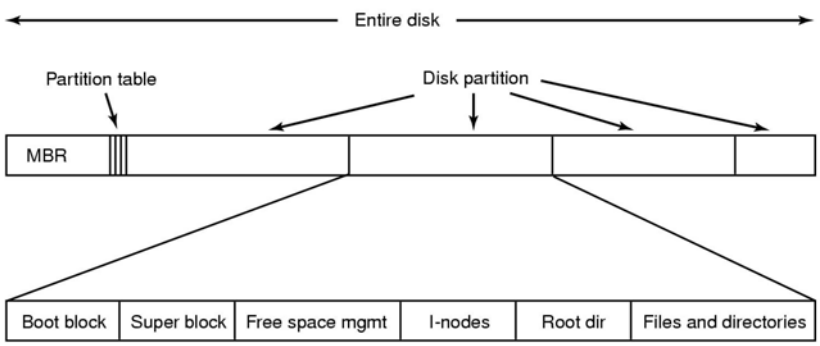
Disk Arm Scheduling Algorithms (3)



The elevator algorithm for scheduling disk requests

17

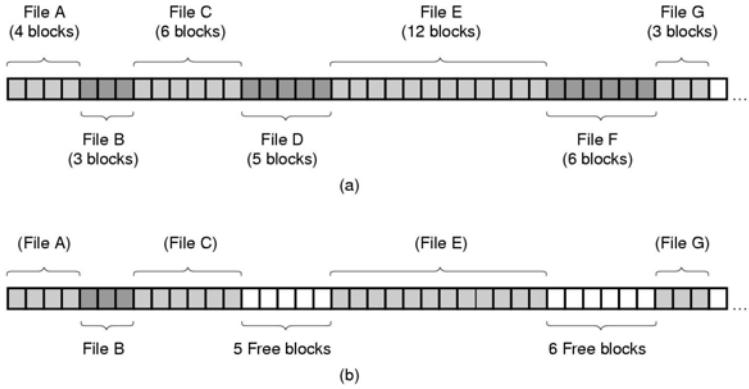
File System Implementation



A possible file system layout

18

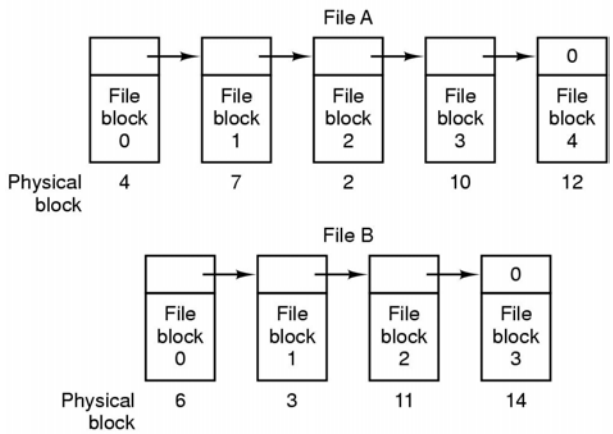
Implementing Files (1)



(a) Contiguous allocation of disk space for 7 files

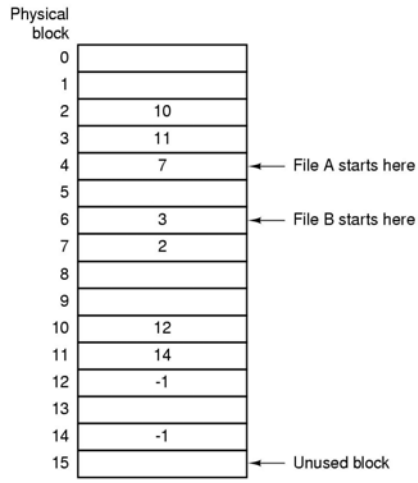
(b) State of the disk after files *D* and *E* have been removed

Implementing Files (2)



Storing a file as a linked list of disk blocks

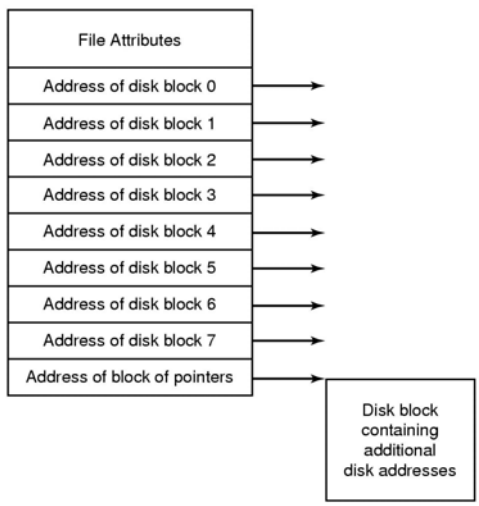
Implementing Files (3)



Linked list allocation using a file allocation table in RAM

21

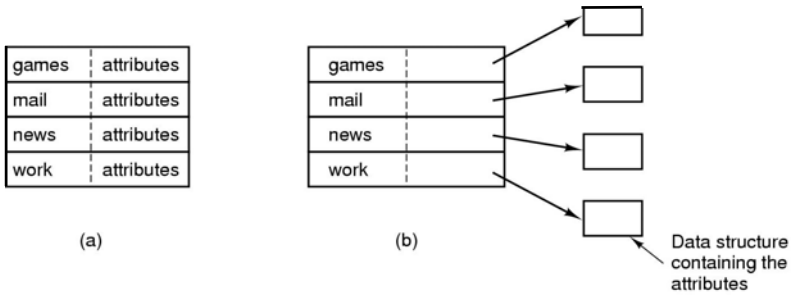
Implementing Files (4)



An example i-node

22

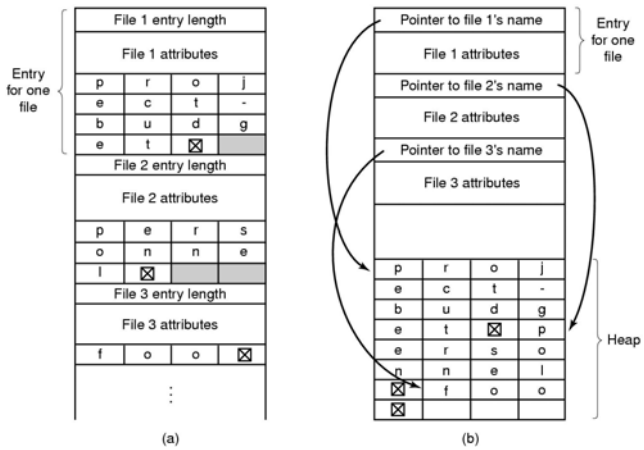
Implementing Directories (1)



(a) A simple directory
- fixed size entries
- disk addresses and attributes in directory entry

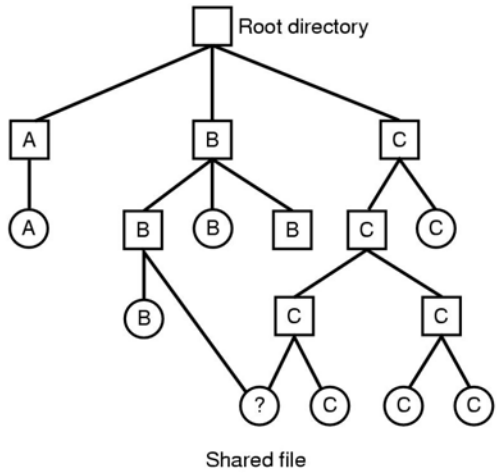
(b) Directory in which each entry just refers to an i-node

Implementing Directories (2)



- Two ways of handling long file names in directory
  (a) In-line (b) In a heap

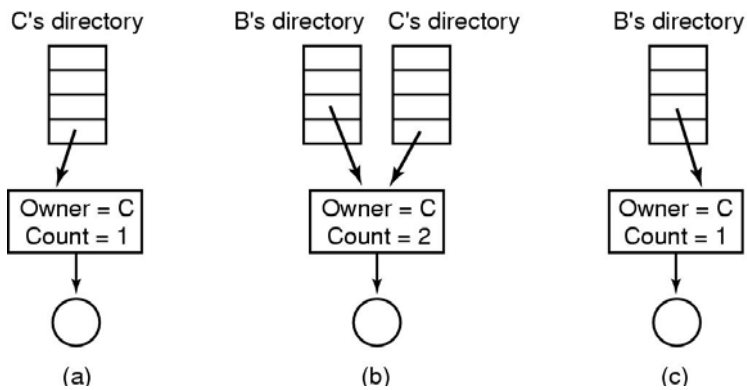
Shared Files (1)



File system containing a shared file

25

Shared Files (2)



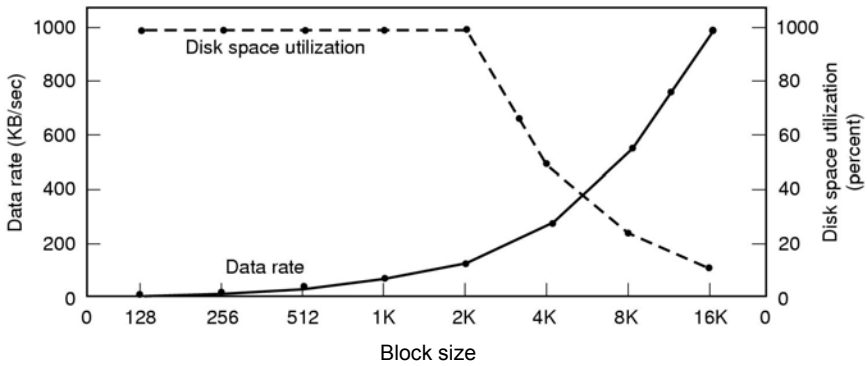
(a) Situation prior to linking

(b) After the link is created

(c)After the original owner removes the file

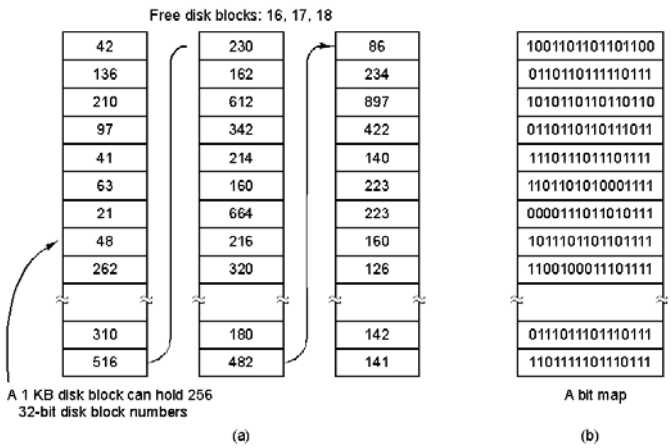
26

Disk Space Management (1)



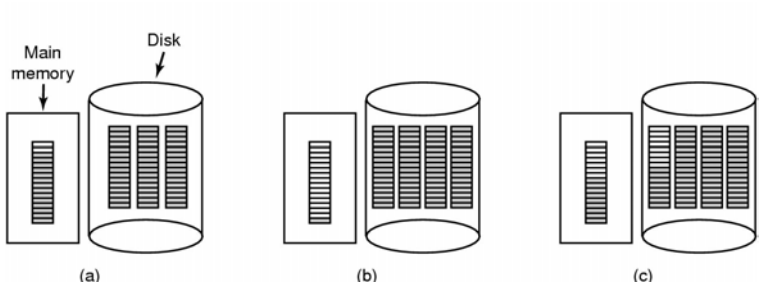
- Dark line (left hand scale) gives data rate of a disk
- Dotted line (right hand scale) gives disk space efficiency
- All files 2KB

Disk Space Management (2)



(a) Storing the free list on a linked list
(b) A bit map

Disk Space Management (3)



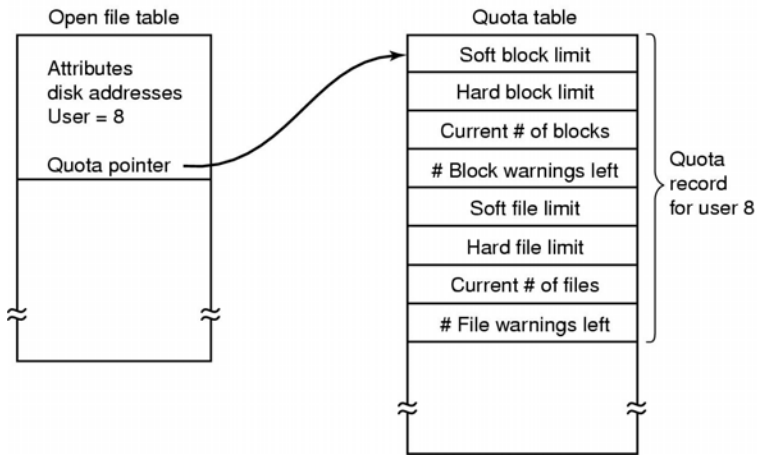
(a) Almost-full block of pointers to free disk blocks in RAM
- three blocks of pointers on disk

(b) Result of freeing a 3-block file

(c) Alternative strategy for handling 3 free blocks
- shaded entries are pointers to free disk blocks

29

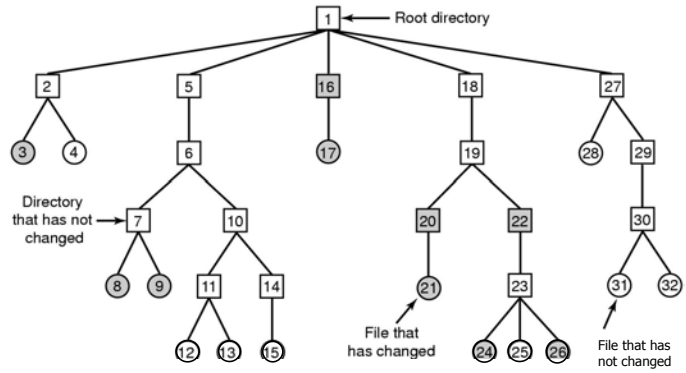
Disk Space Management (4)



Quotas for keeping track of each user's disk use

30

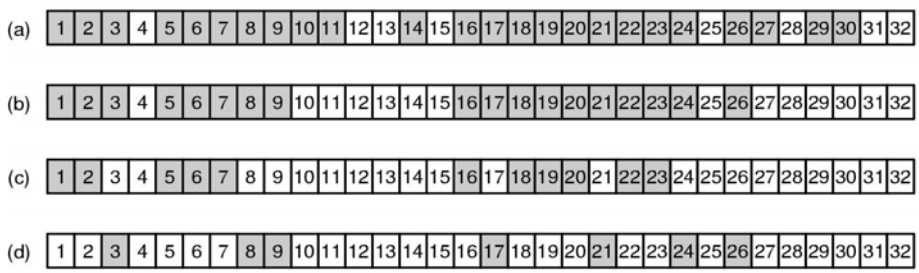
File System Reliability (1)



- A file system to be dumped
  - squares are directories, circles are files
  - shaded items, modified since last dump
  - each directory & file labeled by i-node number

31

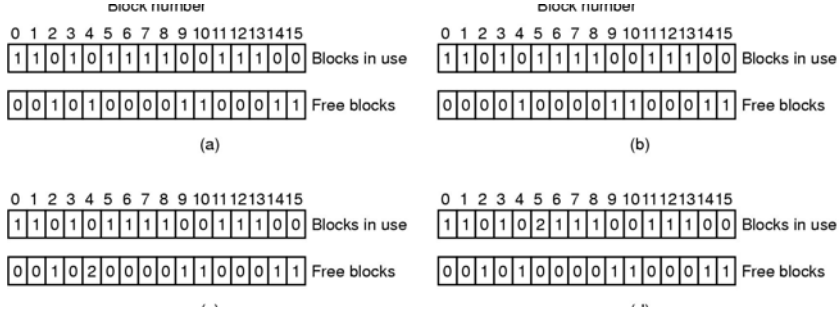
File System Reliability (2)



Bit maps used by the logical dumping algorithm

32

File System Reliability (3)

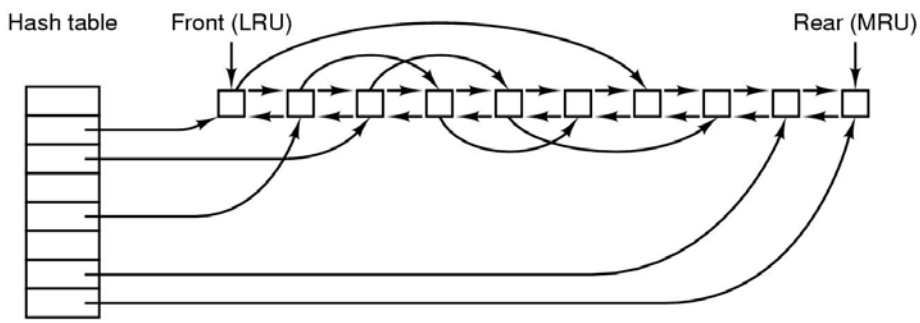


File system states

(a) consistent
(b) missing block
(c) duplicate block in free list
(d) duplicate data block

33

File System Performance (1)



The block cache data structures

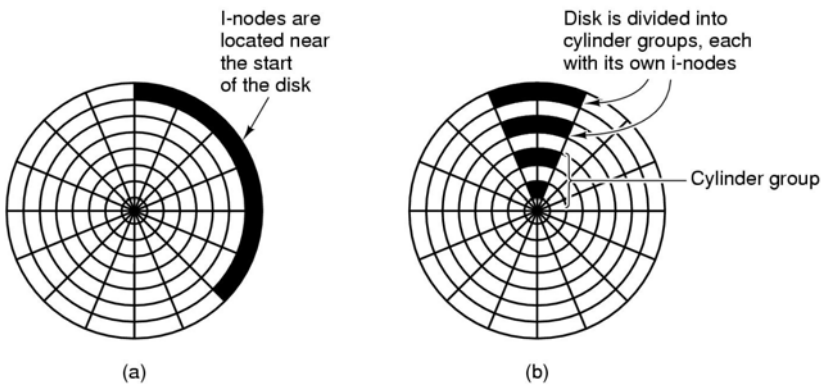
34

File System Performance(2)

- Block cache (also called buffer cache)
  - Modified LRU policy can be used for replacement where blocks corresponding to I-node blocks, indirect blocks, full data blocks, directory blocks, and partially full data blocks are treated differently
    - E.g., partially full blocks go to the end of the LRU list so that they will be cached for a while
  - Delayed write policy
    - "sync" operation performed every 30 seconds
    - Write thru policy for critical file system data (e.g. inodes)
- Block read ahead
  - Most files are read sequentially, so prefetching block k+1 into the cache when block k is accessed usually pays off

35

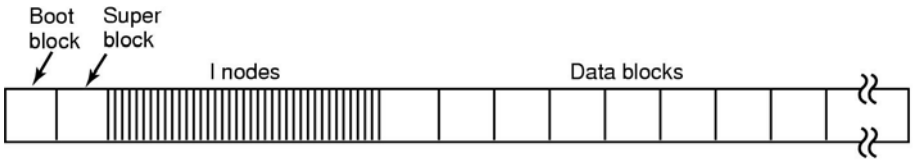
File System Performance (3)



- I-nodes placed at the start of the disk
- Disk divided into cylinder groups
  - each with its own blocks and i-nodes

36

UNIX File System (1)



Disk layout in classical UNIX systems

37

UNIX File System (2)

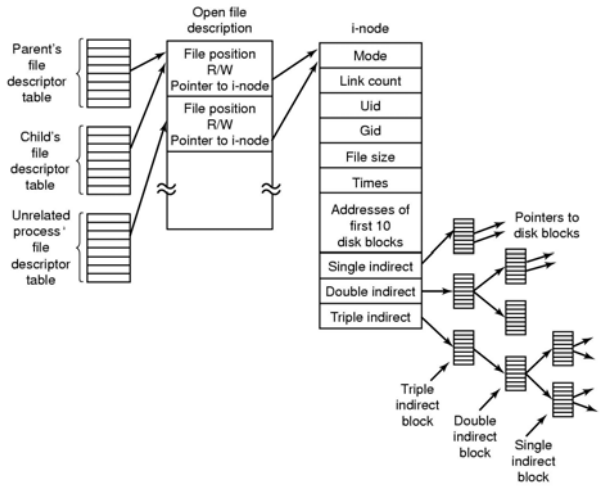
Directory entry fields.

Field	Bytes	Description
Mode	2	File type, protection bits, setuid, setgid bits
Nlinks	2	Number of directory entries pointing to this i-node
Uid	2	UID of the file owner
Gid	2	GID of the file owner
Size	4	File size in bytes
Addr	39	Address of first 10 disk blocks, then 3 indirect blocks
Gen	1	Generation number (incremented every time i-node is reused)
Atime	4	Time the file was last accessed
Mtime	4	Time the file was last modified
Ctime	4	Time the i-node was last changed (except the other times)

Structure of the i-node

38

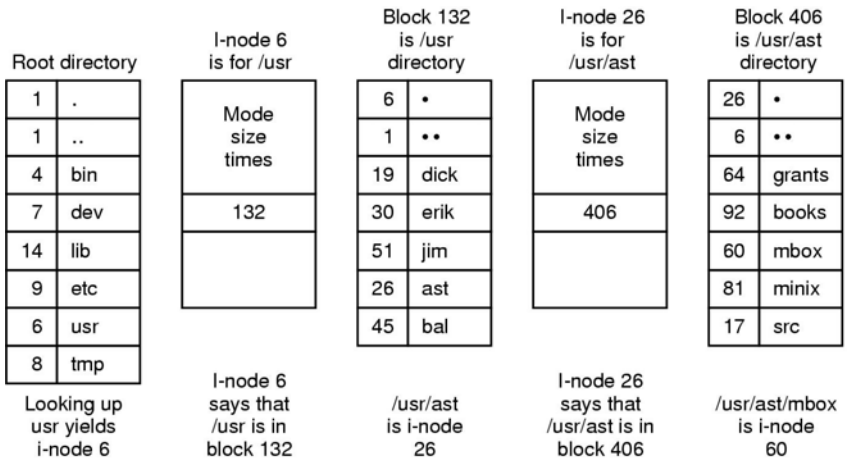
UNIX File System (3)



The relation between the file descriptor table, the open file description

39

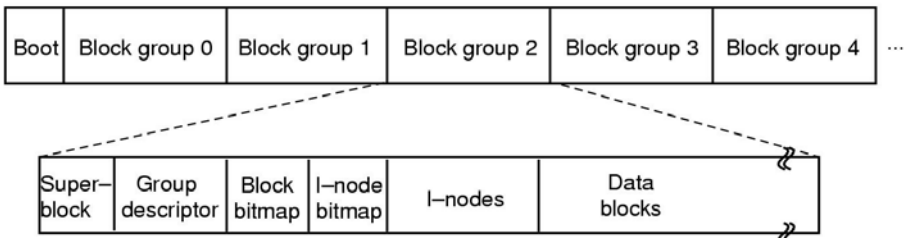
UNIX Filename Lookup



The steps in looking up */usr/ast/mbox*

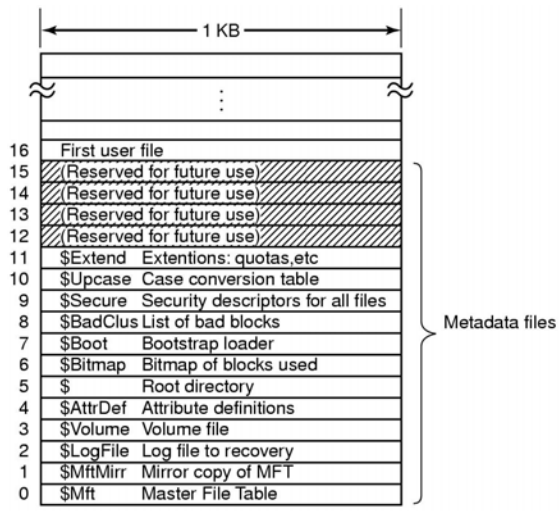
40

The Linux File System



Layout of the Linux Ex2 file system.

NTFS File System Structure (1)



The NTFS master file table

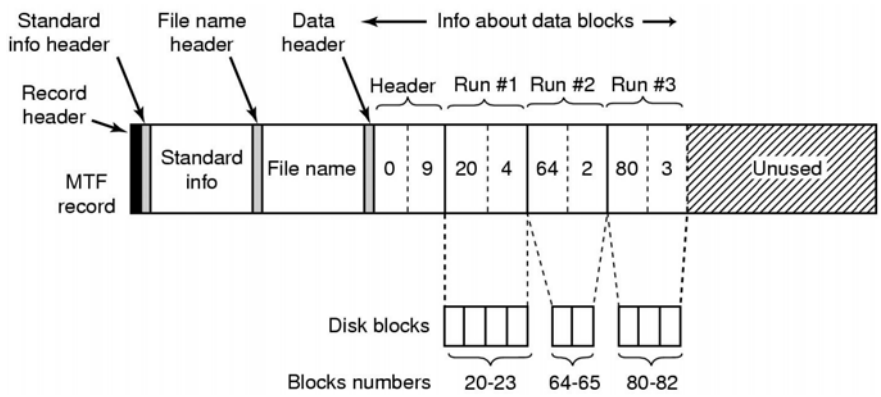
NTFS File System Structure (2)

Attribute	Description
Standard information	Flag bits, timestamps, etc.
File name	File name in Unicode; may be repeated for MS-DOS name
Security descriptor	Obsolete. Security information is now in $Extend$Secure
Attribute list	Location of additional MFT records, if needed
Object ID	64-bit file identifier unique to this volume
Reparse point	Used for mounting and symbolic links
Volume name	Name of this volume (used only in $Volume)
Volume information	Volume version (used only in $Volume)
Index root	Used for directories
Index allocation	Used for very large directories
Bitmap	Used for very large directories
Logged utility stream	Controls logging to $LogFile
Data	Stream data; may be repeated

The attributes used in MFT records

43

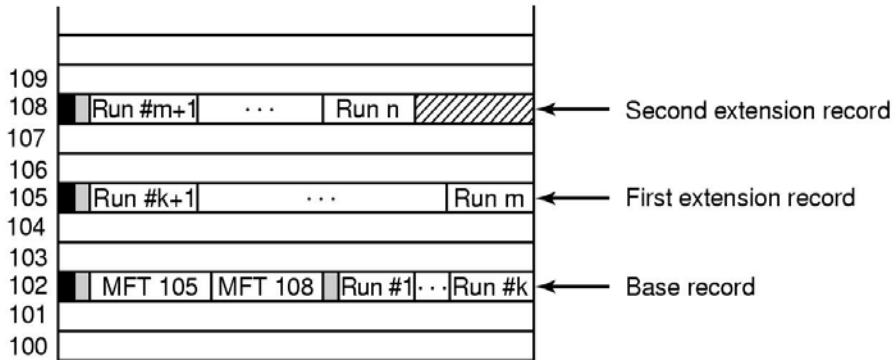
NTFS File System Structure (3)



An MFT record for a three-run, nine-block file

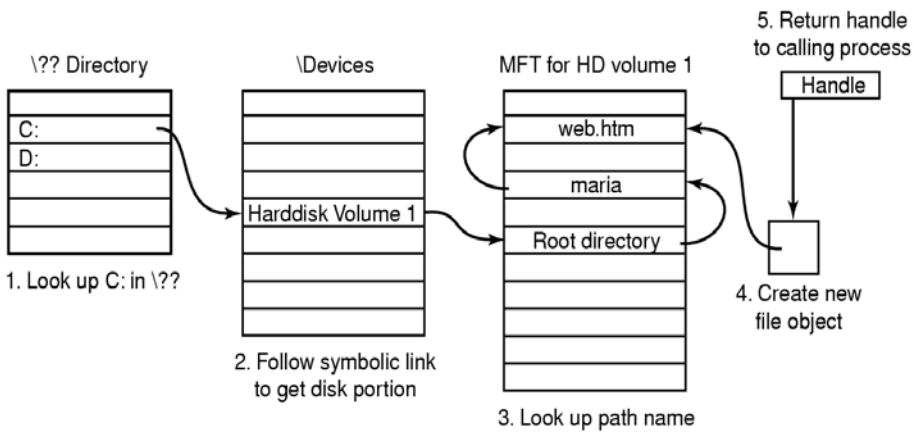
44

NTFS File System Structure (4)



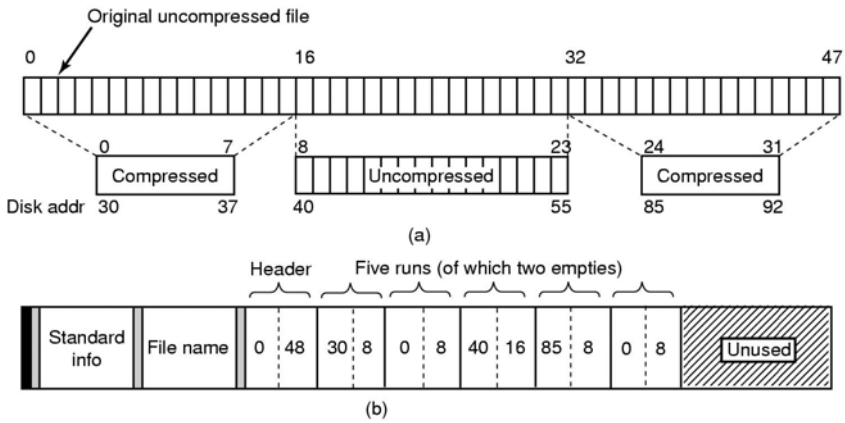
A file that requires three MFT records to store its runs

NTFS File Name Lookup



Steps in looking up the file *C:mariaweb.htm*

NTFS File Compression



(a) An example of a 48-block file being compressed to 32 blocks
(b) The MFT record for the file after compression

47