CS323 Operating Systems
File System V

Yuanyuan Zhou

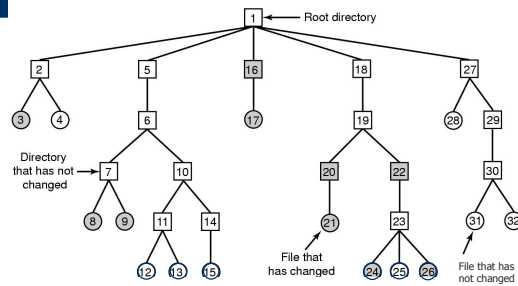Lecture 27

4/4/2003

Content of this lecture

- Administrative announcements
- File systems performance and reliability
- Summary

4/20/2003
CS 323 - Operating Systems,
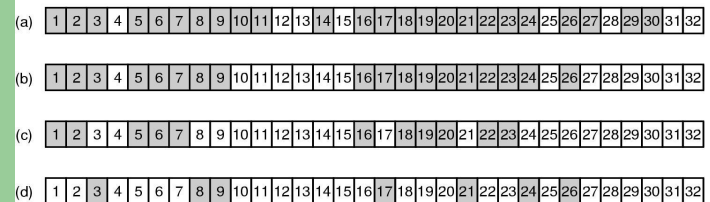Yuanyuan Zhou

---

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

4/20/2003
CS 323 - Operating Systems,
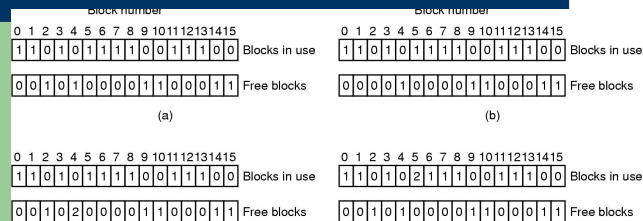Yuanyuan Zhou

---

File System Reliability (2)

- Bit maps used by the logical dumping algorithm

4/20/2003
CS 323 - Operating Systems,
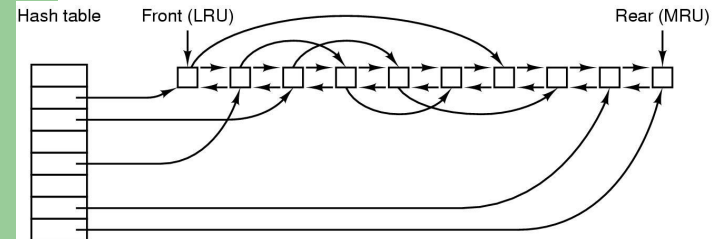Yuanyuan Zhou

File System Reliability (3)

- File system states
  - (a) consistent
  - (b) missing block
  - (c) duplicate block in free list
  - (d) duplicate data block

File System Performance (1)

- The block cache data structures

File System Performance (2)
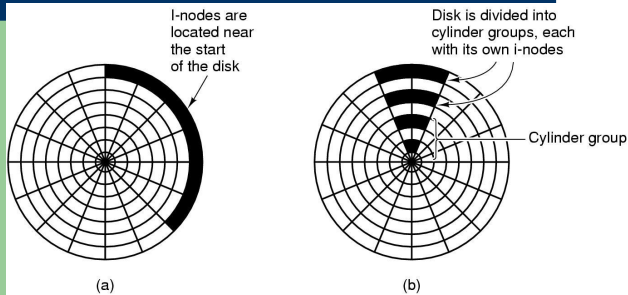
- I-nodes placed at the start of the disk
- Disk divided into cylinder groups
  - each with its own blocks and i-nodes

Log-Structured File Systems

- With CPUs faster, memory larger
  - disk caches can also be larger
  - increasing number of read requests can come from cache
  - thus, most disk accesses will be writes

- LFS Strategy structures entire disk as a log
  - have all writes initially buffered in memory
  - periodically write these to the end of the disk log
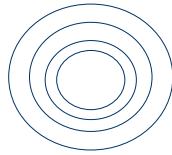 - when file opened, locate i-node, then find blocks

Comparison

Write block A;
write block B;
Write block A;
Write block C

4/20/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Discussion: Tradeoffs

- Traditional file system

- Log structured file system

4/20/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

Example File Systems
CD-ROM File Systems

- The ISO 9660 directory entry
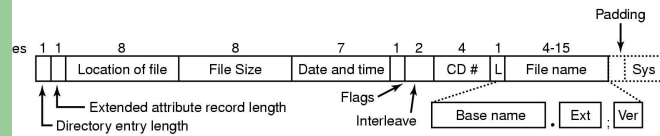
es	1	1	8	8	7	1	2	4	1	4-15	Padding
			Location of file	File Size	Date and time			CD #	L	File name	Sys

Extended attribute record length

Directory entry length

Flags

Interleave

Base name . Ext ; Ver

4/20/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

The CP/M File System (1)

Address

0xFFFF

BIOS

CP/M

Shell

User program

0x100

Zero page

0

4/20/2003

CS 323 - Operating Systems,
Yuanyuan Zhou

The CP/M File System (2)

Bytes

| 1 | 8 | 3 | 1 | 2 | 16 |

File name

User code — File type (extension) — Extent — Block count — Disk block numbers
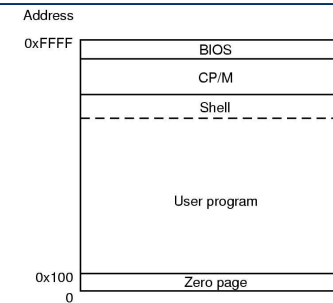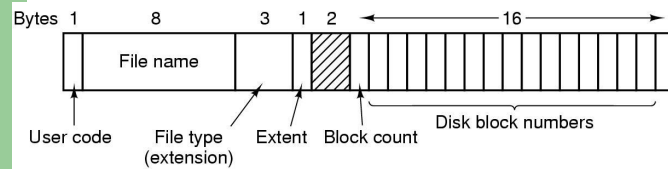
The CP/M directory entry format

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

The MS-DOS File System (FAT-16)

Bytes

| 8 | 3 | 1 | 10 | 2 | 2 | 2 | 4 |

File name — Size
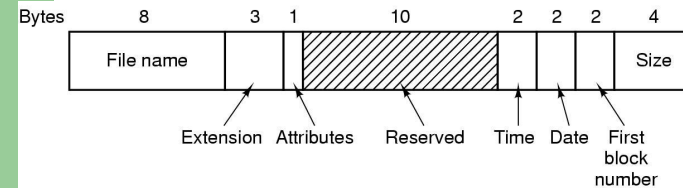
Extension — Attributes — Reserved — Time — Date — First block number

The MS-DOS directory entry

Disadvantage?

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

The MS-DOS File System (2)

Block size	FAT-12	FAT-16	FAT-32
0.5 KB	2 MB		
1 KB	4 MB		
2 KB	8 MB	128 MB	
4 KB	16 MB	256 MB	1 TB
8 KB		512 MB	2 TB
16 KB		1024 MB	2 TB
32 KB		2048 MB	2 TB

- Maximum partition for different block sizes
- The empty boxes represent forbidden combinations

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

The Windows 98 File System (1)

Bytes

| 8 | 3 | 1 | 1 | 1 | 4 | 2 | 2 | 4 | 2 | 4 |

Base name — Ext — N T — Creation date/time — Last access — Last write date/time — File size

Attributes — Sec — Upper 16 bits of starting block — Lower 16 bits of starting block
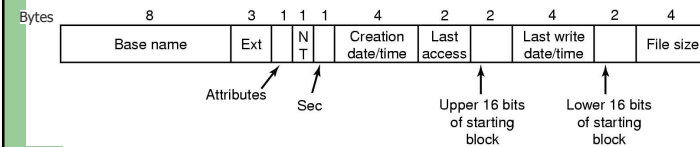
The extended MOS-DOS   directory entry used in Windows 98

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

4

The Windows 98 File System (2)

Bytes	1	10	1	1	1	12	2	4

5 characters | 0 | 6 characters | 0 | 2 characters

Sequence · Attributes · Checksum

An entry for (part of) a long file name in Windows 98

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou
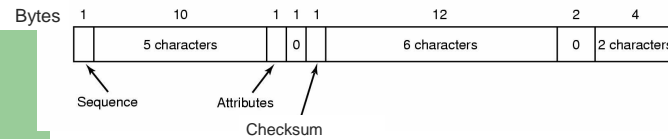

The Windows 98 File System (3)

68	d	o	g			A	0	C K								0		
3	o		v	e		A	0	C K	t		h	e		l	a	0	z	y
2	w	n		f	o	A	0	C K	x		j		u	m	p	0		s
1	T		h	e	q	A	0	C K	u		i	c	k		b	0	r	o
T	H	E	Q	U	I	~	1		A	N T	S	Creation time	Last acc	Upp	Last write	Low	Size

Bytes
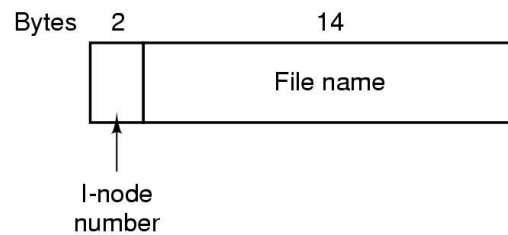
An example of how a long name is stored in Windows 98

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

The UNIX V7 File System (1)

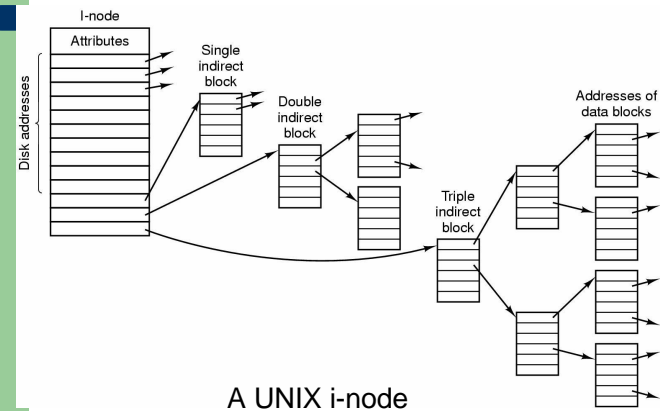| Bytes	2	14
		File name

I-node number

A UNIX V7 directory entry

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

The UNIX V7 File System (2)

A UNIX i-node

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

The UNIX V7 File System (3)

Root directory		I-node 6 is for /usr	Block 132 is /usr directory		I-node 26 is for /usr/ast	Block 406 is /usr/ast directory	
1	.		6	•		26	•
1	..	Mode size times	1	••	Mode size times	6	••
4	bin		19	dick		64	grants
7	dev	132	30	erik	406	92	books
14	lib		51	jim		60	mbox
9	etc		26	ast		81	minix
6	usr		45	bal		17	src
8	tmp						

Looking up usr yields i-node 6   I-node 6 says that /usr is in block 132   /usr/ast is i-node 26   I-node 26 says that /usr/ast is in block 406   /usr/ast/mbox is i-node 60

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

21

CS 323 - Operating Systems, Yuanyuan Zhou
4/20/2003

Levels of Access Methods

- device level read and write sectors or tracks from disk
- the i/o is written at the level of a sequence of transfer commands to the controller
- this is often performed as if the access path is a channel

22

CS 323 - Operating Systems, Yuanyuan Zhou
4/20/2003

Levels of Access Methods

- block level access to a file is in terms of blocks or physical records within a file
- the user must do his own buffering. Access methods include:
  - Read(file, block_no)
  - Write(file, block_no)
  - Wait(file, block_no)

23

CS 323 - Operating Systems, Yuanyuan Zhou
4/20/2003

Levels of Access Methods Continued

- file level access to the file is in terms of acquiring access to a copy of the file that is stored in primary memory
- queued or buffered level access to the file is in terms of logical records that depend on software interpretation. for example, read and write chars in UNIX. buffering is used to provide logical record abstraction and maps i/o into physical records

24

CS 323 - Operating Systems, Yuanyuan Zhou
4/20/2003

6

Levels of Access Methods Continued

- memory mapped file level
  - the file is mapped into virtual memory
  - file access is at the instruction level
  - page faults may read a page of file data from disk to memory
  - an address of a logical record within a file is given by a virtual memory address offset of that record from the beginning of the file

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Levels of Access Methods Continued

- persistent object
  - the file is mapped into virtual memory and
  - access to the contents of the file is provided by an abstract data type interface that is determined by
  - the type of the data stored in the file

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Protection

- in file systems, protection is needed from physical damage (reliability) and improper access (protection)
- reliability (chapter 12) is generally provided by duplicate copies of files

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Protection

- protection - various mechanisms for single-user system and multi-user systems
  - removing the floppy disk,
  - prohibiting access to files of other users

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Types - Controlled Access

- read - possible access to read from file
- write - possible access to write to a file
- execute - load file and execute it
- append - write new information at the end of a file
- delete - delete file and free its space for possible reuse
- list - list name and attributes of a file

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Access Lists and Groups

- associate each file and directory with access list
- problem with access list: length
- solution: condensed version of the access list
  - owner - user who created the file
  - group - a set of users who are sharing the file and need similar access
  - universe - all other users

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Access Lists Example

- UNIX - 3 fields of length 3 bits are used. fields are
- user(u),group(g),others(o), bits are read(r), write(w), execute(x) -
- example
- % Chmod   go+rw file

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Other Protection Approaches

- associate a password with each file
- protect directories - listing of file names might be a protected operation
- encryption

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Consistency Semantics

- UNIX
- Session Semantics
- Immutable-Shared File Semantics

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

UNIX Files

- writes to file immediately visible to other users reading from file
- can share current location into file through open file tables

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Session Semantics

- writes to open file by user not visible immediately – only to new opens.
- closing a write file doesn't make contents visible if applications have file open already

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Immutable-shared Files

- files, once shared, never change
- name may not be reused
- good in distributed system

4/20/2003
CS 323 - Operating Systems, Yuanyuan Zhou

Summary

- file concepts - file attributes, operations, structures
- directory systems - single-level, two-level, tree structure, acyclic structure, general structure
- access methods - sequential, direct, indexed
- protection - possible access protection, access lists

4/20/2003

CS 323 - Operating Systems, Yuanyuan Zhou