4.{25,26,28,29,32,33,35,36}

**25.** The beginning of a free-space bitmap looks like this after the disk partition is first formatted:

1000 0000 0000 0000 (the first block is used by the root directory). The system

always searches for free blocks starting at the lowest-numbered block, so after

writing file *A*, which uses six blocks, the bitmap looks like this: 1111 1110 0000 0000.

Show the bitmap after each of the following additional actions:

(a) File *B* is written, using five blocks. 1111 1111 1111 0000

(b) File *A* is deleted. 1000 0001 1111 0000

(c) File *C* is written, using eight blocks. 1111 1111 1111 1100

(d) File *B* is deleted. 1111 1110 0000 1100

**26.** What would happen if the bitmap or free list containing the information about free disk

blocks was completely lost due to a crash? Is there any way to recover from this disaster,

or is it bye-bye disk? Discuss your answers for UNIX and the FAT -16 file system

separately.

Modern operating systems can recover, this is not a serious problem. Make a list of all blocks in all files. If all blocks are marked free, then find blocks associated with free list by traversing recursively from the root. FAT-16 does not allow links making easy recursive traversal, unix keeps track of what has been traversed. Unix uses free list instead of FAT.

**28.** We discussed making incremental dumps in some detail in the text. In Windows it is

easy to tell when to dump a file because every file has an archive bit. This bit is missing

in UNIX. How do UNIX backup programs know which files to dump?

The time of the last dump is stored in a file on a disk. Every dump is appended to the file, each dump has a dump time and last dump. The files with changes since the dump time are dumped.

**29.** Suppose that file 21 in Fig. 4-25 was not modified since the last dump. In what way

would the four bitmaps of Fig. 4-26 be different?

21 not marked for a,b and d. c is the same.

**32.** The performance of a file system depends upon the cache hit rate (fraction of blocks

found in the cache). If it takes 1 msec to satisfy a request from the cache, but 40 msec

to satisfy a request if a disk read is needed, give a formula for the mean time required

to satisfy a request if the hit rate is *h*. Plot this function for values of *h* varying from 0

to 1.0.

h + 40 \* (1 - h), straight line

**33.** For an external USB hard drive attached to a computer, which is more suitable: a writethrough

cache or a block cache?

Writethough is best. In writethrough, the block in cache and the block in lower memory has the data. Easy to implement.

**35.** Consider a disk that has 10 data blocks starting from block 14 through 23. Let there be

2 files on the disk: f1 and f2. The directory structure lists that the first data blocks of f1

and f2 are respectively 22 and 16. Given the FAT table entries as below, what are the

data blocks allotted to f1 and f2?

(14,18); (15,17); (16,23); (17,21); (18,20); (19,15); (20, −1); (21, −1); (22,19); (23,14).

In the above notation, (*x*, *y*) indicates that the value stored in table entry *x* points to data

block *y*.

f1: 22, 19, 15, 17, 21. f2: 16, 23, 14, 18, 20

**36.** Consider the idea behind Fig. 4-21, but now for a disk with a mean seek time of 6

msec, a rotational rate of 15,000 rpm, and 1,048,576 bytes per track. What are the data

rates for block sizes of 1 KB, 2 KB, and 4 KB, respectively?

6 + 2 + (k/1048576)\*4 = 1KB = 8.000003815, 2KB = 8.000007629, 4KB = 8.000015259