File Systems

Practice Problems Set #3

1. Assume a file system that uses a "contiguous" disk allocation strategy. Assume that there is a file on the disk that is 5 disk blocks long. Describe the steps, including any disk read or write operations, required by the operating system to delete the 3rd block of the file.

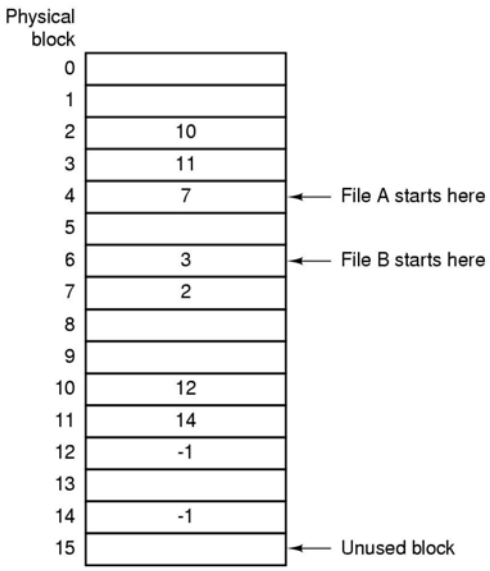
2. Assume a file system that uses a contiguous disk allocation strategy. Assume that there is a file that is 10 disk blocks long (starting block is 1, ending block is 10) and that the file's descriptor information is already in main memory.
   a. If we want to access information on block 6 of the file, how many blocks must be read from the disk? Explain why.
   b. The same questions but assume that the file system uses a linked disk allocation strategy.
   c. The same question but assume that the file system uses an index node (i-node) allocation strategy.

3. Contiguous allocation of files leads to disk fragmentation, as mentioned in class. Is this internal fragmentation or external fragmentation? Make an analogy with another component of an operating system.

4. Free disk space can be kept track of using a free list or a bitmap. Disk addresses require D bits. For a disk with B blocks, F of which are free, state the conditions under which the free list uses less space than the bitmap. For D having the value 16 bits, express your answer as a percentage of the disk space that must be free.

5. It has been suggested that the first part of a UNIX file be kept in the same disk block as its i-node. What good would this do?

6. The figure below shows the structure of the original FAT file system used on MS-DOS. Originally this file system had only 4096 blocks, so a table with 4096 (12-bit) entries was enough. If that scheme were to be directly extended to file systems with 2^{32} blocks, how much space would the FAT occupy?



7. A disk has 4000 cylinders, each with 8 tracks of 512 blocks. A seek takes 1 msec per cylinder moved. If no attempt is made to put the blocks of a file close to each other, two blocks that are logically consecutive (i.e., follow one another in the file) will require an average seek, which takes 5 msec. If, however, the operating system makes an attempt to cluster related blocks, the mean interblock distance can be reduced to 2

cylinders and the seek time reduced to 100 microsec. How long does it take to read a 100 block file in both cases, if the rotational latency is 10 msec and the transfer time is 20 microsec per block?

8. Consider a hierarchical file system in which free disk space is kept in a free space list.
    a. Suppose the pointer to free space is lost. Can the system reconstruct the free space list?
    b. Suggest a scheme to ensure that the pointer is never lost as a result of a single memory failure.

9. Consider the organization of a UNIX file as represented by the inode. Assume there are 12 direct block pointers and a singly, doubly and triply indirect pointer in each inode. Further, assume that the system block size and the disk sector size are both 8K. If the disk block pointer is 32 bits, with 8 bits to identify the physical disk and 24 bits to identify the physical block, then:
    a. What is the maximum file size supported by this system?
    b. What is the maximum file system partition supported by this system?
    c. Assuming no information other than the file inode is already in main memory, how many disk accesses are required to access the byte in position 100K+10, where K=1024?

10. Two computer scientists, Carolyn and Elinor, are having a discussion about i-nodes. Carolyn maintains that memories have gotten so large and so cheap that when a file is opened, it is simpler and faster just to fetch a new copy of the i-node into the i-node table, rather than search the entire table to see if it is already there. Elinor disagrees. Who is right and why? (Support your answer clearly.)

11. Consider a system where free space is kept in a free space list.
    a. Supposed that the pointer to the free space list is lost. Can the system reconstruct the free space list? Explain your answer.
    b. Consider a file system similar to the one used by UNIX with indexed allocation. How many disk I/O operations might be required to read the contents of a small local file at /a/b/c? Assume that none of the disk blocks is currently being cached.
    c. Suggest a scheme to ensure that the pointer is never lost as a result of memory failure.

12. Some file systems allow disk storage to be allocated at different levels of granularity. For instance, a file system could allocate 4 KB of disk space as a single 4 KB block or as eight 512-byte blocks. How could we take advantage of this flexibility to improve performance? What modifications would have to be made to the free space management scheme in order to support this feature?

13. Consider a file system on a disk that has both logical and physical block sizes of 512 bytes. Assume that the information about each file is already in memory.

For each of the three allocation strategies (contiguous, linked, and indexed), answer these questions:

a. How is the logical to physical address mapping accomplished in this system? (For the indexed allocation, assume that the file is always less than 512 blocks long.)
b. If we are currently at logical block 10 (the last block accessed was block 10) and want to access logical block 4, how many physical blocks must be read from the disk?

14. What are the advantages of the variant of linked allocation that uses a FAT to chain together the blocks of a file?

15. Fragmentation on a storage device would be eliminated by the recompaction of the information. Typical disk devices do not have relocation or base registers (such as are used for relocation in memory), so how can we relocate files? Give three reasons why compacting and relocation of files are often avoided.