Homework Assignment #4 Solution

Q1 (Problem 10) Consider the directory tree of Fig. 4-8. If /usr/jim is the working directory, what is the absolute path name for the file whose relative path name is ../ast/x?

ANSWER: The dotdot component moves the search to */usr*, so *../ast* puts it in */usr/ast*. Thus *../ast/x* (a relative path name) is the same as */usr/ast/x* (an absolute path name).

Q2 Contiguous allocation can lead to disk external fragmentation, as mentioned in the text. Please explain why.

ANSWER: Contiguous allocation can lead to wasted storage *between* allocation units (files), not inside them. This is external fragmentation. Please see Figure 4-10 (b) as an example – even we have 11 free blocks available, but because they are not contiguous, we cannot allocate to a file requesting 10 blocks. This wasted storage is caused by external fragmentation. It is precisely analogous to the external fragmentation of main memory that occurs with a swapping system or a system using pure segmentation.

Q3 (Modified Problem 16). Consider the i-node shown in Fig. 4-13. If it contains 10 direct addresses of 4 bytes each and one single indirect block. All disk blocks are 1024B (1KB). What is the largest possible file?

ANSWER: The indirect block can hold 1024/4=256 disk addresses. Together with the 10 direct disk addresses, the maximum file has 256+10=266 blocks. Since each block is 1024 B, the largest file is 266KB.

Q4 (Problem 21). Name one advantage of hard links over symbolic links and one advantage of symbolic links over hard links.

ANSWER: Hard links do not require any extra disk space, just a counter in the i-node to keep track of how many there are. In addition, it can quickly locate shared files, and the performance is better than symbolic links. Symbolic links need space to store the name of the file pointed to.

Symbolic links can point to files on other file systems or on even other machines or over the Internet. Hard links are restricted to pointing to files within their own partition. In addition, symbolic links have easier metadata management compared to hard links.

Q5 (Problem 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
(b) File A is deleted
(c) File C is written, using eight blocks
(d) File B is deleted.

ANSWER: The beginning of the bitmap looks like:
(a) After writing file *B*: 1111 1111 1111 0000
(b) After deleting file *A:* 1000 0001 1111 0000
(c) After writing file *C*: 1111 1111 1111 1100
(d) After deleting file *B*: 1111 1110 0000 1100

Q6 (Problem 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?

ANSWER: In (a) and (b), 21 would not be marked. In (c), there would be no change. In (d), 21 would not be marked.

Q7 (Problem 30). It has been suggested that the first part of each UNIX file be kept in the same disk block as its i-node. What good would this do?

ANSWER: Many UNIX files are short. If the entire file fits in the same block as the i-node, only one disk access would be needed to read the file, instead of two, as is presently the case. Even for longer files there would be a gain, since one fewer disk accesses would be needed. This is more commonly known as “i-node stuffing” or “stuffed i-node”.

Q8 (Problem 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.

ANSWER: The time (in msec) needed is $h+40 \times (1-h)$. The plot is just a straight line (omitted here).

Q9 (Problem 37). A certain file system uses 4-KB disk blocks. The median file size is 1 KB. If all files were exactly 1 KB, what fraction of the disk space would be wasted? Do you think the wastage for a real file system will be higher than this number or lower than it? Explain your answer.

ANSWER: If all files were 1 KB, then each 4-KB block would contain one file and 3 KB of wasted space. Trying to put two files in a block is not allowed because the unit used to keep track of data is the block, not the semiblock. This leads to 75% wasted space. In practice, every file system has large files as well as many small ones, and these files use the disk much more efficiently. For example, a 32,769-byte file would use 9 disk blocks for storage, given a space efficiency of 32768/34816, which is about 94%.

Q10 (Problem 41). How many disk operations are needed to fetch the i-node for a file with the path name /usr/ast/courses/os/handout.t? Assume that the i-node for the root directory is in memory, but nothing else along the path is in memory. Also assume that all directories fit in one disk block.

ANSWER: The following disk reads are needed:
directory for /
i-node for */usr*
directory for */usr*
i-node for */usr/ast*
directory for */usr/ast*
i-node for */usr/ast/courses*
directory for */usr/ast/courses*
i-node for */usr/ast/courses/os*
directory for */usr/ast/courses/os*
i-node for */usr/ast/courses/os/handout.t*
In total, 10 disk reads are required.

THE END.