

CSE 430 Homework 8

Ryan Dougherty

Question 11.9

Consider a file system in which a file can be deleted and its disk space reclaimed while links to that file still exist. What problems may occur if a new file is created in the same storage area or with the same absolute path name? How can these problems be avoided? Let $File_1$ and $File_2$ be the old and new files, respectively. Anyone trying to access $File_1$ will be accessing $File_2$, and the access protection for $File_1$ is used instead of $File_2$'s.

We can avoid the problem by insuring that any and all connections to deleted files are also deleted (when a file is deleted), by doing any one of the following:

- (a) Keep a list of all current connections to the file, and remove all of them when the file is deleted.
- (b) Keep the connections to the file and remove them whenever the deleted file tries to be accessed.
- (c) Keep a list of all current connections to the file and only delete the file after all existing connections to it have been deleted.

Question 11.14

If the operating system knew that a certain application was going to access file data in a sequential manner, how could it exploit this information to improve performance? The OS can prefetch the subsequent blocks into memory so that the application does not have to retrieve the data from hard disk, since memory is faster than hard disk (reduce waiting time).

Question 12.11

What are the advantages of the variant of linked allocation that uses a FAT to chain together the blocks of a file? An advantage of this scheme is that when accessing a block located in the middle of a file, we can figure out the location by moving through the list of pointers stored in the FAT. It is possible to cache most of the FAT → we can determine these pointers only with memory accesses instead of needing to access disk blocks.

Question 12.17

Fragmentation on a storage device can be eliminated by recompaction of the information. Typical disk devices do not have relocation or base registers (such as those used when memory is to be compacted), so how can we relocate files? Give three reasons why recompact and relocation of files are often avoided. Three reasons why recompact and relocation of files are often avoided are:

1. Overhead exists in relocation of files because data blocks have to be read into main memory (and written back out again to new file locations).
2. It is only possible to use relocation registers for sequential files. However, many files are not sequential (i.e. non-contiguous on disk).
3. Sequential files can be placed in non-contiguous blocks if connections between logically sequential ones are maintained.