CSS-430 : Operating Systems : HW09-Questions

David Liu

CSS 430

**Assignment Text**

Complete the following problems from the OSC book, 10th edition:

File System Interface:

* Problem 13.10
* Problem 13.14
* Problem 13.16

File System Implementation

* Problem 14.8
* Problem 14.14

13.10

Question: The open-file table is used to maintain information about files that are currently open. Should the operating system maintain a separate table for each user or maintain just one table that contains references to files that are currently being accessed by all users? If the same file is being accessed by two different programs or users, should there be separate entries in the open-file table? Explain.

The operating system should have a central open-file table that keeps track of which users/processes are accessing that file. The file cannot be deleted until all processes have closed the file. Separate states of the file also need to be maintained if multiple processes access it. The central open-file table is a system capable of performing all these operations.

13.14

Question: 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?

You can fetch the files beforehand which would subsequently reduce the wait time and increase performance between processes.

13.16

Question: Some systems provide file sharing by maintaining a single copy of a file. Other systems maintain several copies, one for each of the users sharing the file. Discuss the relative merits of each approach.

Multiple copies are costly to space and have a problem with potential inconsistencies if they are all modified. Singly copy files would allow for changes to all be seen on one copy but you could potentially corrupt that single copy.

14.8

Question: Contrast the performance of the three techniques for allocating disk blocks (contiguous, linked, and indexed) for both sequential and ran-dom file access.

Contiguous – Sequential – Access of contiguous disk blocks are made easy as the file blocks are linearly ordered

Contiguous – Random – You can easily determine the next disk block you want using FBN

Linked – Sequential - allows for a pointer to the next file even though the disk blocks may be scattered everywhere.

Linked – Random – You have to traverse several undesired blocks until you reach the one you want

Indexed – Sequential - Allows for direct access to be accessed by increasing the index in one central location: the index block.

Indexed – Random – You can determine the index associated with the target disk block if you have enough information. (If it’s ordered then you can find it faster than Linked) (Hashed can be even faster)

14.14

Question: 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:

1. How is the logical-to-physical address mapping accomplished in this system? (For the indexed allocation, assume that a file is always less than 512 blocks long.)
   1. Contiguous – You divide the logical address into blocks (divide by 512) to find the answer and a remainder. The answer represents the file block number and the remainder represents the offset within that block.
   2. Linked – You divide the logical address into blocks (divide by 512) to find the answer and a remainder. You traverse the linked list by Address / 512 + 1 times and find the offset Address % 512 + 1.
   3. Indexed - You divide the logical address into blocks (divide by 512) to find the answer (result) and a remainder. The block is at table[result] and the offset is the remainder.
2. 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?
   1. Contiguous – 1
   2. Linked – 4
   3. Indexed - 2