

13.10

A separate table is not necessary for each user. The open-file table is at the kernel and thus for all users

---

13.14

Add the blocks preceding and/or succeeding the data to a buffer for quicker access

---

14.8

Contrast the performance of the three techniques for allocating disk blocks

## Sequential

Contiguous: Accessing sequential files consist of finding the starting index of the file and traversing through the contiguous blocks. Will result in fragmentation however and initial allocation of blocks will be hard if the file size is large

Linked: Because each block contains a pointer to the next block, reading sequential blocks of memory is easy. Removes fragmentation issue. Slightly slower and more wasteful than contiguous because pointers have to be looked up and stored in the block.

Indexed: Would have to reference the index block N times where N is the block size of the file. Much slower. Limited in file size by index block pointer capacity

## Random

Contiguous: Could multiply the logical block by the block size + starting physical block to get desired block. Limits to file size are not as impactful due to memory being smaller anyways

Linked: Would have to traverse through the entire block list to find the desired block. Very slow

Indexed: Ideal. When wanting a specific block of a file, just need to look up that file in the application table and the blocks allocated to it.

---

14.14) 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.

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.)
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?

Contiguous:

1. If you want logical block 30, multiply  $30 * 512$  and add it to the starting physical block number
2. One block. Just perform the above calculation to directly access it

Linked:

1. If you want logical block 30, will have to traverse (logical physical address by 511) blocks. Once you've found the host block, still have to travel the remainder within the block
2. Four traversals from the beginning

Indexed

1. Divide the desired logical address by 512 to get the logical block number. Find the physical block address at index N in the index table. The remainder is the displacement within that block
  2. One look up for the index table. One lookup for the block
- 

14.15

$2^{13}$  bytes per block

$2^2$  bytes per pointer

$2^{11}$  pointers per block

$(12 + 2^{11} + 2^{22} + 2^{33}) * 2^{13}$  bytes = 7.0403121<sup>13</sup> bytes