

1 Quercus Component

1.
 - c) code
 - d) heap memory
2.
 - a) number of links to this inode
 - c) type of file system object
 - d) file size
 - f) array of pointers to blocks
3.
 - a) blcok bitmap
 - b) inode
 - c) data block
4. b) data block bitmap

2 Markus Component

1. Algorithm A would complete faster.

- We are only moving the used blocks to free blocks
- It doesn't have to be concerned with finding related but scattered data blocks
 - Since it's not finding it, the total rotational delay time and seek time is low

On the other hand Algorithm B has to find all data blocks scattered across the disk, and to make relevant data blocks contiguous, it has to move other blocks around. Finding each block adds seek and rotational delay time, and moving further adds the day. In total, the delay time is much larger than algorithm A.

2. Algorithm B would lead to better performance.

The reason is that

- For, algorithm A, the data blocks to a file are placed more randomly
 - the hard disk drive still has to move its arms to find all related data blocks
 - * This adds rotational delay and seek time
- For algorithm B, the data blocks to a file are placed contiguously
 - Then, we can write that the data blocks to a file are placed on current or nearby tracks.
 - Since data blocks are placed on current or nearby tracks, it requires little or no movement for disk arm
 - It follows from above that rotational delay and seek time are kept to minimum

3. What changes, if any, are made to the inodes? Explain your answer.

- time modified, time accessed
 - time modified because data blocks are repositioned
 - time accessed because file has to be read in order to find its data blocks for defragmentation
- the value of pointers to indirect blocks and data blocks
 - Is because under the defragmentation algorithm, data blocks and indirect blocks are moved so they are located in contiguously. This changes block address. This change needs to be reflected in inode.

What changes, if any, would need to be made to the directory metadata by the defragmentation algorithm? Explain your answer.

- i-number in each directory entry
 - Is because inodes are being moved around from a group block to another so related directory inodes are put in group as close as possible
- Pointer to data blocks
 - Is because defrag algorithm causes data blocks to be placed contiguously
 - Since directory inode stores addresses to these blocks, changes has to be made to reflect the changes
- Time accessed
 - Is because directory to be accessed to find all its data blocks before moving around.
- Time modified
 - Is because i-number in directory entries, and inode's address to directory blocks are changed

Correct Solution

What changes, if any, are made to the FAT (File Allocation Table) by the defragmentation algorithm? Explain your answer.

Direct pointers and indirect pointers must be updated. This is because data blocks are moved around under defragmentation.

What changes, if any, are made to the directory metadata by the defragmentation algorithm? Explain your answer.

Other than pointers to data blocks and indirect blocks, nothings are changed. This is because i-number in directory entries remains the same, and the contents of inode is not changed,

Notes

- I should clarify question with professor regarding time accessed. Shouldn't inode be accessed to read data blocks so on change we can update direct pointers and indirect block pointers?
4. **What changes, if any, are made to the FAT (File Allocation Table) by the defragmentation algorithm? Explain your answer.**
- pointers to next data block
 - FAT is a file system based on linked-list based allocation.
 - Linked list allocation uses next member variable to locate the next data block
 - Defrag moves the data blocks around, and this affects pointers to next data block and pointer to first data block in directory entry

What changes, if any, are made to the directory metadata by the defragmentation algorithm? Explain your answer.

- time accessed, time modified, i-number in each directory entry, pointer to first data block containing directory entries
 - Since defrag moves data blocks around, we can write data blocks are moved. Since data blocks include directory we can conclude i-number in each directory entry and directory's pointer to next data block is changed.
 - Since directory must be read to get its data blocks so they can be placed contiguously, we can write directory is accessed. Since directory is accessed, we can conclude time accessed must be changed.
 - Since i-number and first pointer are changed, we can write directory is modified. Since directory is modified, we can conclude must be reflected in time modified

Correct Solution

What changes, if any, are made to the FAT (File Allocation Table) by the defragmentation algorithm? Explain your answer.

FAT table needs to be changed to create links to new blocks

What changes, if any, are made to the directory metadata by the defragmentation algorithm? Explain your answer.

Pointer to first data block must be changed. This is because under defragmentation, data blocks are moved around, and the data block it points to is now at a different address