10+7/2 = ~~8.5/10~~ 9/10

COSC 450 Operating System Mini-Test #3-Again

Take-home submit by 11/16/22

Please try to do yourself! Please do not share solution!!!!!

Name: Jung An.

1. (2 pt.) One way to use contiguous allocation of the disk and not suffer from holes is to compact the disk every time a file is removed. Since all files are contiguous, copying a file requires a seek and rotational delay to read the file, followed by the transfer at full speed. Writing the file back requires the same work. Assuming a seek time of 5 msec, a rotation delay of 4 msec, a transfer rate of 8MB/sec, and an average file size of 8 KB, how long does it take to read a file into main memory and then write it back to disk at a new location? Using these numbers, how long would it take to compact half of a 16 GB disk.

seek = 5msec

rotation = 4msec

transfer rate = 8MB/sec

Average size = 8KB

formula = 1seek + (Average file/transfer) + rotation delay

5mse + ($2^3 \cdot 2^{10}$ byte / $2^3 \cdot 2^{20}$ byte/sec · $\frac{1sec}{1000ms}$) + 4msec

read time = 5msec + ($\frac{1000msec}{2^{10}}$) + 4msec = 9.977msec

read + write = 9.977×2 = 19.953msec

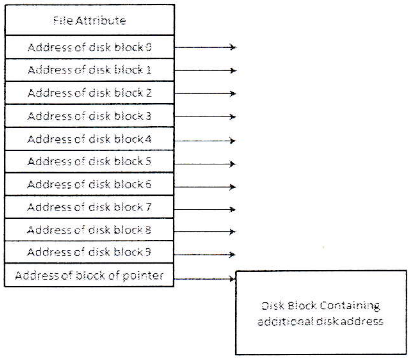
8GB/8kB ratio = $2^3 \cdot 2^{30} / 2^3 \cdot 2^{10} = 2^{20}$

time = 2^{20} · 19.953 = 20922368msec = 20922.368sec = ✓

348.7061333min = 5.811768889 hours

2. (2 pt.) LINUX like system use i-node to maintain the file system. Attributes and block addresses are saved in i-node. One problem with i-nodes is that if each one has room for a fixed number of disk addresses, what happens when a file grows beyond this limit? One solution is to reserve the last disk address not for a data block, but instead for the address of block containing more disk-block addresses as shown following picture.

Picture shows that i-node contains 10 direct addresses and these were 4 bytes each. A block size is 2 KB. If a file use i-node and one extra block to save block information, what world the largest possible file size could be?



block of info = 2^{11} byte / 2^2 byte = 2^9 byte

2^9 + 10 = 522 blocks

522 · 2kB = 1044KB = ✓

3. (4 pt.) In the file system, two methods are widely used to keep track of free blocks: a linked list and a bitmap. Let's say a block size is 8-KB and 64-bit disk block number in a file system. 8KB = 8 × 2^{10} × 8 bit = 2^{16} bit

a. How many maximum blocks are needed for keep track 128-GB disk with linked list?

block = 2^{16} bit $/ 2^{6}$ bit $= 2^{10} - 1 = 1023$

#blocks = $2^{7} \cdot 2^{30} / 2^{3} \cdot 2^{10} = 2^{24}$ ✓

$2^{24} / 1023 = \underline{16401}$

b. How many blocks are needed for keep track of 128-GB disk with bitmap?

#blocks $= 2^{7} \cdot 2^{30} / 2^{3} \cdot 2^{10} = 2^{24}$

$2^{24} / 2^{16} = \underline{2^{8} \text{ blocks}}$ ✓

c. (1 pt.) What is the maximum disk size supported by the operating system?

$2^{64} \cdot 2^{13} = 2^{77}$ Byte ✗

or 2^{30} bits

4. (2 pt.) Free disk space can be kept track of using a free list or bitmap. Disk addresses requires D bits. For a disk with B blocks, F of which are free, state the condition under which the free list uses less space than the bitmap. For D having the value 16 bits, express your answer as a percentage of the disk space that must be free.

free list store free block in block since it is linked list, it only need to find initial block to find rest

F.L = F · D

when F · D < B, free list uses less space

bit stores in all block

B.M. = B

if D = 16, F · 16 = B ⇒ $16 = \frac{B}{F}$ ⇒ $\frac{1}{16} = \frac{F}{B}$ ✓

it requires .0625 or 6.25%