1

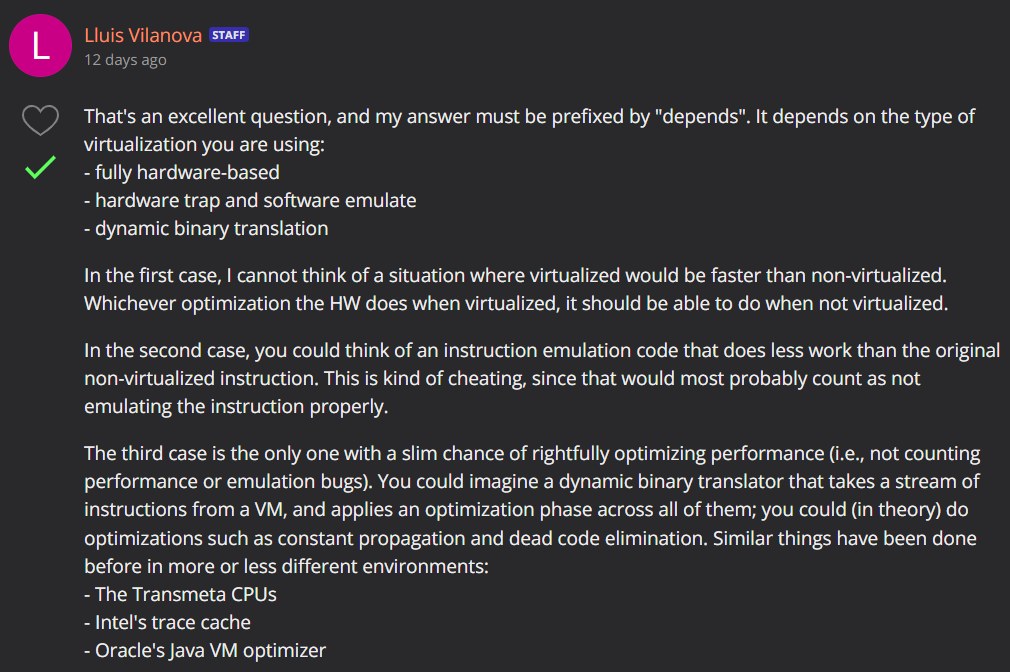
a)

i)

1. Legacy software - software produced for older OSes may be incompatible/buggy on newer ones.
2. Software Development and Testing – Can test system crashing errors without bringing the machine down
3. Consolidation – running many servers on one

ii) Binary Translation dynamically translates trap-inducing instructions into the code to resolve them. This is not applied to user code, and only to code of guest OS because only the Guest OS should be priveleged enough to execute trapping instructions

iii)



bi) p is a constant pointer to a pointer to int

ii) q is a constant pointer to a pointer to a const int

iii) p is a const pointer to a const int pointer that is cast to an ‘a’ pointer to a char pointer

c)

i) T1 calls B.s() which calls signal(c). Nothing is in the queue so nothing happens. T2 calls B.w() which calls wait(c). T2 blocks.

ii) T2 calls B.w() which calls wait(c). T2 blocks and is added to the queue. T1 calls B.s() which calls signal(c). T2 unblocks.

d)

i) Capability

A person has a key, it can be passed between different people

ii) ACL

The safe stores the iris patterns of authorised users

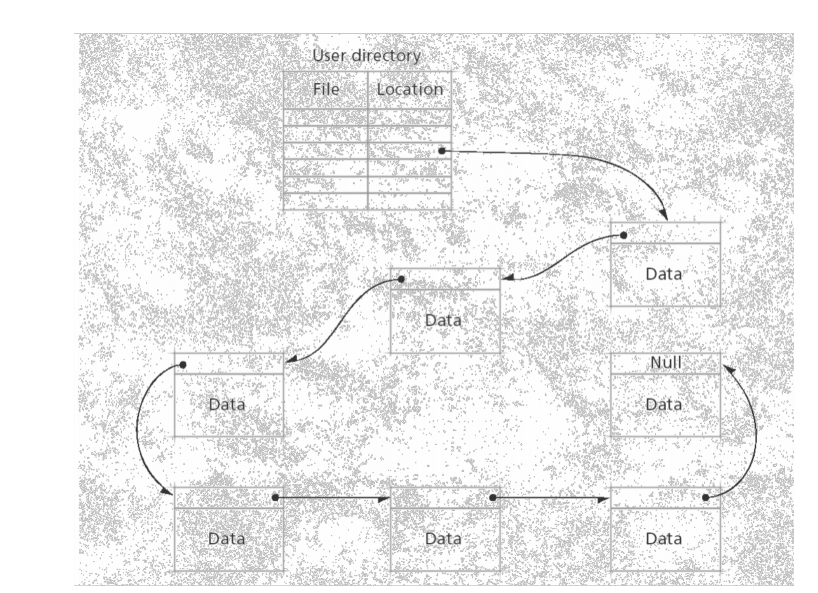
iii) Capability

Same as a key, but knowledge based

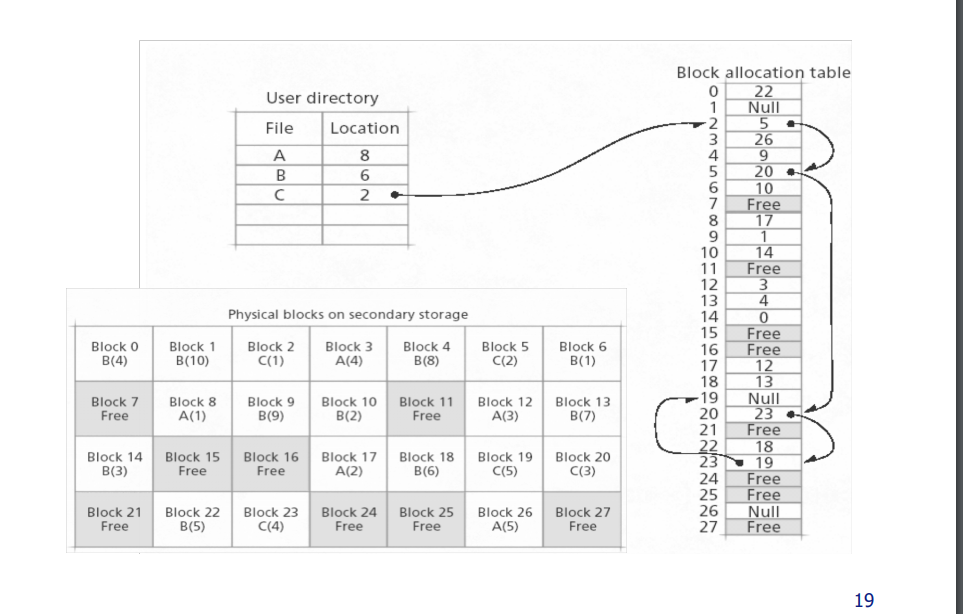
2)

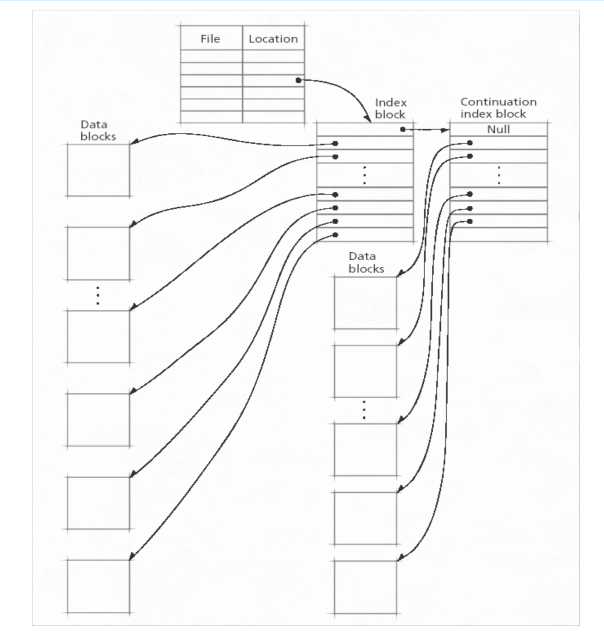
a)

Block linkage - Filesystems page 16



Block allocation table - Filesystems page 18

Index blocks - Filesystems page 26



b)

Using block chaining has the greatest overhead since you need to traverse the list until you find the block you want to delete.

Using index blocks, there is less overhead since all searching may take place within the index block, and there are less continuation blocks than in block chaining.

Block allocation tables have the least overhead.

c)

i) If you’re using block chaining to access the last block you need to access the block in the following order (And the corresponding cylinders for the seek calculation):

Cylinder 0(to get the metadata about where the file starts), block 221, block 2, block 287, block 42 and finally block 187.

Assuming that no other read is waiting for scheduling.

Going to 0 (Cylinder 100 to 0): 2 + 0.02 \* 100 = 4

Going from 0 to 221 (Cylinder 0 to 280): 2 + 0.02 \* 280 = 7.6

Going from 221 to 2 (Cylinder 280 to 23): 2 + 0.02 \* 257 = 7.14

Going from 2 to 287 (Cylinder 23 to 134): 2 + 0.02 \* 111 = 4.22

Going from 287 to 42 (Cylinder 134 to 250): 2 + 0.02 \* 116 = 5.32

Going from 42 to 187 (Cylinder 250 to 20): 2 + 0.02 \* 230 = 6.6

Total is 4 + 7.6 + 7.14 + 4.22+ 5.32+ 6.6 = 33.88 ms

ii)

We first access the block 0 for metadata, we then access block 187 directly:

Going from 100 to 0: 2 + 0.02 \* 100 = 4

Going from 0 to 187: 2 + 0.02 \* 20 = 2.4

Total is 4 + 2.4 = 6.4

d)

increase block size so we can store one file in less block thus reduce the seek time

increase the rotation speed so that we reduce the transfer time and rotation delay

Do block compaction so that blocks are closer together/on the same cylinder, reducing external fragmentation