

Answer Set 2

20 FEBRUARY, 1995

1. *What would be an optimal hash function, should it use logical device number?*

As long as the hit rate times the average time to scan the hash bucket is less than the access time of the disk, the hash function is giving improvement. It is unlikely that the logical device number would be useful in the hash function.

2. *Why does getblk raise the processor priority before manipulating the free list.*

The most important thing to mention here is that the contention is specifically between getblk, which is running in the kernel state, and brelse, which could be run during an I/O interrupt. Both getblk and brelse manipulate the free list, so this is a critical region.

3. *Why does the algorithm getblk raise the processor priority before checking if a block is busy.*

This is a difficult question. Recall that getblk and brelse can find themselves running simultaneously.

The algorithm brelse manipulates the free list: this is what caused the problem for the previous exercise. It also generates wake-ups for processes which went to sleep on busy blocks. It could happen that getblk will miss its wakeup if it does not complete its call to sleep after it has decided to do so.

4. *When should an invalid buffer appear on a hash queue?*

All buffers must be on queues. The possibility exists for I/O errors. If all buffers are on queues, then so must be buffers returned during I/O errors. Also, the interrupt process *absolutely must* release its buffer, no matter if the I/O went through or not. Else the buffer will be dead. (See end of first paragraph on page 46.)

5. *What happens when a delayed write block is taken: a) from the hash queue and b) from the free list.*

A delayed write block taken from the hash queue requires no special processing; if taken from the free list the block must now be written to disk.

6. *How can you prevent starvation in the buffer allocation algorithm.*

It is possible to assign buffers on a first-come, first-served basis to sleeping processes. However this is dangerous from the point of view of deadlock. The designers of Unix decided to error on the side of deadlock-free, starvation possible.