Problem 1

No, a reference count would not be able to trace back where each of the references is being made from. In order to implement a strong remove, we would need to have reverse links back to the references.

Problem 2

Copying a file does not allow changes to be seen by the other copy, and uses twice as much space. Sharing introduces concurrency problems.

Problem 3

Remote file systems allow you to centrally locate, manage and service your storage. However, in addition to the methods by which local storage can fail, remote storage can also fail as a result of network problems, or can crash, or be scheduled for shutdown.

Problem 4

1.
   - Contiguous: 2N+1 (moving all blocks down one by reading then writing)
   - Linked: 1, write block
   - Index: 2, write block, change index.
2.
   - N+1
   - $\frac{N}{2}$ +1
   - 2
3.
   - 1
   - N+2
   - 2
4.
   - 1
   - 1
   - 1
5.
   - 1
   - $\frac{N}{2}$
   - 1
6.
   - 1
   - N + 1
   - 1

Problem 5

1. Option three requires deciding between many more sizes of extents, therefore it is the most complex. Option one has no decisions to make, so it is the least complex.
2. Option one is the best at minimizing external fragmentation, because there are never external fragments. Option three is the worst at minimizing external fragmentation.
3. Option three is the best at minimizing internal fragmentation, because allocations can more closely conform to actual file requirements. Option one is the worst because it cannot.

Problem 6

1. $1.031 * 10^8$ K blocks.
2. 103 GB, 79 MB, 215.1 KB

Problem 7

The operations yet to be performed on the data are recorded, so the system can undo partial changes as it recovers.

Problem 8

When assigining priority to interrupts, the designer has to consider which interrupts are most time criticaland assign those higher priority than those with looser timeframes.

Problem 9

Memory mapped IO is generally faster than using the CPU for IO, but DMA locks the CPU out of the memory bus while the device is writing into memory.

Problem 10

1. A mouse should use interrupt driven IO, without buffering, spooling or caching.
2. A tape drive should use a spooled, interrupt driven IO.
3. A disk drive should use a cached, buffer, interrupt driven IO.
4. A GPU should use interrupt driven IO, without buffering, spooling or caching.

Problem 11

No, the operating system accesses memory through physical addresses.

Problem 12

Their has to be some level of translation between the virtual addresses and the physical address done in the kernel space.

Problem 13

Cannot be done, as it would require the file permisionings to have two different orderings.

Problem 14

1.
   - O(M*N)
   - O(M)
   - O(M*N)
2.
   - O(1)
   - O(M)
   - O(N)
3.
   - O(M)
   - O(N)
   - O(1)
4.
   - O(M)
   - O(1)
   - O(1)

Problem 15

The matrix requires looking up a spot in the matrix, the list requires searching the list.

Problem 16

False. Sometimes bugs can expose security holes.