CS 212: Final Review Worksheet 2

Question 1: One process opens a file that is stored on a hard disk and reads the data in the file, while another process opens a virtual file and reads the data in the virtual file. What differences are there (if any) in the code used by these two processes for opening and reading these two different types of files? What differences are there (if any) in the code that is run by the kernel in these two scenarios?

Question 2: Imagine you are writing a multithreaded application and `shared_data` is a pointer to a structure shared by all threads. One of the threads occasionally reports the progress of the work being done by printing the value of the member `x` from this shared structure like this:

```
double x = shared_data->x;
printf("The current value of x is %lf\n", x);
```

Does this code need to be protected by a lock? If so, which portion of it needs to be locked?

Question 3: Now imagine a similar scenario to the previous question, but you need to report the sum of two of the members of the structure:

```
double sum = shared_data->x + shared_data->y;
printf("The current sum of x and y is %lf\n", sum);
```

Does this change anything about locking requirements?

Question 4: Say a process writes data to a new file. In addition to the data contained in the file, what additional data needs to be written to the disk?

Question 5: Is it necessary for the blocks of a file to be stored in contiguous addresses on a disk? Why or why not? If it is not necessary, is it at all advantageous?

Question 6: Imagine a system is booting up, and it has a file system that uses journaling. The OS checks the journal and finds that there is a complete transaction in the journal. Does the OS need to take any action at this point? If so, what should the OS do?