

Answers

1. The reason behind using *open* call is because of accessing it with different parameters. There are different versions of *open* system call such as *freopen()*, *openat* and etc. If we didn't have the *open* call, we had to specify the path for opening the file which also requires fetching the i-node for that file. The main problem with this is not knowing the time for flushing the i-node back to the disk. As a consequence, we would have a more messy and slower system which could still work.
2. For the user it looks like the same thing; however, for the operating system there are differences between these two system calls. The purpose behind *rename* is that it deletes the first file and copies it into the new file. *rename* and *copy* are similar, however, there are differences as well. The fundamental difference is that *rename* deletes the first file by creating the new one, however, *copy* copies it into the new one and there can be memory issues if the disk is full. Second difference is that when we *rename* the file it doesn't change the creation time for the file.
3. To mention that, it is crucial for contiguous file system there are no any differences if one single data block is corrupt; however, there are also disadvantages and negative sides of this. First of all, it is impossible to open the corrupted file which means data block that has been allocated for it is wasted and it should be deleted from the system. The job for the system to be done is to get that corrupted file and remove it.

However, for the linked file system if one file is corrupted it can affect to other files as well because of loss reference. For instance, if one data block is inaccessible, so it can also be mentioned that other data blocks become inaccessible as well that brings the failure in the system.

The final way is called indexed or table based which means it is impossible to have an effect on the whole list if one file is corrupted. The reason behind is that data blocks in indexed file systems are not allocated contiguously. Although it doesn't make any problem, problems can arise when the number of corrupt files is large.

4. If we show the 4 terabytes (TB) with bytes (B) it will be 4^{21} which gives us 2^{42} . If we do the same thing with 4 kilobytes it will be 2^{12} which gives us 2^{30} block numbers in one block. Thus, the answer is $2^{30} - 1 = 1,073,741,823$ block numbers are available.

5. If we say that all the files are 1 kilobytes (KB) and the block size for the file is 4 kilobytes, then we can calculate the wastage by dividing them which is $\frac{1}{4} = 75\%$ which means it will contain 3 kilobytes of wasted space. In real life, this one can hardly happen because the files use the disk more effectively.