

Q2:

a) When the command "cat testfile" was run, the read system call was called which changed the access time (atime) to September 21st, 12:10pm. Because the file was changed with a command that calls read instead of utime, ctime was also changed to September 21st, 12:10pm. When the touch command was used to change modification time, mtime was changed to September 19th, 12:00 am.

b) The fsck program must be run in order to mount this volume because the volume was not explicitly unmounted. fsck will determine that the filesystem is inconsistent and might find inodes with no corresponding pathnames. If this is a 2TB volume with 1,000,000 allocated inodes this will take a long time because fsck recursively checks that every file and directory (i.e. their pathnames) correspond correctly with the inode table and free block map. There are plenty of unused nodes that correspond to unallocated space that need to be visited on top of the 1,000,000 inodes that need to be individually checked which makes this program very slow. If the system is a journalling filesystem, the transaction IDs in the journal are examined from oldest to newest and either there is a BEGIN entry without a COMMIT entry or there is a BEGIN entry and COMMIT entry. In the first case, no valid action can be taken because the transaction never happened. In the second case, the transaction is executed again and the blocks are rewritten in the same places in the filesystem.

c) When the read system call is used to open the video file, a copy of the contents of the regularly accessed segment of the file is kept in the kernel memory (i.e. it is cached). These areas of memory are in 4k chunks, meaning that a small video segment that is frequently accessed is easily accessible. As a result, the first file read system call on the video takes a long time and explains the laggy-ness. However, once that video segment has been watched once already, the data is copied and cached so the next read of that same video segment is much faster. However, because the whole video was not cached, when starting from the beginning of the video, the read system call must be called again as the first time this data is being read, so it is laggy again. The file data cache mechanism is being utilized in this example.

d) In the first execution of the mv command, the file is moved to a different directory (/A/B/C) and renamed to Q. This just requires a call to rename() and executes quickly. The reason the second execution of the mv command takes so long is because /A/B/C is a different volume from /A/B/X. This means that the mv command is executing "cp /A/B/C/Q /A/B/X/Y" then "rm /A/B/C/Q" and the utime, chown and chmod system calls are used to change all the metadata in the new path to be the same as the metadata in the old path. This is a much more time consuming process than the previous execution of mv which just called rename() because it was moving/renaming a file from one location on a volume to another location on the same volume.