

Problem 1



Problem 2

A)

- i) One will have to run “fsck” and check the file system, making sure that there are no corruptions and correct it if there are any.
- ii) One example is the ghost inode that is not linked but yet not free from the bitmap.
- iii) Since “fsck” will have to go through both used and unused inodes, not having the journaling, it will take a long time.

B)

The first possibility is that the current user does not have the write permission for the directory. By triggering the unlink(2) system call for the file “foo”, we are modifying the data block of the directory “dir”. However, in order to modify the contents, a write permission for the directory would be needed. And without that permission, we will see the EACCES error.

Another possibility is if the current user does not have the execute permission for the directory, “dir” or the root directory “/”. Without the execute permission, one will not even be able to reference the directory, which would mean that one will not be able to access the “foo” file.

C)

One reason is because the UNIX system divides the volume of a file system into 4 distinct areas: Header(superblock), Inode Table, Free Map, and Data Blocks. While the videos will be stored within the Data Blocks, data for the entire volume and the metadata is within the other sections, which will use some portion of the disk’s storage.

Another reason would be the reserve factor. Since the SATA disk is a HDD, it would have fragmentations in the data blocks. In order to decrease the disk’s seeking activities, most HDD systems use a reserve factor of 5~10%. And due to this reserve factor, the user will have noticeably less videos stored.

D)

One should acknowledge the fact that the “Z” directory might be within a different volume from A, B, and C. Although not defined in the question, the Z directory could’ve been previously mounted, which means that, in order to move the file, we would have to go through a different volume, which might take quite a while.