Zidane Karim 9/22/2025
ECE357 Problem Set 2

1a)

i) **Fsck –** examines the filesystem by sector to "clean" the superblock. It does this by recursively checking each node of the graph for consistency errors. An example of a consistency issue it can find is finding an inode that has no specific pathname. When we run "rm" on a UNIX based OS, we are just decrementing the nlink count; so an inode with 0 nlinks is difficult to exactly place. The program solves this problem by placing it in a lost+found directory in the root directory. Another example is the computer shutting down during a file write, and inconsistencies within finding that path name again.
ii) It would not entirely avoid these issues since the computer can still shut dow/crash in the middle of a write operation for example, and data can still be corrupted. This would in turn still possibly create inconsistencies despite the fact that there is no race condition between cache and the disk

1b) The file system reserves space for things such as the root directory, and journals, so some of that space is unusable for videos. It also reserves 5-10% to prevent fragmentation.

Another reason would be that Linux on EXT4 volumes uses an extent based block system. This reserves 12 bytes per header alone, so this starts to add up across multiple blocks. Then, for example a 4K disk block wastes 4 bytes at the end, which also adds up. Multiple of these additions across blocks ends up taking space away from videos.

Furthermore, if a file component takes up less than the size of a block, it will still reserve the entire block, consuming disk space.

1c) It could be the case that /A/Z/F3 exists on a different filesystem than /A/C/, which in turn would require a complete rewrite into the /A/Z/ subspace, where as /A/B and /A/C could be mounted on the same volume.

1d)

st_size – 3 +1 = 4

st_nlink -1

st_mode – S_IFREG

st_atime - September 24th, 2025, 4:55PM

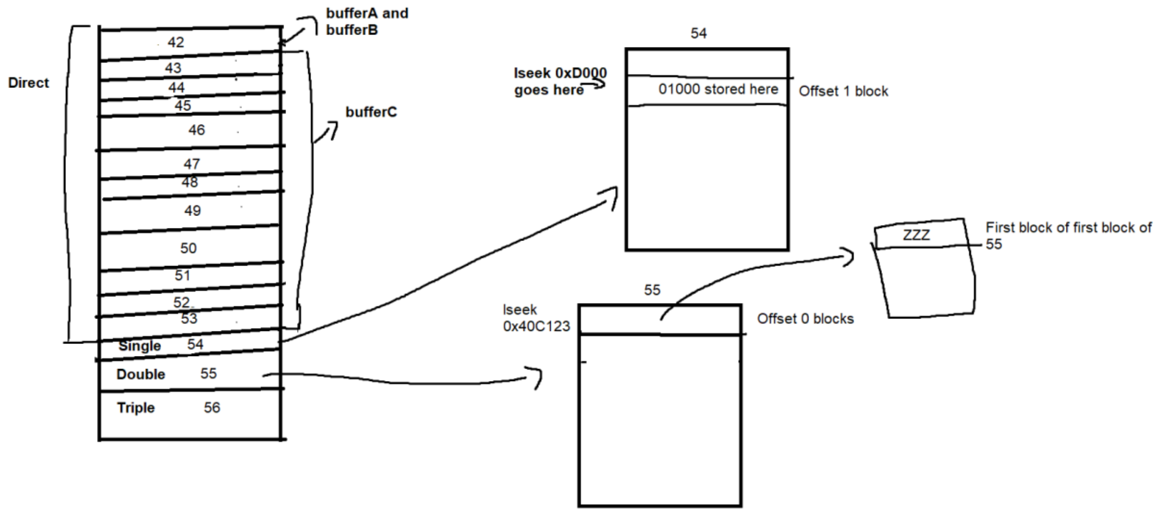
st_mtime –September 18th, 2025, 10:30 PM

st_ctime - September 24th, 2025, 4:30PM

Zidane Karim
ECE357

9/22/2025
Problem Set 2

2a)



2b)

st_size = 0x40C123 + 3 + 1 = 0x40C127, since the files are sparse we can just look at the offsets

st_blocks = 136

This comes from the fact that we used 12 direct blocks, 2 indirect blocks, and three more “nested” blocks, for a total of 17 4096-size blocks. Since st_blocks is measured in 512-byte units, we can multiply 17 by (4096/512) or 17*8 to get 136 for st_blocks.