# mount_umount.c

int mount (char *filesys, char *mntPoint)
// ask for filesys + mntpoint in main

If the user didn't supply parameters, I display all current mounted systems from the global Mtable array. Next, I run through the MTable array to check if the filesys is already mounted. If it is, I return with an error, if not I allocate a new entry in the Mtable array for our filesys were trying to mount. Next, I open the filesys for Read/write under linux and check to make sure filesys is an EXT2 filesystem. I load the mntPoint DIR into a MINODE and check it is a DIR and not currently Busy. Next, I set the mount table entry information through a local mountptr pointer including name, dev, etc.. Finally, I mark the minode for mntpoint as mounted on and point to the Mtable entry. Return 0 for success.

int umount (char *filesys)

First, I search through the global Mtable array to check that user provided filesys is a currently mounted system. I point a local Mounttable entry(mnt) at the Mtable entry in the array, return error if the filesys isn't mounted. I then make sure the mnt->dev isn't the same as the cwd->dev. If so, I return error. I also run through all open files to make sure none of them belong to our filesys (same dev number). If any do, I return an error. Lastly, I reset the mount_point's INODE dev to 0 and mounted flag to 0 before calling iput and returning 0 for success.

## ① Downward Traversal

- when traversing /a/b/c/x, once we reach /a/b/c, we should see that the minode has been mounted (mountflag =1). we must
1) Follow minode's mntPtr to locate mount table entry
2) From mount table's dev number, get root (ino=2) INODE into memory.
3) Then continue to search for X under root Inode of mounted device.

## ② Upward Traversal

- Assume we're at /a/b/c/x + traversing upward to . which will cross mount point /a/b/c. When we reach root Inode of mounted filesys, we see (ino=2) but dev is different than real root. Using its dev number, we locate its mount table entry which points to /a/b/c/. we switch to minode of /a/b/c/ + continue the upward traversal.