```
#include "type.h"

/*
Gets minode of pathname
Allocates new OFT, sets OFT's minode to what you just loaded
Assigns that OFT to the first empty OFT in running->OFT
*/

int open_file(char path[124])
{
        int i, ino, fd = 0, perm = 0, mask = 0, mode = -1, offset = 0;
        MINODE *mip;
        INODE* ip;
        OFT* ofile = NULL;


        char buf[1024];

        if (!strcmp(third, ""))//checks to see if a mode was entered
        {
                printf("No open mode specified.\n");
                return;
        }

        //this sequence of if and else if's is used to determine what mode the user
entered
        if (!strcmp(third, "0")) //read
                mode = 0;

        else if (!strcmp(third, "1")) //write
                mode = 1;

        else if (!strcmp(third, "2")) //RW
                mode = 2;

        else if (!strcmp(third, "3")) //append
                mode = 3;

        else
        {
                printf("Invalid mode.\n");
                return;
        }



        if (path[0] == '/')
                ino = get_Inode(root, path);//gets the inode of the root since the
first character of path was a /

        else
                ino = get_Inode(running->cwd, path);//gets the inode of the path that
was entered

        if (ino == 0)//check that file exists
        {
                printf("Error, no such file exists.\n");
                return;
        }


        mip = iget(dev, ino);//gets the memory inode of the file that we want to open
        ip = &mip->INODE;//sets the inode pointer the memory inode


        if (!S_ISREG(ip->i_mode))//checks whether or not it is a file because we cant
open none files
        {
                printf("Error, not a file.\n");
                iput(mip);//disuper_poses of the memory inode
                return;
        }

        if(checkDuplicates(mip))
        {
                printf("Error, the file is already being used.\n");
                iput(mip);
                return;
        }

        iput(mip);//dispose of memory inode


        //loop OpenFileTable
        for (i = 0; i < NOFT; i++)
        {


                if (i == NOFT - 1)//at the end of the open file table without finding
a spot
                {
                        printf("No room exists in the open file table.\n");
                        iput(mip);//dispose of memory inode
                        return;
                }
                ofile = &OpenFileTable[i];
                if(ofile->refCount == 0)
                {
                        ofile->refCount++;
                        break;
                }


        }

        ofile->mode = mode;
        ofile->inodeptr = mip;

        switch(mode)
        {
                case 0: ofile->offset = 0;
                        break;
                case 1: clearBlocks(mip);
                        ofile->offset = 0;
                        break;
                case 2: ofile->offset = 0;
                        break;
                case 3: ofile->offset = mip->INODE.i_size;
                        break;
        }

        for (fd = 0; fd < NFD; fd++)//walks through to find an open spot
        {
                if (running->fd[fd] == NULL) //found a spot that isnt occupied
                {
                        running->fd[fd] = ofile;
                        break;
                }

                if (fd == NFD -1)//checks to see if we are at the end of the file
descriptor array, index 9
                {
                        printf("No open spots remain.\n"); //no spots left
                        iput(mip);
                        return;
                }
        }
        return fd;
        iput(mip);
}
```