```c
#include <stdio.h>
#include <stdlib.h>
#include "type.h"

#include "util.c"
#include "alloc_dealloc.c"
#include "pwd.c"
#include "mkdir.c"
#include "rmdir.c"
#include "creat_file.c"
#include "chmod_file.c"
#include "touch_file.c"
#include "mount_root.c" //includes cd and ls
#include "link.c"
#include "open.c"
#include "close.c"
#include "pfd.c"
#include "symlink.c"
#include "unlink.c" //using it at rm also
#include "stat.c"
#include "cp.c"
#include "mv.c"
#include "lseek.c"
#include "read.c"
#include "write.c"
#include "cat.c"
#include "menu.c"



int main()
{
    char *commands[23] = {"mkdir", "rmdir", "ls", "cd", "pwd", "creat", "link",
"unlink", "symlink", "stat", "chmod", "touch", "quit", "open", "close", "pfd",
"write", "cat", "read", "cp", "mv", "lseek", "menu"};
    char device_name[64], input[128] = "";
    char command[64], pathname[64] = "";
    char line[256];

    int (*functions[23]) (char path[]);
    int i;



    functions[0] = make_dir;
    functions[1] = remove_dir;
    functions[2] = ls;
    functions[3] = cd;

    functions[4] = my_pwd;
    functions[5] = creat_file;
    functions[6] = my_link;
    functions[7] = my_unlink;

    functions[8] = my_symlink;
    functions[9] = my_stat;
    functions[10] = chmod_file;
    functions[11] = touch_file;
    functions[12] = quit;
    functions[13] = open_file;
    functions[14] = my_close;
    functions[15] = my_pfd;
```

```
    functions[16] = my_write;
    functions[17] = my_cat;
    functions[18] = read_file;
    functions[19] = cp_file;
    functions[20] = mv_file;
    functions[21] = my_lseek;
    functions[22] = menu;


    init();//intializes everything before the disk gets mounted


    printf("Please enter the disk image to open: ");
    fgets(line, 256, stdin);//saves the disk image to open

    line[strlen(line)-1] = 0;//sets last character to zero(null)

    mount_root(line);//mounts the disk that the user entered

    printf("\n");

    menu();//print menu to user

    printf("\n\n");

    while(1)
    {
        printf("Input command: ");
        fgets(input, 128, stdin);//saves what the user entered into input

        input[strlen(input) - 1] = 0;//sets the last character of the input array to
zero(null term)
        if(input[0] == 0)
            continue;

        sscanf(input, "%s %s %s", command, pathname, third);//breaks the command apart

        for(i = 0; i < 23; i++)//loops through the function table
        {
            if(!strcmp(commands[i], command))//checks the function table for the
function
            {
                (*functions[i])(pathname); //calls it then breaks out of the loop
                break;
            }
        }

        if(i == 23) //if looped through whole function table and nothing was found
then tell the user that they entered an invalid command.
            printf("Error, invalid command.\n");

        //resets the strings that get data read into them
        strcpy(pathname, "");
        strcpy(command, "");
        strcpy(input, "");
        strcpy(third, "");
    }
    return 0;
}
```