```c
#include "type.h"

/*
Loads inode into memory
Prints out information of inode
Puts inode back on disk
*/

void my_stat(char *path)
{
    int ino;
    MINODE *mip;
    INODE *ip;
    char *cp;
    char buf[1024];
    char name[64];
    char *my_atime;
    char *my_mtime;
    char *my_ctime;
    DIR *dp;

    ino = get_Inode(running->cwd, path);//gets the inode of the file that we want to
stat
    mip = iget(dev, ino);//gets the memory inode of the file

    strcpy(name, basename(path));//copies the basename of the file into name

    ip = &mip->INODE;//sets the inode pointer to the memory inode pointer

    printf("  File: %s\n", name);//prints the name
    printf("  Size: %d\tBlocks: %12d ", ip->i_size, ip->i_blocks);//prints the sizes

    if(S_ISDIR(ip->i_mode))//checks to see if it is a dir then prints if it is
        printf("  Directory\n");

    else
        printf("  File\n");//else prints that it is a file

    printf("Inode: %d Links:%d \n", ino, ip->i_links_count);//prints the links that
the file has

    my_atime = ctime( (time_t*)&ip->i_atime);//gets the times on the file
    my_mtime = ctime( (time_t*)&ip->i_mtime);
    my_ctime = ctime( (time_t*)&ip->i_ctime);

    printf("Access: %26s", my_atime);//prints the times out
    printf("Modify: %26s", my_mtime);
    printf("Change: %26s", my_ctime);

    mip->dirty = 1;//sets dirty to 1 for the memory inode
    iput(mip);//disuper_poses of the memory inode

    return;
}
```