

```

#include "type.h"

/*
Get parent of dest argument
Allocate new inode for child
Enter child dir entry
String copy first argument into dest iblock_array from inode just allocated
*/

void my_symlink(char *path)
{
    int ino, i;
    int link_ino;
    int parent_ino;
    char temp[64], parent[64], child[64];
    char file1_name[64];

    MINODE *mip;
    MINODE *p_mip;
    MINODE *link_mip;

    INODE *ip;
    INODE *parent_ip;
    INODE *link_ip;

    strcpy(temp, path);
    strcpy(file1_name, basename(temp)); //gets the basename of the file that needs the
link

    //file1 file
    ino = get_Inode(running->cwd, path); //gets the inode of the file that we want to
make the link for
    mip = iget(dev, ino); //gets the memory inode for the file that we want to make
the link for

    if(strlen(path) >= 60) //checks to see if the name is too long
    {
        printf("Error, the name is too long.\n");
        return;
    }

    if(!mip) //checks to see if we could find the file in memory
    {
        printf("Error, %s does not exist.\n", path);
        return;
    }

    strcpy(temp, third);
    strcpy(parent, dirname(temp));
    strcpy(child, basename(third));

    parent_ino = get_Inode(running->cwd, parent); //gets the parent inode of the file
    p_mip = iget(dev, parent_ino); //gets the parent memory inode of the file

    if(!p_mip) //checks to see if parent memory inode exists
    {
        printf("Error, cannot get parent mip.\n");
        return;
    }

    if(!S_ISDIR(p_mip->INODE.i_mode)) //checks to see if the parent is a dir
    {

```

```
        printf("Error, parent is not a directory.\n");
        return;
    }

    if(get_Inode(running->cwd, child) > 0) //child doesn't already exist
    {
        printf("[!] ERROR: %s already exists.\n", child);
        return;
    }

    link_ino = creat_helper(p_mip, child); //creates the file for the link
    link_mip = iget(dev, link_ino); //gets the memory inode for the new file
    link_ip = &link_mip->INODE; //sets the link inode to the memory inode pointer

    link_ip->i_mode = 0120666;
    link_ip->i_size = strlen(file1_name);

    link_mip->dirty = 1;
    iput(link_mip); //disuper_poses of the link memory pointer
    iput(mip); //disuper_poses of the file memory pointer
}
```