```c
/* ANRC RHKI */
/* Lab12: Syscalls Lab */
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/unistd.h>
#include <linux/semaphore.h>
#include <linux/mm.h>
#include <linux/highmem.h>
#include <asm/unistd.h>

#define DRIVER_AUTHOR "ANRC"
#define DRIVER_DESC   "Lab12"

MODULE_LICENSE("GPL");           // Get rid of taint message by declaring code as GPL.

/*  Or with defines, like this: */
MODULE_AUTHOR(DRIVER_AUTHOR);    // Who wrote this module?
MODULE_DESCRIPTION(DRIVER_DESC); // What does this module do?

int init(void);
void cleanup(void);

int set_page_rw(long unsigned long _addr)
{
        unsigned int level;
        pte_t *pte = lookup_address(_addr, &level);
        if(pte->pte &~ _PAGE_RW) pte->pte |= _PAGE_RW;
        return 0;
}

void **sys_call_table;

asmlinkage int (*original_call) (const char*);

//asmlinkage long sys_unlink(const char *pathname)
asmlinkage long my_sys_unlink(const char *pathname)
{
   printk("unlinker: File unlink attempted and stopped!\n");
   //return original_call(pathname);
   return -1;
}

int init_module()
{
    // sys_call_table address in System.map
        //[user@localhost labs]$ cat /boot/System.map-2.6.32-358.el6.x86_64 | grep sys_call_table
        //ffffffff816004a0 R sys_call_table
        //ffffffff8160a5f8 r ia32_sys_call_table

        /* get address of sys_call_table */

        /* change page attributes to RW */

        /* replace function call address of unlink */

    return 0;
}

void cleanup_module()
{
   // Restore the original call to unlink
}
```