```c
/* ANRC RHKI */
/* Lab15: Page Cache */
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

#include <linux/kprobes.h>
#include <linux/pagemap.h>

#define DRIVER_AUTHOR   "ANRC"
#define DRIVER_DESC     "Lab15b"

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?

/* globals for stats */
int total_cache_add     = 0;
int total_cache_remove  = 0;

void handler_post_add(struct kretprobe_instance *ri, struct pt_regs *regs, unsigned long flags)
{
        total_cache_add++;
}

void handler_post_remove(struct kretprobe_instance *ri, struct pt_regs *regs, unsigned long flags)
{
        total_cache_remove++;
}

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

static struct kprobe kp_add = {
        .pre_handler = NULL,
        .post_handler = handler_post_add,
        .fault_handler = NULL,

        .addr = (kprobe_opcode_t *) add_to_page_cache_lru,
};

static struct kprobe kp_remove = {
        .pre_handler = NULL,
        .post_handler = handler_post_remove,
        .fault_handler = NULL,

        .addr = (kprobe_opcode_t *) delete_from_page_cache,
};

int init(void)
{
        register_kprobe(&kp_add);
        register_kprobe(&kp_remove);
        printk("cachetest: kprobes add_to_page_cache()/delete_from_page_cache() registered\n");
        return 0;
}

void cleanup(void)
{
        unregister_kprobe(&kp_add);
        unregister_kprobe(&kp_remove);
        printk("cachetest: kprobes unregistered\n");

        /* tally up the totals and % */
        printk("cachetest: %d calls to add_to_page_cache() -- %d calls to delete_from_page_cache() \n",
```

```
                        total_cache_add,
                        total_cache_remove);
}

module_init(init);
module_exit(cleanup);
```