```
/**
 * seconds.c
 */
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/proc_fs.h>
#include <linux/jiffies.h>
#include <asm/uaccess.h>
#include <asm/param.h>

#define BUFFER_SIZE 128
#define PROC_NAME "seconds"
#define MESSAGE "Hello World\n"

unsigned long jiff;
unsigned long startJiff;
unsigned long currentJiff;
unsigned long elapsedJiff;
unsigned long seconds;
int hz;

ssize_t proc_read(struct file *file, char *buf, size_t count, loff_t *pos);
static struct file_operations proc_ops = {
        .owner = THIS_MODULE,
        .read = proc_read,
};
/* This function is called when the module is loaded. */
int proc_init(void)
{
        proc_create(PROC_NAME, 0, NULL, &proc_ops);
        printk(KERN_INFO "/proc/%s created\n", PROC_NAME);

   //Sets the value of startJiff to jiffies when the module is loaded
   startJiff = jiffies;
return 0;
}

/* This function is called when the module is removed. */
void proc_exit(void) {
        // removes the /proc/hello entry
        remove_proc_entry(PROC_NAME, NULL);
        printk( KERN_INFO "/proc/%s removed\n", PROC_NAME);
}


ssize_t proc_read(struct file *file, char __user *usr_buf, size_t count, loff_t
*pos)
{
        int rv = 0;
```

```
        char buffer[BUFFER_SIZE];
        static int completed = 0;
        if (completed) {
                completed = 0;
                return 0;
        }
        completed = 1;

    //Gets the current value of jiffies, subtracts the starting value, and divides by HZ
    currentJiff = jiffies;
    elapsedJiff = currentJiff - startJiff;
    hz = HZ;
    seconds = elapsedJiff / hz;
        rv = sprintf(buffer, "Seconds Elapsed: %lu\n", seconds);

        // copies the contents of buffer to userspace usr_buf
        raw_copy_to_user(usr_buf, buffer, rv);

        return rv;
}
/* Macros for registering module entry and exit points. */
module_init( proc_init );
module_exit( proc_exit );
MODULE_LICENSE("GPL");
MODULE_DESCRIPTION("Jiffies Module");
MODULE_AUTHOR("Austin Biggs");
```