```c
/* ANRC RHKI */
/* Lab11: Signals Lab */
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

#include <linux/delay.h>
#include <linux/kthread.h>

#include <asm/signal.h>
#include <linux/delay.h>

#define DRIVER_AUTHOR "ANRC"
#define DRIVER_DESC   "Lab11"

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?

struct task_struct *ts;

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

int thread(void *data)
{
        struct task_struct *task;
        int last_pid = 0;
        unsigned long delay = jiffies + 5*HZ;   /* 5 second delay */

        while(1)
        {
                for_each_process(task)
                {
                        /* find a sleep process */

                        printk("signalex: Found siguser process [%d], sending signals\n", task->pid);

                                /*
                                 * #define SIGABRT              6
                                 * #define SIGTRAP              5
                                 * #define SIGQUIT              3
                                 * #define SIGKILL              9
                                 */

                        /*  int send_sig_info(int, struct siginfo *, struct task_struct *); */

                        /* terminate the process after 5 seconds*/
                }

                msleep(100);
                if (kthread_should_stop())
                        break;
        }

        return 0;
}


int init(void)
{
        printk(KERN_INFO "signalex: init_module() called\n");
        ts = kthread_run(thread, NULL, "kthread");

        return 0;
```

```
}

void cleanup(void)
{
   printk(KERN_ALERT "Unloading signalex ...\n");
   kthread_stop(ts);
}

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