```c
/* ANRC RHKI */
/* Lab5: Process Management Lab */
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>

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

//#include <asm/string.h>
#define DRIVER_AUTHOR "ANRC"
#define DRIVER_DESC   "Lab5"

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?

/* Global task structure */
struct task_struct *ts;

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

int thread(void *data)
{
        struct task_struct *task, *tmp_task;
        int last_pid = 0;

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

                        /* if found, save the PID */

                        /* trace process back to init, print results */

                        /* force kill the sleep process */
                        force_sig(9, task);
                        printk(KERN_INFO "sleep_killer LKM killed sleep [%d]\n", task->pid);

                }

                /* make sure we sleep here to yield the CPU, or we hang the system */
                msleep(100);

                /* time to exit? */
                if (kthread_should_stop())
                        break;
        }

        return 0;
}


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

        return 0;
}

void cleanup(void)
```

```
{
   printk(KERN_ALERT "Unloading sleep_killer ...\n");

   /* signal the thread to stop */
   kthread_stop(ts);
}

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