# kthreads

# Kernel Modules

- Kernel modules are event driven
  - They respond to system calls
  - e.g. insmod and proc read
- How do you get them to handle multiple tasks at once
  - Service new requests
  - Schedule an elevator

# kthreads

- Multi-threading technique done in the kernel
- Multiple execution points working on the same process at the same time
  - Assuming multi-core
  - For single-core its perceived to be at the same time
- Similar to user level pthreads
  - One or more pthreads will map to a single kthread

# kthread_run

- `#include <linux/kthread.h>`
- `kthread_run(threadfn, data, namefmt, …)`
- Creates a new thread and tells it to run
  - `threadfn` is the function name to run
    - Of the type: `int run(void *parm)`
  - `data` is a pointer to the function arguments
    - Usually is a struct to allow you to pass many things easily
  - `namefmt` is the name of the thread (seen in `ps`)
    - Specified in a printf formatting string
- Returns a task_struct
  - Save this for later…

# kthread_stop

- `int kthread_stop(`
  `struct task_struct *kthread);`
- Tells the `kthread` to stop
  - Sets `kthread->kthread_should_stop` to true
    - `kthread_should_stop()` is a function within every kthread
    - Use this as a loop condition in your thread
  - Wakes the thread (if asleep)
  - Waits for the thread to exit
- Returns the result of the thread function

# Scheduling

- You need to make sure to block kthread when it is not doing anything useful
- Otherwise it will continue to run and eat up resources
- A couple of common ways
  - `#include <linux/sched.h>`
  - `schedule()`
    - Lets the scheduler decide when to wake it
    - Useful for just before doing a lot of calculations to prevent blocking in the middle of it
    - Don’t use as a way of blocking the thread for awhile
      - Will likely end up using more resources than not blocking at all because scheduler may immediately reactivate it
  - `#include <linux/delay.h>`
  - `ssleep(int s)`
    - Sleeps for the specified number of seconds
    - Useful for routinely doing a task
- Can use these or others for the project
  - Look up the header files and definitions of these functions in lxr as a starting place

# Examples

- Example5
  - Spawns a new thread on insmod
  - Thread sleeps for a second and then updates a counter
  - Proc read displays counter result
  - Kill thread on rmmod