

# Linux Kernel Module Driver for Keyboard LEDs

Bridging the Gap Between the Kernel Space and the  
User Space:

Connor Reeder

# Problem Statement:

The driver which Linux currently uses to activate and deactivate the Caps Lock and Num Lock LED lights on a Toshiba Satellite C55-A5286 currently does not allow for control from user space applications. The functionality of those two lights is bound to the standard functions of Caps Lock mode and Num Lock mode, respectively. Thus, there is no way to repurpose the lights to serve other functions in the event that the user does not use those lights for their current function.

# Idea

- Write a Linux kernel module which containing a driver for the Toshiba Satellite C55-A5286 keyboard LED lights which will replace the one currently in use.
- The driver will mount each of the two LED lights as a linux special file node in the /dev directory so as to allow any user space application to read and write to it like any other device.
- It will be mounted as a character device, thus requiring applications to read and write to it in block-aligned sizes.
- Create a simple user space program which will use the caps lock light as a notification for some type of event.

# Demo Plan

- Break down the procedure that was used to develop the linux kernel module, the risks that were involved, and what exactly it does to carry out its task.
- Show how the interface works between the kernel space module and the user space application, including the device files `/dev/capslight` and `/dev/numlight`, and the Linux commands necessary to develop, install, or remove a module.
- I will demonstrate lighting with `echo {0,1} > {/dev/capslight,/dev/numlight}` and reading with `cat {/dev/capslight,/dev/numlight}` in a bash script that blinks a light when any terminal command fails.
- Present the documentation or specifications that were researched in order to understand the specifics of how to control the LED lights built into the Toshiba Satellite C55-A5286.

# Timeline

