CS 460 -- In Class Lab — Linux Character Driver Module
April 16, 2012 — Blink your Keyboard Lights with text

10 points

If you do not have a working keyboard-light-blinking kernel module, you can download a working version from here:
http://zeus.cs.pacificu.edu/chadd/OnCampus/cs460s12/kbleds_soln.tar.gz

Goal:

Write a character driver that will blink the keyboard lights once for each character in a string written to the device. Reading from the device should return how many times the keyboard lights have blinked since the module was loaded.

Resources:

http://lwn.net/Kernel/LDD3/ (chapter 3)

http://tldp.org/LDP/lkmpg/2.6/html/x569.html

Reuse the ArchLinux virtual machine built in the previous lab.

► Use the chardev.c given in the TLDP page listed above as a starting point. The given module will track how many times the attached /dev file is read from. The module simply printks a message when data is written to the /dev file. The only change you should need to make to the source code is to not capture the return value of unregister_chrdev (which is now a void function).

Upon loading the module, the init_module function will register the character device with the kernel and printk a message explaining the mknod command you need to run to create the /dev file.

Build this module using a Makefile based on the Makefile in the previous lab. Load the module, run mknode, and then test:
cat < /dev/chardev
echo “hello” > /dev/chardev
cat < /dev/chardev
echo “hello” > /dev/chardev
echo “hello” > /dev/chardev
cat < /dev/chardev

Use rmmod to remove the module and delete /dev/chardev.

Notice that any time /dev/chardev is read from or written to, the file is first opened (device_open) and then later released (device_release).

► Create a new module, based on the one above and your previous keyboard-light-blinking module, that will

1) on a WRITE: blink the keyboard lights once for each character in a string written to the device. The write function could block until all the blinks are done or the write function could schedule the blinks to happen in the future with a set of timers. **BONUS**: *Printk the user's string with a nice message. Remember to use copy_from_user() to get data from the user's buffer to an internal buffer in the driver. You may need to use kmalloc() to create a buffer.*

2) on a READ: return a string stating how many times the keyboard lights have blinked since the module was loaded.
"I have blinked the keyboard lights # times!

3) You must decide how to handle two processes accessing the device at the same time. An acceptable solution is to not allow the file to be opened simultaneously. **BONUS:** *Maybe multiple processes can read at the same time? You can investigate how you might use the* ***loff_t * off*** *parameter in read/write to track file state (rather than use a global variable).*

Answer the following questions and turn them in:

The example character device only allows one process to have the device open at once.

1. Was this implemented correctly?
2. Can you implement it better with semaphores?
3. Why is it necessary to have only one instance of the file open at once?
4. Pay particular attention to the while loop in device_read. Why is this necessary?

Check out the semaphores in chapter 5. But beware of deadlocking the kernel!

Save your work!

Put both of today's modules into Subversion.