

Linux driver assignments 4— pseudo char device driver

- you must pass as a module parameter to tell the driver to manage N no. of pseudo devices
- each pseudo device is a kernel buffer + a kfifo object allocated using the following:
 - kcalloc() for the buffer
 - kfifo() for the kfifo object
 - the combination works as the pseudo device
- you must allocate one private device object per pseudo device and the total no. of private device objects is based on the users requirement passed via the module parameter
- your device object must be maintaining all the resources and status information regarding each pseudo device – example, kfifo, cdev and spinlock must be maintained in each private device object
- as mentioned in the class, you must follow the blocking/non-blocking rules mentioned in the chapter 6 of LDD – page 151-152 -also look into pseudo_class1.c for a sample implementation !!!
- add an ioctl() method in your driver's file operations that will provide ioctl commands for resetting a pseudo device and querying a pseudo device for available no of data bytes

Linux driver assignments 4 – pseudo char device driver - continued

- write an application that will use 2 threads per pseudo device – one thread writes and one thread reads – for instance, if you are testing 5 devices, you must use 10 threads, in your application
- only one read and one write thread is allowed at a time in your driver on a per device basis
- your application must open all pseudo devices before creating their read/write threads – in addition, each thread must be passed appropriate parameters to tell which device must be handled in the respective thread
- if your driver supports multiple devices, you must test all the devices simultaneously in your application using multithreading, as described above