Linux driver assignment 5 – pseudo char device driver

- you must pass as a module parameter to tell the driver to manage N no. of pseudo devices(meaning, created and managed by the driver)
- each pseudo device is a a kfifo object + associated kernel buffer allocated:
- 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 private device object must be maintaining all the resources and status information regarding each pseudo device – example, kfifo, cdev, spinlock(s) (if needed) and wait queue(s)(if needed) must be maintained in each private device object
- as mentioned in the class notes, 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 !!!
- you must implement open/read/write/release/ioctl methods in your device driver for managing the pseudo devices

Linux driver assignments 5 – pseudo char device driver - continued

- your ioctl() method of your driver's file operations must support ioctl commands for resetting a pseudo device and querying a pseudo device for available no of data bytes currently filled in a pseudo device – you have to implement 2 ioctl commands
- 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 application/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 (re-entrant thread functions/methods must be written)
- if your driver supports multiple devices, you must test all the devices simultaneously in your application using multithreading, as described above-meaning, open all 5 devices or 10 devices at the same time in your application using as many threads as needed to write and read from the pseudo devices!!

Linux driver assignments 5 – pseudo char device driver - continued

Hints for the assignment 5 :

- read the class notes for character device driver/pseudo device driver
- read ch13/ULK/3 for relevant parts on character device drivers and their Architecture
- read relevant sections of chapter LDD/3
- read relevant sections of chapter6/LDD/3
- read relevant sections of chapter5/LDD/3
- read kfifo / queues related section from LKD/3
- refer to relevant kernel source files as mentioned in the class notes
- use pseudo_class.c and pseudo_class1.c provided – pseudo_class1.c is closer to assignment 5, but requires improvements, additions and changes as needed !!! these examples are different from examples used in chapter 3/6 of LDD/3 – these examples are practically better than LDD/3's corresponding examples !!!