EEL 4732/5733 Advanced Systems Programming

Assignment 5

due Saturday, April 6th by midnight.

In this assignment you are going to write a character device driver by extending the simple driver we studied in class (see slides DDIntro.pptx) in the following ways:

1. **Define a device structure** and embed `structure cdev` in that structure:

```
struct asp_mycdev {
   struct cdev dev;
   char *ramdisk;
   struct semaphore sem;
   int devNo;
   // any other field you may want to add
};
```

2. **Support a variable number of devices** that can be set at load time (default will be 3) (see DDScullIntro.pptx). The device nodes will be named /dev/mycdev0, /dev/mycdev1, ..., /dev/mycdevN-1, where N is the number of devices.

3. **Provide an entry function that would be accessed via `lseek()` function.** That entry function should update the file position pointer based on the offset requested. You should set the file position pointer as requested as long as it stays nonnegative, i.e., $0 \leq requestedposition$. In the case of a request that goes beyond end of the buffer, your implementation needs to expand the buffer, fill the new region with zeros, and update the offset as requested.

4. **Provide an entry function that would be accessed via `ioctl()` function.** You should let the user application clear the data stored in the `ramdisk`. You should define symbol `ASP_CLEAR_BUF` for the command to clear the buffer. Your driver function should also reset the file position pointer to 0. Please see DDScullBasic.pptx for implementation details.

5. **Each device can be opened concurrently and therefore can be accessed for read, write, lseek, and ioctl concurrently.** It is your

responsibility to provide appropriate synchronization to prevent race conditions.

6. All the resources (including the ramdisk, the device structures, and device nodes) should be **recycled/freed at the time of unloading your device driver module**.

The assignment is due Saturday, April 6th by midnight. Please submit all your files along with a Makefile and a README file on CANVAS.