Linux driver assignments – 2 - procfs related

1. based on the example of proc file – proc_seq_example1.c and proc_seq_test.c, complete the following:
   - understand the objects/data-structures, methods and interconnections !!!
   - execute the code with dump_stack() and test the code flow, using proc_seq_test.c !!!
   - try different scenarios and corner cases, as discussed in the class lecture !!!!

Note : you must be comfortable with creating and managing a proc file - now, move on to complete the second part of this assignment, which is very different from the example code - you will also be needing such coding techniques used in the second problem, during the driver programming !!!

Linux driver assignments – 2 - procfs related

2. based on the example of proc file – proc_seq_example1.c , implement the following:

   Create a new proc entry under /proc/proc_test/test. When you access this file you must get the pid, tgid, cmd and kernel-stack address of every process in the system.

   Please note that this is very different from the proc_seq_example1.c. Here, you do not create dummy structures, but using existing system Structures and extract information on the fly !!!

Hints:
- read about procfs from ch4 of LDD/3 – follow the new techniques of procfs, not the old techniques of procfs !!!
- read class notes related to procfs
- read about procfs from the comments given in proc_seq_example1.c
- execute and test sample procfs module and user-space code
- read about procfs from <KSRC>/Documentation/filesystems/seq_file.txt
- refer to kernel source headers and source files referred to in proc_seq_example1.c
- first, get a good understanding of proc_seq_example1.c and its working
- read ch3 of LKD/3 for process related data structures – master pd list
- read ch3 of ULK/3 for process related data structures – master pd list