Linux device driver assignments - 4

Kernel memory management interfaces / system APIs related

- create a dynamic module that will allocate 'n' contiguous page frames in low memory zone via buddy system allocator – 'n' must be a module parameter – you must allocate 'n' page frames in the init() method of the module and release appropriately in the exit() method of the module - start with n=1 and keep increasing – verify that the allocation is as per request using appropriate utilities and proc interfaces that you are aware of - also print the logical addresses of the allocated page frames !!! kmap() and kunmap() are needed for getting the logical addresses of the allocated memory blocks
- repeat the above problem by allocating page frames in the highmem zone instead of lowmem zone – repeat the rest of the steps as in the previous case – in this case, you must use kmap() and kunmap() for dynamic mapping of highmem zone page frames, for each page frame, unlike previous case

Linux Device Driver Assignment - 4

- use a kernel module to allocate several objects using kmalloc(256,GFP_KERNEL) and maintain the objects in a list – use /proc/slabinfo interface to verify the changes in the slab allocator corresponding to size-256 slab cache – meaning, name of the slab cache, no. of slabs, page frames per slab, objects per slab, no of objects and many more – refer to man 5 slabinfo or man 5 proc for more details.
- do not forget to free the allocated objects, in clean-up module.

- you can try the above assignment by allocating 500 objects !!!
- you can also try for larger allocations, if needed !!!

Linux Device Driver Assignment - 4

- before doing the above assignments, you must read the following:
  - KMA pdf and related text file that is provided
  - refer to memory management chapter 12 of LKD/3
  - refer to memory management chapter 8 of ULK/3
  - understand the buddy allocator, non-contiguous memory allocator and slab allocators using tools discussed in the class room :
    - /proc/buddyinfo
    - /proc/slabinfo
    - /proc/meminfo
    - /proc/vmallocinfo
- in all the above cases, to observe better results, you must enter single user mode by typing - init 1 on command line and loading the module - once you have tested, you can again return to the normal mode by typing Init 5, which will bring you back to your regular session !!!

- since the buddy allocators and slab allocators are used by several subsystems and components, it difficult to see expected results – you will see slight deviations, but closer to what you expect !!! Meaning, do not expect precise numbers, for your testing !!! however, you will see changes !! in addition, these allocators also use other policies, which also lead to certain results that are not easy to interpret !!!