Operating Systems Concepts Training

Exercises

You are required to do one exercise each. An exercise may require you to either:

(i) explain how certain facility or function is implemented in Linux Kernel with associated data structures, flow-charts

OR

(ii) do a small project, mostly by writing a kernel module, to achieve certain modification/s in the behaviour.

In case of some modules you may fail to achieve the requirements. In that case you should, after presenting your effort, clearly explain why you were not able to complete it.

Write any program, shell-script (or Perl script) you need to give the required demonstartions.

-oOo-

1. You baught a desktop m/c. The vendor claimed a particular CPU chip installed, but you want to check that claim. You have a Linux Live CD. How can you do?

2. Dynamically installable kernel modules can be a security issue. One "remedy" could be prohibit Dynamic kernel modules, but many plug-n-play devices install their drivers as dynamic modules on-the-fly. Search internet if: (i) this really is an issue, (ii) there is any viable solution is suggested. Report your findings.

3. Write a C program which copies a given file into a new file with given name. Compile and run it under "strace" report and explain all the lines of the output.

4. Write a C program which will persists for time long enough for you to obtain its memory regions map. (Keep it as simple as possible). Display and explain the map. (Hint: the information is in /proc directory)

5. You baught a desktop m/c. The vendor claimed 3GB RAM installed, but you want to check that claim. You have a Linux Live CD. How can you do?

6. What per process information is available from the /proc directory? What are the ways in which you can use this information?

7. Though most of the "files" in /proc directory are read-only, a few can be written into. Find 3 of them and explain for what purpose they are made writable.

8. Explain use of "lsof" command with demonstartion, especially how to see all the opened files of a particular process.

9. Explain use of "cscope" utility with demontration.

10. Explain use of "ctags" utility with demontration.

11. [U] Explain the display given by "top" utility and demonstrate:
    (i) how will you kill a process
    (ii) how will you display processes owned by a specific user
    (iii) how will you change the order in which processes are displayed
    (iv) how can you decide if their is "thrashing" going on

12. Explain and demonstrate the "readprofile" utility.

13. [U] You try to "umount" a device but get a "umount: device busy" response. How will you find which users are using that device/dir/file?

14. Explain how Swapping is implemented and disk swap areas are managed.

15. Explain with a working example program use of mmap() for file memory mapping.

16. Linux 2.6 kernel has built−in kernel profiling, which can be activated by using "profile=2" (for 4 bytes granuality) as an parameter while booting. Play with this facility and explain various kinds of outputs that can be obtained by its reporting utility. Also, try to experimentally estimate the profiling overhead.

17. Explain the exact chain of events (or function invocation) within the kernel when the following C−lib function is executed: open()

18. Explain the exact chain of events (or function invocation) within the kernel when the following C−lib function is executed: read()

19. Explain the exact chain of events (or function invocation) within the kernel when the following C−lib function is executed: write()

20. Explain the exact chain of events (or function invocation) within the kernel when the following C−lib function is executed: mkdir()

21. Linux uses FHS as a standard for its file system. What is the structure of the root file system (/) in Android?

22. A text file is given. We want to find the frequency of occurrence of each unique word in it and display the list in decreasing order of occurrence. Write a pipe using standard utilities to do this. (A word consists of a sequence of non−white−space characters).