Operating System Lab

Experiment 1

Aim: Understanding the Linux Operating System

Theory:

1. Linux is a multi-user and multi tasking network aware OS.
2. It has the operating system core called the Linux Kernel which is commonly named as vmlinuz.
3. The GNU OS Ring surrounds the kernel which provides tools and interfaces with the kernel.
4. The Shell is a component from the GNU OS Ring.
5. Application Layer resides over the GNU OS Ring and interfaces with it.
6. This is why a complete Linux system is often called GNU-Linux.
7. The Shell provides a command line interface to the Operating System's functionality/services.
8. The most common and default shell in Linux is BASH (Bourne Again Shell)
9. Other shells available on linux are the Korn Shell, T Shell, SASH (Stand Alone Shell) and the most feature-full shell named Z shell.
10. Linux supports a large number of file system formats for its file storage. They are namely: ext2fs, ext3fs, jfs, reiserfs, reiser4, vfat, ntfs etc.
11. Basically the linux filesystem is a hierarchical filesystem.
12. Standard filesystem structure is :-

| | |
|---|---|
| / | (root) |
| home | (user's folders) |
| mcp | |
| user1 | |
| user2 | |
| opt | (optional software packages) |
| media | (external media devices like pen-drives) |
| bin | (executables) |
| mnt | (internal media devices like non-root hard disk partitions) |
| boot | (folder with booting-time information) |
| dev | (folder with device descriptions) |
| etc | (folder with configuration files) |
| tmp | (temporary files) |
| lib | (shared software libraries) |
| usr | (installed software packages) |
| root | (super user's files) |

13. Path referencing
    (a) Absolute path (paths like /home/mcp/Documents/abc.c)
    (b) Relative path (paths like ./Documents/abc.c or ../mcp/Documents/abc.c)
    Note: Linux file names are case sensitive.
14. File Permissions : Every file has an associated permission like read write and execute associated with it which is separate for
    (a) “the owner of file”,
    (b) “the group the owner belongs to” and
    (c) “all others”.

A sample output for a directory abc and file a.out of ls-l shows
drwxr-xr-x 3 mcp users 4096 2007-12-22 11:56 abc
-rwxr-xr-x 1 mcp users 19947 2008-09-12 12:28 a.out
(d) here the first letter is for – for ordinary file and d for directory
(e) next 3 letters are permission for owner (read write and execute)
(f) second 3 letters are for the group (read write and execute)
(g) third 3 letters are for all others (read write and execute)
(h) presence of – shows that the permission is not present.
(i) next number is the number of hard links to the file
(j) next is the name of the owner
(k) next is the name of the group the owner belongs to
(l) next number is the size of the file in bytes.
(m) the date is the date and time of last change
(n) the final detail is the name of the file.

15. File Management commands
    (a) ls (List the directory contents)
        i. -l (Long form)
        ii. -m (Comma separated listing)
        iii. -X (Sort by Extension)
        iv. -r (Sort reverse)
        v. -R (List recursively)
    (b) cd (Change directory)
    (c) mkdir (Create a directory)
    (d) rm (Remove file)
    (e) rmdir (Remove directory)
    (f) cp (copy a file/s)
    (g) mv (Rename a file or Move a directory)
    (h) find (find a file like find /home/mcp -name abc.c)
    (i) chmod (change permissions for the file)
        i. chmod o+x ab.out (make ab.out executable for others)
        ii. chmod g+x ab.out (make ab.out executable for group)
        iii. chmod 751 ab.out (make ab.out rwx for owner, r-x for group and --x for others)
        iv. chmod 732 ab.out (make ab.out rwx-wx-w-)
        v. chmod 432 ab.out (make ab.out r---wx-w-)
16. File data commands
    (a) cat (echo the contents of the file)
    (b) sort (echo the contents of the file in sorted form)
    (c) head (show first 10 lines of a file)
    (d) tail (show last 10 lines of a file)
    (e) less (scroll through the file by page up, page down, space, home, end and q)
    (f) more (scroll through the file line by line by space)
17. Redirections
    (a) sending the contents to a file through > (ls -l >files.txt)
    (b) appending the contents to a file through >> (ls -l >> files.txt)
18. Pipe
    (a) sending the output of one command to another using | (cat files.txt|grep ab.c)
    (b) another example ls -l | less