Unix Lab
## Experiment no. 4: To study Unix File System, environmental variable and File Permissions.

Content:

Unix file system is a logical method of organizing and storing large amounts of information in a way that makes it easy to manage. A file is a smallest unit in which the information is stored. All data in Unix is organized into files. All files are organized into directories. These directories are organized into a tree-like structure called the file system.

Files in Unix System are organized into multi-level hierarchy structure known as a directory tree. At the very top of the file system is a directory called "root" which is represented by a "/". All other files are "descendants" of root.
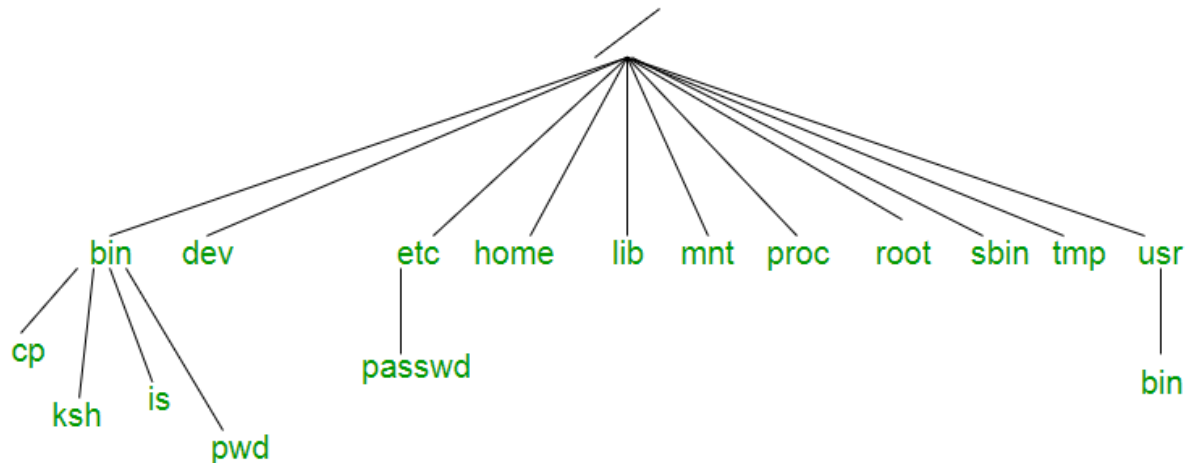
Directories or files	Description
/	Root
/bin	Binary files of fundamental utilities
/boot	Files for successful boot process
/dev	Devices as files
/etc	Configuration and system database
/home	Directory for user
/lib	System libraries
/media	Removable devices as files
/proc	Information of processes as files
/root	Home directory of super user root (administrator)
/tmp	Place for temporary files
/var	A directory whose contains changes often
/var/log	Log files
/var/mail	Mail files
/var/spool	Spool directory for print jobs

Command	Description
touch	Create empty file(s) *touch [options] filename*
cat	Create a file with entered content *cat [options] filename*
cp	Copy file from source to destination *cp [OPTION]source destination*
mv	Moves file from source to destination mv *[OPTION]source destination*
rm	Removes files *rm filename*
mkdir	Makes a directory *mkdir dirName*
rmdir	Removes a directory *rmdir dirName*

Experiment no. 4: To study Unix File System, environmental variable and File Permissions.

cd	Changes a directory *cd path*
pwd	Prints current working directory *pwd*
less	Show content of file *less fileNamePath*
wc	Counts words, lines, characters or multibyte characters *wc [-l –w –c –m] file*
find	Finds a file in directory

Complete run of above commands:
[root@localhost ~]# ls
dos   hello.c hello.js
[root@localhost ~]# touch newFile
[root@localhost ~]# cat >> newFile << EOF
> this is test file
> this contains multi line text
> line 3
> line 4
> line 5
> line 6
> line 7
> line 8
> line 9
> line 10
> and this is last line of file
> EOF
[root@localhost ~]# cat newFile
this is test file
this contains multi line text
line 3
line 4
line 5
line 6
line 7
line 8
line 9
line 10
and this is last line of file
[root@localhost ~]# ls
dos   hello.c hello.js newFile   newFile2
[root@localhost ~]# mkdir temp
[root@localhost ~]# mv newFile
newFile   newFile2
[root@localhost ~]# ls
dos   hello.c hello.js newFile   newFile2 temp
[root@localhost ~]# mv newFile2 temp
[root@localhost ~]# ls
dos   hello.c hello.js newFile   temp
[root@localhost ~]# cd temp
[root@localhost temp]# ls

```
newFile2
[root@localhost temp]# rm newFile2
[root@localhost temp]# cd ..
[root@localhost ~]# ls
dos     hello.c  hello.js  newFile  temp
[root@localhost ~]# rmdir temp
[root@localhost ~]# pwd
/root
[root@localhost ~]# cat hello.c
/* This C source can be compiled with:

   tcc -o hello hello.c

   or if you have more time:

   gcc -o hello hello.c
*/
#include <stdlib.h>
#include <stdio.h>

int main(int argc, char **argv)
{
   printf("Hello World\n");
   return 0;
}
[root@localhost ~]# wc hello.c
    16     38    242 hello.c
[root@localhost ~]# wc -w hello.c
38 hello.c
[root@localhost ~]# wc -l hello.c
16 hello.c
[root@localhost ~]# wc -c hello.c
242 hello.c
[root@localhost ~]# find he*.*
hello.c
hello.js
```