USP Assignment

Q1: Describe the methods to change the file permissions with examples.

Ans: Every file in unix has the following attributes :-

Owner permissions :- The owner's permissions determine what actions the owner of the file can perform on the file.

Group permissions :- The group's permissions determine what actions a user, who is a member of the group that a file belongs to, can perform on the file

Other (world) permissions :- The permissions for other indicate what action all other users can perform on the file.

Permission Indicator

Using ls -p command, it displays various information related to the file permission as follows - $ ls -l /home/anurod
-rwxr-xr-- 1 anurod users 1024 NOV 2 12:50 myfile
drwxr-xr-- 1 anurod users 1024 NOV 2 12:50 mydir

r (Read) : grants the capability to read i.e view the contents of the file.
w (write) : grants the capability to modify or remove the contents of the file.
x (execute) : User with execute permissions can run a file or a program.

classmate
Date
Page

chmod (change mode)

The owner of a file can change the permissions for user (u), group (g), or others (o) by adding (+) or subtracting (-) the read, write and execute permissions.

There are two basic ways of using chmod to change file permissions. The symbolic methods and the absolute form.

example : chmod a+x myfile

Changing owners and groups

Chown : The chown command stands for "change owner" and is used to change the owner of a file.

chgrp : The chgrp command stands for "change group" and is used to change the group of a file.

ex : $chown user filelist

ex : $chgrp group filelist

Q2- Bring out the differences between hard links and soft links with example.

Ans - Hard link : A hardlink acts as a copy (mirrored) of the selected file. It accesses the data available in the original file.

Soft link : A softlink acts as a pointer or a reference to the file name.

Hard link	Soft link
1) Files that are hardlinked take the same inode number.	Files that are soft linked take the different inode number
2) It cannot be used across file systems.	It can be used across file systems.

3) Hard links are not allowed for directories.	Soft links can be used for linking directories.
4) Data present in the original file will still be available in hard links.	Soft links only point to the file name, it does not retain data of the file.
5) Hard links are comparatively faster.	Soft links are comparatively slower.
ex: $mkdir Test $cd Test $touch sample1 $ln sample1 sample2 $ls -il sample1 sample2	ex: $ln -s sample2 sample3 $ls -il sample2 sample3

Q3) Use find command to locate from your home directory

a i) All files having inode number 9076

b ii) All directories having permissions 666

c iii) All files not accessed for more than a year

d iv) All but C program files

Ans 3) i) $ find -inum <inode number>
$ find -inum <9076>

ii) $find -type d -perm 666

iii) $find -atime +365 -type f

iv) $find ~~-iname "*.c"~~ -not -iname "*.c"

Ans 4 i) Command substitution is the mechanism by which the shell performs a given set of command then substitutes their output in the place of \$ command.

ex: DATE = 'date'
echo "Date is \$Date"

ii) set and Shift: Shift is a built in command in Bash which after getting executed shifts/moves command line argument to one position left. ex: \$shift [num]
\$shift 4.

iii) Trap: The trap command allows you to catch signals and execute code when they occur.
If no arguments are specified, trap prints the list of commands associated with each signals.

ex:
trap -l

iv) here: The most syntax for here documents, originating in Unix Shells is << followed by a delimiting identifier (EOF or END), followed by, starting on the next line by the text to be quoted, and then closed by the same delimiting identifier on its own line.

ex: Command <<HERE
text 1
text 2
text N
\$varName
HERE

Q5. Write a shell script that accepts filenames as arguments, for every filename. It should first check whether it exits in current directory and then convert its name to uppercase, but only if a file with new name does not exist.

Ans :-
```
#!/bin/sh
for x in $*
do
if [ -f $x ]
then echo "file $x exists : "
else
echo "$x doesn't exist "
fi
tr '[a-z]' '[A-Z]' < $x
```

Q6: A file's current permission are rw-r-xr--. Specify the chmod expression required to change them for the following:

i) rwxrwxrwx ii) r--r----- iii) ---------- using both the relative and absolute methods of assigning permissions.

i) chmod a+rwx fName] relative
chmod a=rwx fName] Absolute
chmod 777 fName]

ii) chmod a-rwx FName] relative
chmod 000 fName] Absolute

iii) chmod a-rwx filename] relative
chmod 000 filename] Absolute

Q= Use find to locate from your home directory all
i) files with extension .html or .HTML
ii) files having inode no. 9076
iii) directories having permission 666
iv) files modified yesterday

Ans i) $ find $HOME -name "*.[Hh][Tt][Mm][Ll]" -print

ii) $ find -inum <9076>

iii) $find $HOME -perm 666 -type d -print

iv) $find $HOME -mtime $-1 -print

Q= Use find to :-

i) move all files modified within last 24hrs to posix directory under parent directory

ii) locate all files named a.out or core in your home directory tree and remove them interactively

iii) locate file login.sql in the /oracale directory tree, and then copy it to your own directory.

iv) change all directory permissions to 755 and all file permissions to 644 in your home directory tree.

Ans i) $find .type f -mtime -1 -exec mv {} $HOME/posix \;

ii) $find ./\(-name a.out -name core \) -type f -exec rm {} \; -ok

iii) ~~locate f~~ $find /home/oracle -name "login.sql" -exec cp /home/oracle {} \;

iv) $find $HOME -type d -perm 777 -exec chmod 755 {} \;

$find $HOME -type f -perm 777 -exec chmod 644 {} \;