USP Assignment

Q.1. Describe the method to change the file permission with examples

→ Every file in unix has the following attributes :-

owner permission :- The owner's permission determine ~~it~~ what actions the owner of the file can perform on the file.

Group permission :- The groups permissions determine what actions ~~a~~, 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 -l command, it displays various information related to the permission as follows - ls -l /home/amrood -rwxr...1 amrood users 1024 Nov 2 12:50 my file

drwxr-xr--1 amrood users 1024 Nov 12:50 my file

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 (excute) : user with excute permission can run a file as a programs.

Chmod (change mode)

The ~~exec~~ 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 chood to change file permission. The Symbolic method and the absolute form.

example: chood a+4 my file.

changing owners and groups

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

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

ex: $chown : user filelist.

ex: $ chgrp group filelist.

Q:2 Bring out the differences between hard user and soft Colink with example.

→ Hard link : A hardlink acts as a copy (mirrored) of the selected file. It access the data available the original file.

Soft link :- A soft link acts an a pointer or a reference to the file name.

Hard link	Soft link
1. Files that are hard linked 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 ~~dist~~ directories.	soft links can be used for liking 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 compariteruly faster.	Soft links are comparitively owner.

ex: $ mkdir Test
$ cd test
$ torch sample1
$ ln sample 1 sample 2
$ ls - il sample sanple 2
	ex: $ en-s sample 2 sample 3
	$ le-il sample 2 Sample 3

Q.3 Use find Command to locate from your home director

ai.) All file haming inode member 9076

bi) All directories haming permissions 666

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

d(ii) All but C program file.

Ans 3a i:) $ find - imum <inode number>

$ find - imun <9076>

bii) $ find - type d - perm 666

ciii) $ find - atime + 365 - type of

d iv:) $ find - not - iname "*c"

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

ex: Date = 'date'
echo "Data is $ Date"

ii.) Set & shift: Shift is a built in Command in Bash which after getting executed shifts / never command line argument to one position left. ex: $ shift ([illegible])
$ shift 4

iii.) Trap: The trap Command allows you to catch signals and execute code where they occur.
If no argument are specified, trap prints the list of Commands associated with each signals.

ex:-

trap -l

iv.) here: The most symtox for here documents originality in chix Shells is << followed by a destributing identifier (E of or Fxxg), followed
text to be quoted, and the starting on $ the next by the closed by the same ~~dot~~
identifier on its own line.

ex:- text 1
text 2
text N
$ usarname
HERE

Q:5 Write a Shell Script that accepts file name an arguments, for every fluence. It should first check where it exits in current directory and than Convert its name to uppercase, but only if a file with new name does not exist.

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

Q.6 A file's current permission are rw_ r_ xr_ specify the chmod expression required to change them for the following.

i.) rwx rwx rwx .... ii.) r_ _ r_ _ _ _ iii) _ _ _ _ _ _ using both the relative and absolute methods of assiging permissions.

i.) chmod a + rwx fname] relative
Chmod a = rwx fname
chmod 777 fname] relative

ii.) chmod a - rwx fname] relative
Chmod 000 fname] relative

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

Q = use find to locate from your home directory all

i.) files with extention html or HTML.
ii.) files having inode no. 9076
iii.) directories having permission 666
iv.) files modified yesterday.

→ i.) $ find $ HOME- name "*. [Hh] [TE] [Mm] [Ll]"-print
ii.) $ find -inum <9076>
iii.) $ find $HOME -per 666-type d-print
iv.) $ find $HOME -mtime -1-print

Q = use find to :-

i.) ~~Meame~~ all files modified within last 24 hrs to ponyn ~~dist~~ directory under directory.

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

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

iv.) Change all directory permission ~~to~~ 755 and all file permission ~~to~~ 644 in your home directory tree.

→ i.) $ find . type of -htime -1 - exec MV {} $ HOME/ pocix\;

ii.) $ find v\| (-name -a'out -name core \) - type of -exec m {};-oh

iii) $ find /home /oxacle - name "login.sqe" - execup\| home/own{};

iv.) $ find $HOME - typed -perm 777 -exec chmod 755 {} \;

$ find $HOME - type if - perm 777 - exec chmod C44 {}\;