20/12/21

USP Assignment Set 1

1) Describe the methods to change the file permissions with examples.

A) ~~Fa~~

⟹ Absolute method : specifies a set of three numbers that together determine all the access classes and types. We should mention the state of all permissions, cannot ~~specify~~. change only particular attributes.

Read 4, write 2, Execute 1

Three numbers in order : user, group and others.

for example → user has all permissions = 4+2+1 = 7
→ group has read and execute = 4+0+1 = 5
→ others have executing permissions = 0+0+1 = 1

```
chmod 751 filename.
```

→ Symbolic method :
The owner can change permissions using u(user), g(group), o(others), a(all) and operators such as +(add access)
−(remove access)
=(overwrite)

read(r), write(w), execute(e).

For example → remove read and write permissions of group and others

```
chmod go -rw filename
```

→ set permissions exact to read for others.

```
chmod o=r filename.
```

→ add write to others.

```
chmod o-w filename.
```

2) Bring out the difference between hard links and soft links with examples.

A) Hardlink ~~is the copy~~ work with files; The inode for all hard links are the same. ~~The~~ It accesses the data available in original file.
If original file is deleted, the hard link with still contain the data of that file.

Hardlinks.	Softlinks
- Take same inode number	- take different inone number.
- Not allowed for directories	- Can be used for directories.
- Cannot used across file systems	- Can be used across file systems
- Data in original will be in hard links.	- softlinks only point to the filename, it doesn't retain data.
- If original file removed, link will still work and have contents of original.	- doesn't work if original file is removed.
- Comparatively faster.	- slower.

3) Find command to locate from your home directory.

a) All files having inode number 9076.

```
find . -inum 9076
```

b) All directories having permissions 666

```
find . -type d -perm 666.
```

c) All files not accessed for more than a year

```
find . -atime +365
```

DATE: PAGE:

d) All but C program files.

$ find -c - name "*.c"

4) Explain i) Command substitution
ii) Set and shift
iii) trap
iv) here.

A) i) It is when a shell performs a given set of commands and then substitutes their output in the place of the commands. Uses backquotes.

Eg: DATE = `date`
echo "Date is $DATE"

ii) Set and shift
Set is used to set values of command line arguments that will be used by program.

Shift used to move the command line arguments to one position left. One integer argument.

iii) Trap allows to catch signals and execute specified code when they occur. Works as a signal handler.

iv) Here

5) Write shell script - accepts filenames as arguements, for every filename. It should first check whether it exists in current directory and then convert its name to uppercase, but only if a file with new name doesn't exist.

```
#!/bin/sh

~~echo "enter number of filenames"~~
~~read n~~
i=1
~~while [ $i -le $n ]~~
~~for i in~~
~~while [ $n -gt 0 ]~~
~~do~~
    ~~if [ -e $i ]~~

for i in "$@"
do
    if [ -e $i ]
    then
        j=$(echo $i | tr 'a-z' 'A-Z')
        if [ !-e $j ]
        then
            `mv $i $j`
        ~~done~~ fi
    ~~done~~ fi.
done.
```

P.T.O

6) file's current permissions are rw_r_xr__
write chmod to change to

i) ~~rwx~~ rwxrwxrwx

relative : chmod ~~o+x~~ a=rwx file
~~chmod~~

absolute : chmod 777 file.

ii) r__r_____

relative : chmod u-w file
chmod g-x file
chmod o-r file.

absolute : chmod 440 file.

iii) _________

relative : chmod u-rw file.
g-rx file.
o-r file.
} a-rwx file.

absolute : chmod 000 file.