8)Write an Interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.

AIM: To Write an Interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.

PROGRAM:

```
echo 1.copy
echo 2.rename
echo 3.remove
echo 4.link
echo 5.exit
echo "Enter Your Choice"
read ch
case $ch in
1)echo "Enter Source File"
read s
echo "Enter the Destination File"
read d
cp $s $d
;;
2)echo "Enter Old File Name"
read of
echo "Enter Neew File Name"
read nf
mv $of $nf
;;
3)echo "Enter the Filename to delete"
read df
rm $df
;;
4)echo "Enter File1"
read f1
echo "Enter File2"
read f2
ln $f1 $f2
;;
5)exit 0
;;
Esac
```

OUTPUT:

```
[20A91A0566@Linux ~]$ vi switch.sh
[20A91A0566@Linux ~]$ sh switch.sh
1.copy
2.rename
3.remove
4.link
5.exit
Enter Your Choice
1
Enter Source File
oldfile
Enter the Destination File
newfile
[20A91A0566@Linux ~]$ cat newfile
Hai
Hello
[20A91A0566@Linux ~]$ sh switch.sh
1.copy
2.rename
3.remove
4.link
5.exit
Enter Your Choice
3
Enter the Filename to delete
oldfile
[20A91A0566@Linux ~]$ cat oldfile
cat: oldfile : No such File or directory
```

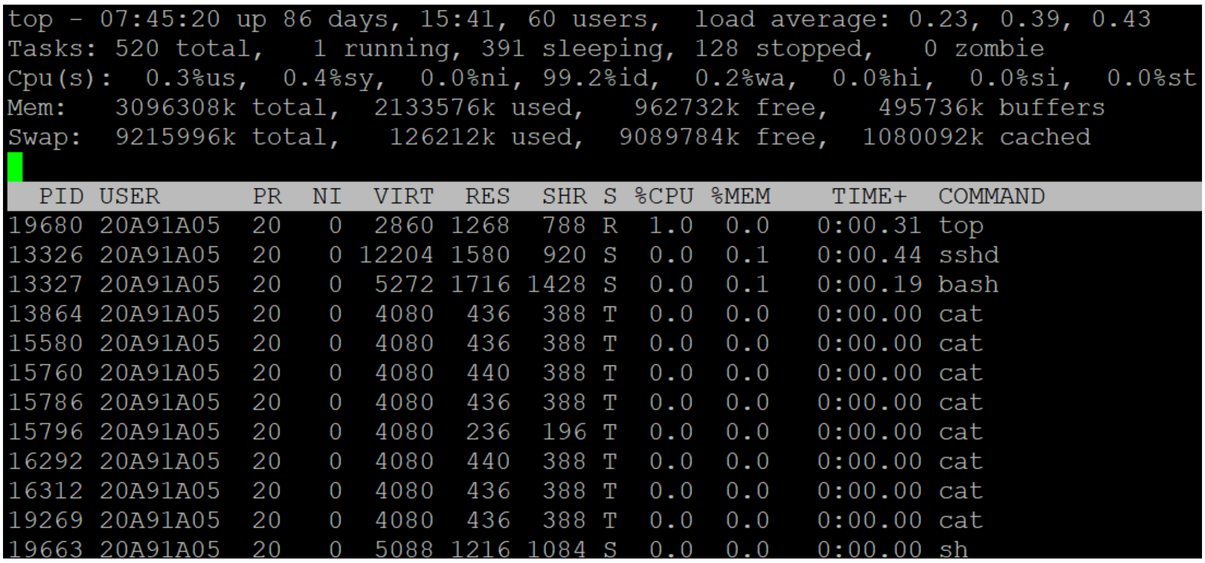
9a)Write a Shell Script that takes a login name as command line Argument and reports when that person logs in

AIM: A Shell Script that takes a login name as command line Argument and reports when that person logs in

PROGRAM:

```
echo "Who are You?"
read user
echo $user
name=$(whoami)
if [ $user==$name ]
then
top -u $user
else
echo "Not logged in"
fi
```

OUTPUT:



```
top - 07:45:20 up 86 days, 15:41, 60 users,  load average: 0.23, 0.39, 0.43
Tasks: 520 total,   1 running, 391 sleeping, 128 stopped,   0 zombie
Cpu(s):  0.3%us,  0.4%sy,  0.0%ni, 99.2%id,  0.2%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   3096308k total,  2133576k used,   962732k free,   495736k buffers
Swap:  9215996k total,   126212k used,  9089784k free,  1080092k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
19680 20A91A05  20   0  2860 1268  788 R  1.0  0.0   0:00.31 top
13326 20A91A05  20   0 12204 1580  920 S  0.0  0.1   0:00.44 sshd
13327 20A91A05  20   0  5272 1716 1428 S  0.0  0.1   0:00.19 bash
13864 20A91A05  20   0  4080  436  388 T  0.0  0.0   0:00.00 cat
15580 20A91A05  20   0  4080  436  388 T  0.0  0.0   0:00.00 cat
15760 20A91A05  20   0  4080  440  388 T  0.0  0.0   0:00.00 cat
15786 20A91A05  20   0  4080  436  388 T  0.0  0.0   0:00.00 cat
15796 20A91A05  20   0  4080  236  196 T  0.0  0.0   0:00.00 cat
16292 20A91A05  20   0  4080  440  388 T  0.0  0.0   0:00.00 cat
16312 20A91A05  20   0  4080  436  388 T  0.0  0.0   0:00.00 cat
19269 20A91A05  20   0  4080  436  388 T  0.0  0.0   0:00.00 cat
19663 20A91A05  20   0  5088 1216 1084 S  0.0  0.0   0:00.00 sh
```

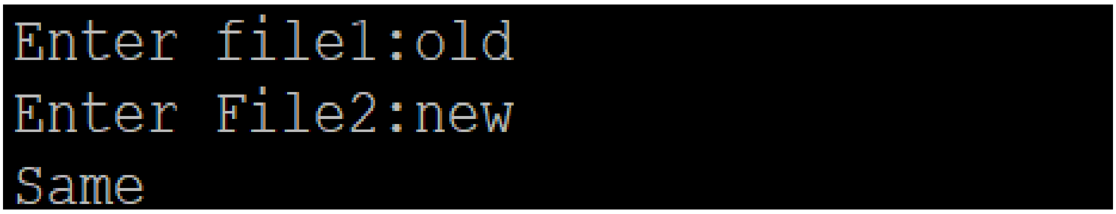
9b) Write a shell script which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted

AIM: A shell script which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.

PROGRAM:

```
echo -n "Enter file1:"

read file1

echo -n "Enter File2:"

read file2

`cmp $file1 $file2>equal`

if [ ! -s equal ]

then

echo Same

rm $file2

else

echo Different

fi
```

OUTPUT:



```
Enter file1:old
Enter File2:new
Same
```

10) Write a C program that takes one or more file or directory names as a command line input and reports the following information on the file:

i) File type.

ii) Number of links.

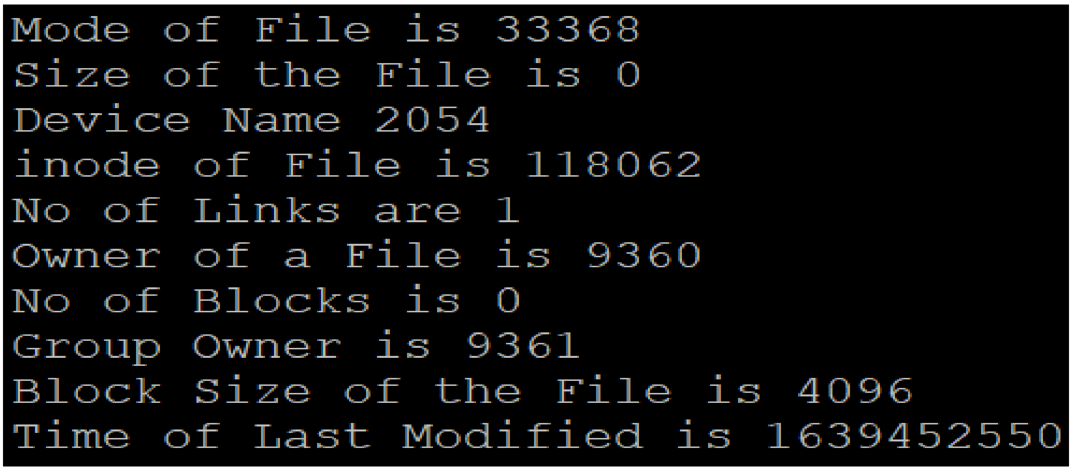
iii) Read, write and execute permissions.

iv) Time of last access (Note : Use stat/fstat system calls).

PROGRAM:

```
#include<stdio.h>

#include<unistd.h>

#include<sys/stat.h>

#include<sys/types.h>

#include<fcntl.h>

void main()

{

int fd;

struct stat buf;

fd=open("hello.txt",O_RDONLY|O_CREAT,600);

if(fd!=-1)

{

if(fstat(fd,&buf)==0)

{

printf("Mode of File is %u",buf.st_mode);

printf("\nSize of the File is %u",buf.st_size);

printf("\nDevice Name %u",buf.st_dev);

printf("\ninode of File is %u",buf.st_ino);

printf("\nNo of Links are %u",buf.st_nlink);

printf("\nOwner of a File is %u",buf.st_uid);

printf("\nNo of Blocks is %u",buf.st_blocks);

printf("\nGroup Owner is %u",buf.st_gid);

printf("\nBlock Size of the File is %u",buf.st_blksize);
printf("\nTime of Last Modified is %u",buf.st_ctime);

}

else

printf("Error in fstat() syscall");

}

else

printf("Error in open() syscall");

}
```

OUTPUT:



```
Mode of File is 33368
Size of the File is 0
Device Name 2054
inode of File is 118062
No of Links are 1
Owner of a File is 9360
No of Blocks is 0
Group Owner is 9361
Block Size of the File is 4096
Time of Last Modified is 1639452550
```