

9a) AIM:Write 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 signed in"
```

```
fi
```

output:

```
[20A91A05B6@Linux ~]$ sh 9a.sh
who are you?
20A91A05B6
20A91A05B6
top - 06:39:39 up 86 days, 14:35, 39 users,  load average: 0.43, 0.48, 0.37
Tasks: 329 total,   1 running, 316 sleeping,   12 stopped,   0 zombie
Cpu(s):  1.7%us,   0.5%sy,   0.0%ni, 97.6%id,   0.3%wa,   0.0%hi,   0.0%si,   0.0%st
Mem:   3096308k total,  2191700k used,   904608k free,   529464k buffers
Swap:   9215996k total,   126212k used,  9089784k free,  1147680k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
14971	20A91A05	20	0	2848	1244	868	R	0.7	0.0	0:00.71	top
13296	20A91A05	20	0	12292	1672	920	S	0.0	0.1	0:00.06	sshd
13297	20A91A05	20	0	5276	1708	1428	S	0.0	0.1	0:00.02	bash
14963	20A91A05	20	0	5088	1216	1088	S	0.0	0.0	0:00.00	sh

9b)AIM: 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.

Program:

```
echo -n"enter file2"
```

```
read file1
```

```
echo -n "enter file2"
```

```
read file2
```

```
diff $file1 $file2
```

```
r=`echo $?`
```

```
if [ $r != 0 ]
```

```
then
```

```
echo "different"
```

```
else
```

```
echo "same"
```

```
rm $file2
```

```
fi
```

output:

```
[20A91A05B6@Linux ~]$ sh 9b.sh
enter file1bts
enter file2bts2
8d7
<
different
[20A91A05B6@Linux ~]$ cp bts bts3
[20A91A05B6@Linux ~]$ cat bts3
sa
jin
suga
jhope
jimin
v
jk
[20A91A05B6@Linux ~]$ sh 9b.sh
enter file1bts
enter file2bts3
same
[20A91A05B6@Linux ~]$
```

10)AIM:

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<fcntl.h>

void main()

{

int fd;

struct stat buf;

fd=open("new1.txt",O_RDONLY|O_CREAT,600);

if(fd!=-1)

{

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

{

printf("mode of files %u",buf.st_mode);

printf("\n size of the file is %u ",buf.st_size);

printf("\n device name %u",buf.st_dev);

printf("\n inode of file is%u",buf.st_ino);

printf("\n no of links are %u ",buf.st_nlink);

printf("\n owner of a file %u",buf.st_uid);

printf("\n no of blocks is %u",buf.st_blocks);
```

```
printf("\n group owner is %u ",buf.st_gid);

printf("\n block size of the file is %u ",buf.st_blksize);

printf("\n tim of last modifiels %u ",buf.st_ctime);

}

else

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

}

else

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

}
```

Output:

```
[20A91A05B6@Linux ~]$ vi filedetails.c
[20A91A05B6@Linux ~]$ cc filedetails.c
[20A91A05B6@Linux ~]$ ./a.out
mode of files 33368
size of the file is 0
device name 2054
inode of file is 127100
no of links are 1
owner of a file 9372
no of blocks is 0
group owner is 9373
block size of the file is 4096
tim of last modifiels 1639452681 [20A91A05B6@Linux ~]$
```