

## EXPERIMENT 4:

A) Write a shell script that determines the period for which a specified user is working on the system.

- echo "Enter the desired username:"
- read a
- last \$a

```
GNU nano 6.2
echo "Enter the desired username:"
read a
last $a
```

```
suraj@surajpandit:~$ nano period.sh
suraj@surajpandit:~$ ./period.sh
Enter the desired username:
suraj
suraj      tty2      tty2      Fri Nov  4 22:27   still logged in
suraj      tty2      tty2      Fri Nov  4 13:07 - crash (09:18)
suraj      tty2      tty2      Wed Nov  2 09:48 - crash (2+03:18)
suraj      tty2      tty2      Tue Nov  1 19:22 - crash (14:25)
suraj      tty2      tty2      Mon Oct 31 22:32 - crash (20:48)
suraj      tty2      tty2      Fri Oct 14 14:29 - crash (17+08:02)
suraj      tty2      tty2      Fri Oct 14 13:24 - crash (01:05)
suraj      tty2      tty2      Wed Oct 12 21:02 - crash (1+16:21)
suraj      tty2      tty2      Tue Oct 11 19:36 - crash (03:32)
suraj      tty2      tty2      Fri Oct  7 13:13 - crash (4+06:22)
suraj      tty2      tty2      Thu Oct  6 21:37 - crash (15:34)
suraj      tty2      tty2      Thu Oct  6 17:58 - crash (03:38)
suraj      tty2      tty2      Wed Oct  5 16:16 - crash (1+01:41)
suraj      tty2      tty2      Tue Oct  4 12:17 - crash (1+00:29)
suraj      tty2      tty2      Sat Sep 24 21:27 - crash (9+14:50)
suraj      tty2      tty2      Sat Sep 17 12:04 - crash (7+09:22)
suraj      tty2      tty2      Fri Sep 16 13:04 - crash (22:57)
suraj      tty2      tty2      Thu Sep 15 22:56 - crash (14:07)
suraj      tty2      tty2      Thu Sep 15 22:05 - crash (00:49)
suraj      tty2      tty2      Thu Sep 15 22:03 - crash (00:01)
suraj      tty2      tty2      Thu Sep 15 19:08 - crash (02:54)
suraj      tty2      tty2      Thu Sep 15 18:44 - down   (00:13)
suraj      tty2      tty2      Thu Sep 15 18:42 - crash (00:02)
suraj      tty2      tty2      Thu Sep 15 18:41 - 18:42 (00:00)
suraj      tty2      tty2      Thu Sep 15 18:40 - crash (00:00)
suraj      tty2      tty2      Thu Sep 15 18:39 - 18:40 (00:01)
```

**B) Write a shell script that displays all the lines between start and end line numbers passed as argument.**

- `echo "Enter the Filename:"`
- `read a`
- `echo "Enter the starting line:"`
- `read sl`
- `echo "Enter the ending line:"`
- `read el`
- `sed -n $sl,$el\p $a`

```
GNU nano 6.2
echo "enter filename"
read a
echo "enter the starting line:"
read sl
echo "enter the ending line:"
read el
sed -n $sl,$el\p $a
```

```
suraj@surajpandit:~$ nano line.sh
suraj@surajpandit:~$ ./line.sh
enter filename
file.txt
enter the starting line:
1
enter the ending line:
3
Tell me and i forget.
teach me and i remember.
involve me and i learn.
suraj@surajpandit:~$ cat file.txt
Tell me and i forget.
teach me and i remember.
involve me and i learn.
by benjamin
suraj@surajpandit:~$
```

**C) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.**

- **if [ \$# -eq 0 ]**
- **then**
- **echo NO ARGUMENTS**
- **else**
- **pattern=\$1**
- **shift**
- **for fname in \$\***
- **do**
- **if [ -f \$fname ]**
- **then**
- **echo DELETING: \$pattern FROM : \$fname**
- **sed '/\$pattern'/d' \$fname**
- **else**
- **echo \$fname : FILENAME NOT FOUND**
- **fi**
- **done**
- **fi**

```
GNU nano 6.2
if [ $# -eq 0 ]
then
echo NO ARGUMENTS
else
pattern=$1
shift
for fname in $*
do
if [ -f $fname ]
then
echo DELETING: $pattern FROM: $fname
sed '/$pattern'/d' $fname

else
echo $fname :FILE NAME NOT FOUND
fi
done
fi
█
```

```
suraj@surajpandit:~$ nano delete.sh
suraj@surajpandit:~$ cat file.txt
Tell me and i forget.
teach me and i remember.
involve me and i learn.
by benjamin
suraj@surajpandit:~$ ./delete.sh
NO ARGUMENTS
suraj@surajpandit:~$ ./delete.sh teach file.txt
DELETING: teach FROM: file.txt
Tell me and i forget.
involve me and i learn.
by benjamin
suraj@surajpandit:~$ █
```