

Experiment -1

1.\$ Cal:- It is used to see calendar of specific month or year.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ cal
    August 2023
Su Mo Tu We Th Fr Sa
                1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
aib@pamolil-virtual-machine:~$
```

\$ cal 2023:- It is used to get the current calendar of the specified year.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ cal 2023
                2023
    January    February    March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7      1  2  3  4      1  2  3  4
 8  9 10 11 12 13 14    5  6  7  8  9 10 11    5  6  7  8  9 10 11
15 16 17 18 19 20 21   12 13 14 15 16 17 18   12 13 14 15 16 17 18
22 23 24 25 26 27 28   19 20 21 22 23 24 25   19 20 21 22 23 24 25
29 30 31              26 27 28              26 27 28 29 30 31

    April      May          June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
                1          1  2  3  4  5  6      1  2  3
 2  3  4  5  6  7  8    7  8  9 10 11 12 13    4  5  6  7  8  9 10
 9 10 11 12 13 14 15   14 15 16 17 18 19 20   11 12 13 14 15 16 17
16 17 18 19 20 21 22   21 22 23 24 25 26 27   18 19 20 21 22 23 24
23 24 25 26 27 28 29   28 29 30 31             25 26 27 28 29 30
30

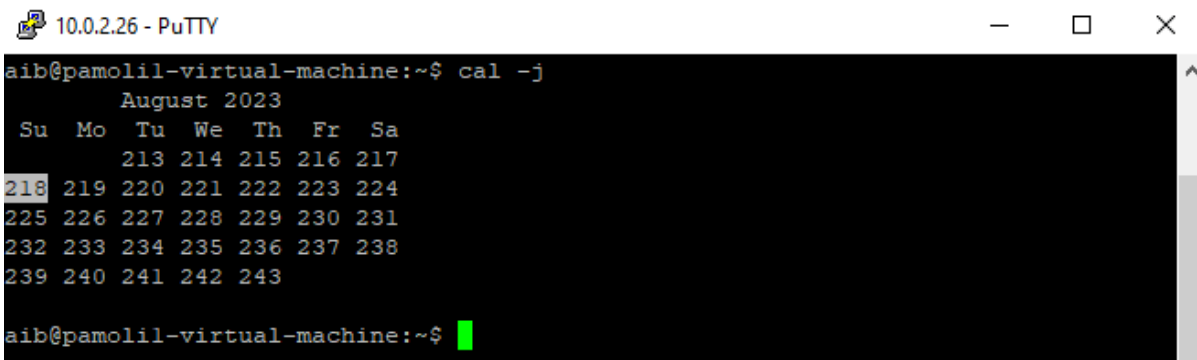
    July        August      September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
                1          1  2  3  4  5      1  2
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    3  4  5  6  7  8  9
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   10 11 12 13 14 15 16
16 17 18 19 20 21 22   20 21 22 23 24 25 26   17 18 19 20 21 22 23
23 24 25 26 27 28 29   27 28 29 30 31         24 25 26 27 28 29 30
30 31

    October     November    December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7    1  2  3  4          1  2
 8  9 10 11 12 13 14    5  6  7  8  9 10 11    3  4  5  6  7  8  9
15 16 17 18 19 20 21   12 13 14 15 16 17 18   10 11 12 13 14 15 16
22 23 24 25 26 27 28   19 20 21 22 23 24 25   17 18 19 20 21 22 23
29 30 31              26 27 28 29 30         24 25 26 27 28 29 30
31
```

\$ cal -3:- It is used to get the calendar of first specified month.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ cal -3
                2023
    July        August      September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
                1          1  2  3  4  5      1  2
 2  3  4  5  6  7  8    6  7  8  9 10 11 12    3  4  5  6  7  8  9
 9 10 11 12 13 14 15   13 14 15 16 17 18 19   10 11 12 13 14 15 16
16 17 18 19 20 21 22   20 21 22 23 24 25 26   17 18 19 20 21 22 23
23 24 25 26 27 28 29   27 28 29 30 31         24 25 26 27 28 29 30
30 31
```

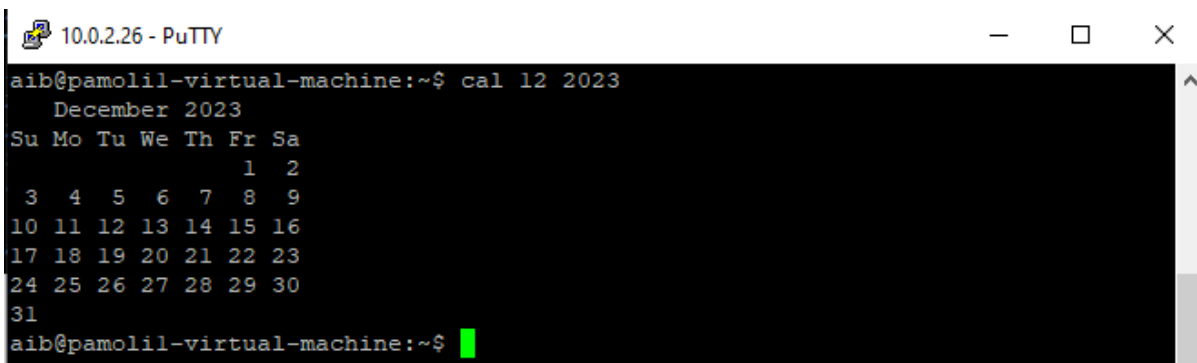
\$ cal -j:- It is used to get the calendar of current month with current number of day in the year.



```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ cal -j
    August 2023
Su  Mo  Tu  We  Th  Fr  Sa
           213 214 215 216 217
218 219 220 221 222 223 224
225 226 227 228 229 230 231
232 233 234 235 236 237 238
239 240 241 242 243

aib@pamolil-virtual-machine:~$
```

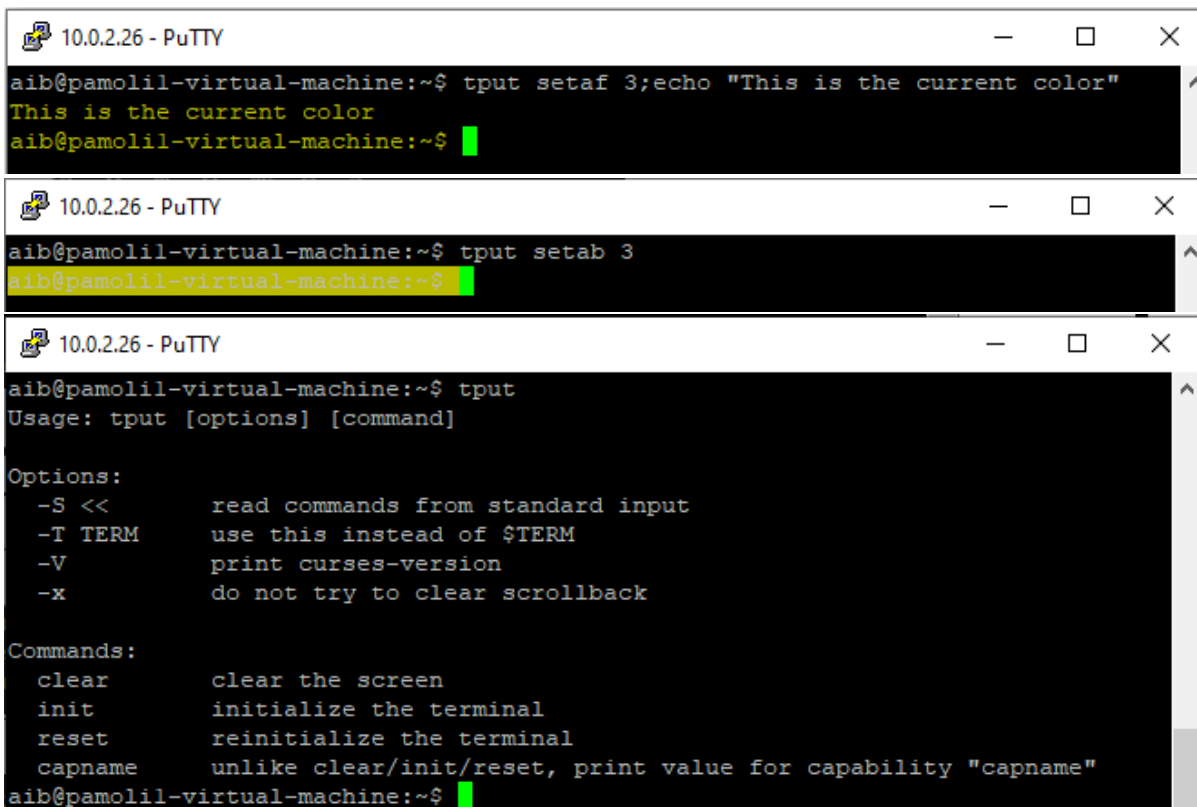
\$ cal [month][year]:- It is used to get the calendar of specified month and year.



```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ cal 12 2023
    December 2023
Su  Mo  Tu  We  Th  Fr  Sa
           1  2
  3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31

aib@pamolil-virtual-machine:~$
```

2. \$ tput:- It is used to access terminal properties and manipulates how properties such as colors, position are displayed in it.



```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ tput setaf 3;echo "This is the current color"
This is the current color
aib@pamolil-virtual-machine:~$

10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ tput setab 3
aib@pamolil-virtual-machine:~$

10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ tput
Usage: tput [options] [command]

Options:
-S <<      read commands from standard input
-T TERM    use this instead of $TERM
-V         print curses-version
-x         do not try to clear scrollbar

Commands:
clear      clear the screen
init       initialize the terminal
reset      reinitialize the terminal
capname    unlike clear/init/reset, print value for capability "capname"

aib@pamolil-virtual-machine:~$
```

3.\$ who:- It is used to get the details of all the user currently logged in the server or computer.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ who
aib pts/0 2023-08-06 15:32 (10.103.32.193)
pamolil :0 2023-07-28 12:32 (:0)
```

4.\$ Ps:- It is used to list the currently running processes and their PIDs along with some other information.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ ps
PID TTY TIME CMD
6750 pts/0 00:00:00 bash
6963 pts/0 00:00:00 ps
```

5.ls:- It is used to get the names of all the files and directories present inside the current working directory.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:/home$ ls
aib aib4 aib9 aiit3 aiit8 alpana keshavk pamolil tanvis
aib1 aib5 aiit aiit4 aiit9 anirudhd khitiza rajitu umesh
aib10 aib6 aiit1 aiit5 aismohali ankurk krishnak rishabhk vijayk
aib2 aib7 aiit10 aiit6 akash arzoon manavp rishir vishalc
aib3 aib8 aiit2 aiit7 akshats ashmitrajp merugun saras
aib@pamolil-virtual-machine:/home$
```

\$ls a*:- It is used to get the names of the files starting with 'a' in the current directory.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ ls a*
abc2.txt abc.sh abc.txt abhi.sh a.sh ashu.txt
aib@pamolil-virtual-machine:~$
```

\$ls -list:- It is used to get the details of the files inside the current directory.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~/Rohit$ ls -list
total 12
788376 4 drwxrwxr-x 2 aib aib 4096 Aug 6 17:19 Anshu
788374 4 drwxrwxr-x 2 aib aib 4096 Aug 6 17:19 Him
788341 4 drwxrwxr-x 2 aib aib 4096 Aug 6 17:19 Himanshu
aib@pamolil-virtual-machine:~/Rohit$
```

\$ls -l ab*:- It is used to get the details of the files starting with 'ab' in the current directory.

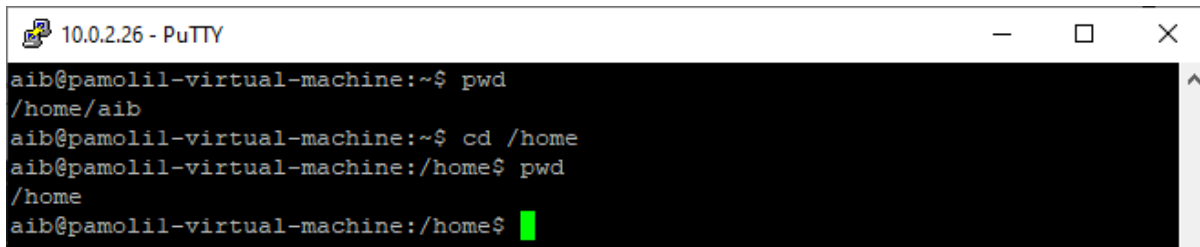
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ ls -l ab*
-rw-rw-r-- 1 aib aib 39 Aug 3 13:17 abc2.txt
-rw-rw-r-- 1 aib aib 86 Aug 3 10:41 abc.sh
-rw-rw-r-- 1 aib aib 42 Aug 3 13:17 abc.txt
-rw-rw-r-- 1 aib aib 184 Aug 3 10:23 abhi.sh
aib@pamolil-virtual-machine:~$
```

6. **\$ pwd:-** It is used to get the current working directory.

A terminal window titled "10.0.2.26 - PuTTY" showing the command 'pwd' being executed. The output is '/home/aib'.

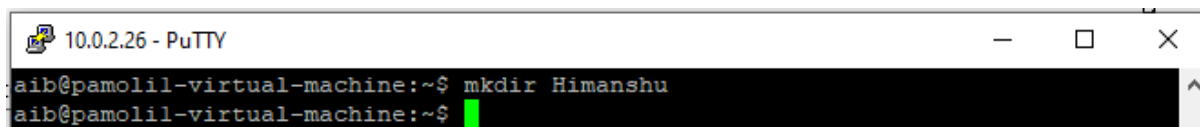
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ pwd
/home/aib
aib@pamolil-virtual-machine:~$
```

7. **\$cd :-** It is used to change the current working directory.

A terminal window titled "10.0.2.26 - PuTTY" showing the command 'cd /home' being executed. The prompt changes from '~\$' to '/home\$'.

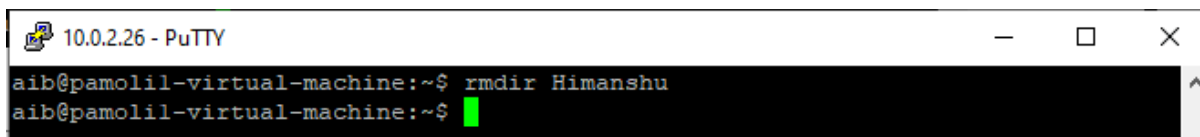
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ pwd
/home/aib
aib@pamolil-virtual-machine:~$ cd /home
aib@pamolil-virtual-machine:/home$ pwd
/home
aib@pamolil-virtual-machine:/home$
```

8. **\$mkdir :-** It is used to make the directory inside the current working directory.

A terminal window titled "10.0.2.26 - PuTTY" showing the command 'mkdir Himanshu' being executed. The prompt returns to '~\$'.

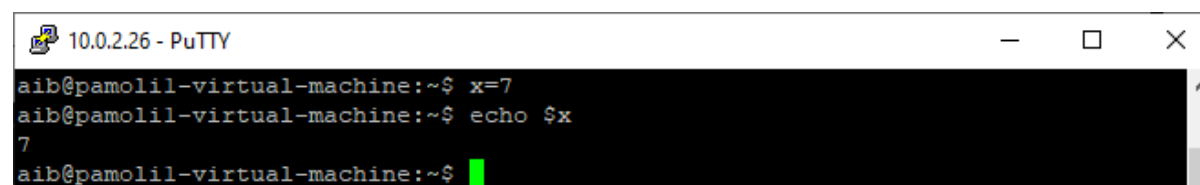
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ mkdir Himanshu
aib@pamolil-virtual-machine:~$
```

9. **\$rmdir:-** It is used to delete an unempty directory inside the current directory.

A terminal window titled "10.0.2.26 - PuTTY" showing the command 'rmdir Himanshu' being executed. The prompt returns to '~\$'.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ rmdir Himanshu
aib@pamolil-virtual-machine:~$
```

10. **\$echo:-** It is used to print the variable or string.

A terminal window titled "10.0.2.26 - PuTTY" showing the command 'echo \$x' being executed after 'x=7' has been assigned. The output is '7'.

```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ x=7
aib@pamolil-virtual-machine:~$ echo $x
7
aib@pamolil-virtual-machine:~$
```

Experiment-2

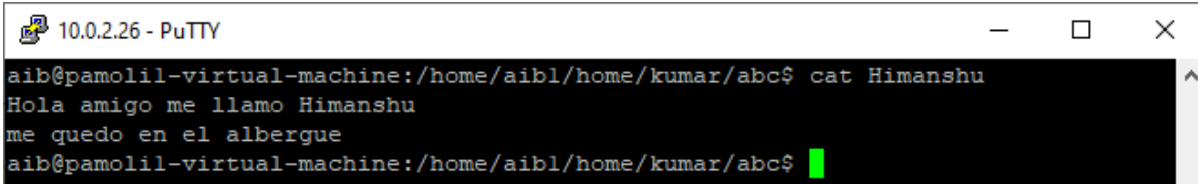
1.\$cat:- It provides us space to write content in our own language. It helps to create ,read and concatenate files.



```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ cat
Holo me llamo Himanshu
Holo me llamo Himanshu
aib@pamolil-virtual-machine:~$
```

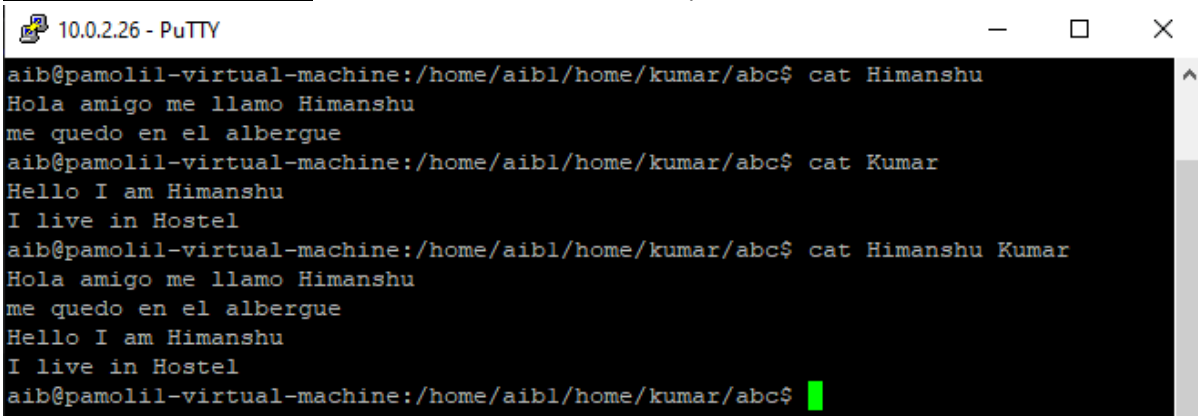
We used ctrl+D to exit from the write mode.

\$cat filename:- It is used to read a single file within a directory.



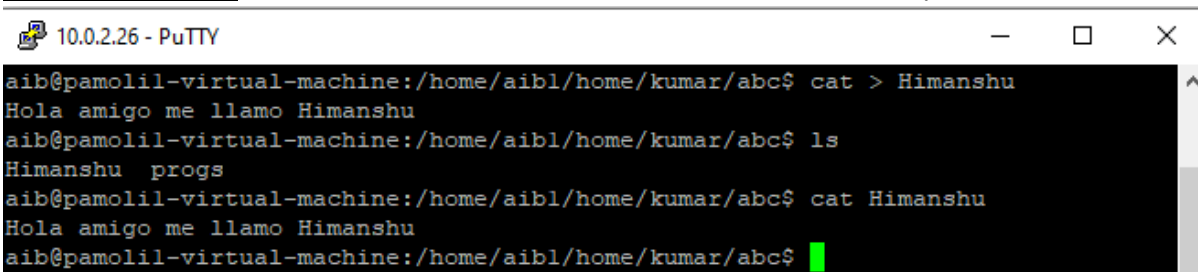
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ cat Himanshu
Hola amigo me llamo Himanshu
me quedo en el albergue
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$
```

\$cat filename1 filename 2:- It is used to read two or multiple file.



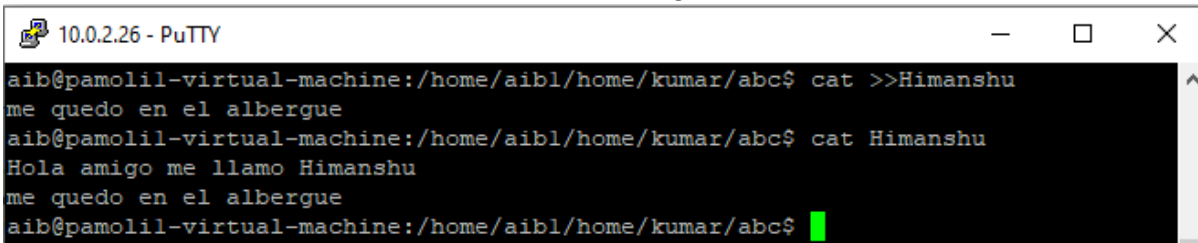
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ cat Himanshu
Hola amigo me llamo Himanshu
me quedo en el albergue
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ cat Kumar
Hello I am Himanshu
I live in Hostel
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ cat Himanshu Kumar
Hola amigo me llamo Himanshu
me quedo en el albergue
Hello I am Himanshu
I live in Hostel
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$
```

\$cat > newfilename:- It is used to create a new file inside the current directory.



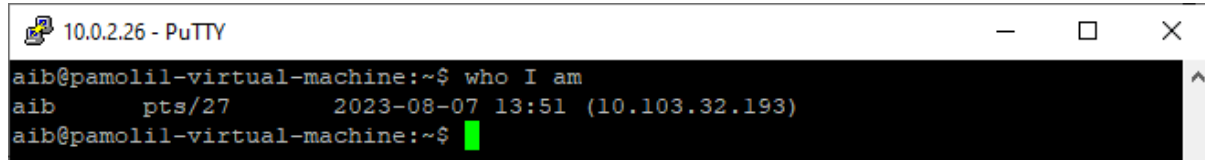
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ cat > Himanshu
Hola amigo me llamo Himanshu
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ ls
Himanshu  progs
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ cat Himanshu
Hola amigo me llamo Himanshu
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$
```

\$cat >> filename:- It is used to add content in the existing file.



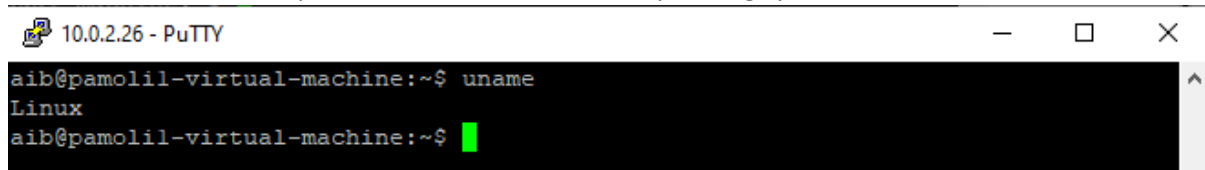
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ cat >>Himanshu
me quedo en el albergue
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ cat Himanshu
Hola amigo me llamo Himanshu
me quedo en el albergue
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$
```

2.\$who I am:- It is used to get my current user details.



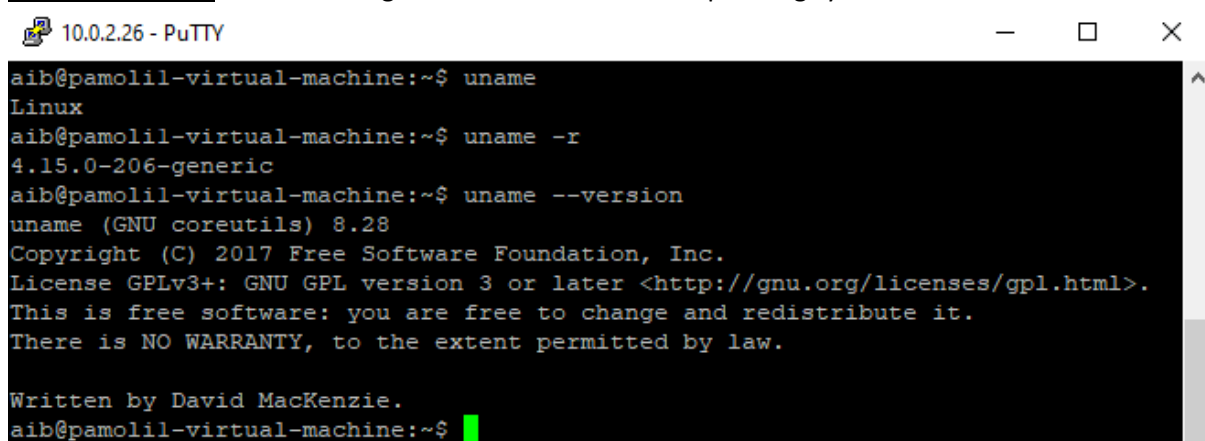
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ who I am
aib      pts/27      2023-08-07 13:51 (10.103.32.193)
aib@pamolil-virtual-machine:~$
```

3.\$uname:- It is used to provide the name of current operating system.



```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ uname
Linux
aib@pamolil-virtual-machine:~$
```

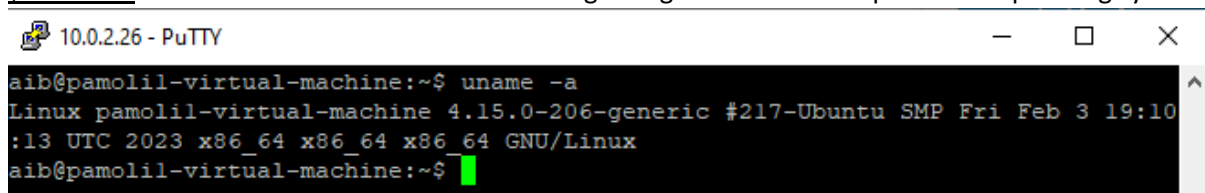
\$uname -version:- It is used to get the current version of operating system.



```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ uname
Linux
aib@pamolil-virtual-machine:~$ uname -r
4.15.0-206-generic
aib@pamolil-virtual-machine:~$ uname --version
uname (GNU coreutils) 8.28
Copyright (C) 2017 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

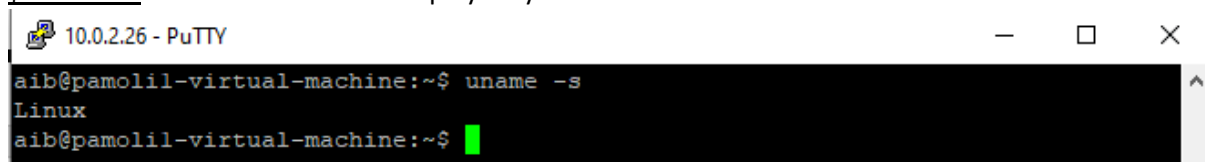
Written by David MacKenzie.
aib@pamolil-virtual-machine:~$
```

\$uname -a:- It is used to view the information regarding the current computer and operating system.



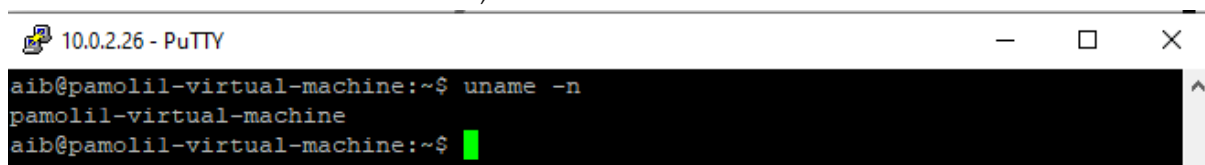
```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ uname -a
Linux pamolil-virtual-machine 4.15.0-206-generic #217-Ubuntu SMP Fri Feb 3 19:10:13 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
aib@pamolil-virtual-machine:~$
```

\$uname -s:- It is used to display only kernel name.



```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ uname -s
Linux
aib@pamolil-virtual-machine:~$
```

\$uname -n:- Hostname is the name by which the computer is identified in the local network. To display the hostname with the uname command, run the command.



```
10.0.2.26 - PuTTY
aib@pamolil-virtual-machine:~$ uname -n
pamolil-virtual-machine
aib@pamolil-virtual-machine:~$
```

\$uname -v:- It is used to print the current version of kernel.

10.0.2.26 - PuTTY

```
aib@pamolil-virtual-machine:~$ uname -v
#217-Ubuntu SMP Fri Feb 3 19:10:13 UTC 2023
aib@pamolil-virtual-machine:~$
```

\$uname -m:- Machine name is the computer's hardware, which can be displayed using the "-m" option with the uname command.

10.0.2.26 - PuTTY

```
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ uname -m
x86_64
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$
```

\$uname -o:- It is used to get the information about the current operating system.

10.0.2.26 - PuTTY

```
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ uname -o
GNU/Linux
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$
```

\$uname -r:- It is used to release the current kernel.

10.0.2.26 - PuTTY

```
aib@pamolil-virtual-machine:~$ uname -r
4.15.0-206-generic
aib@pamolil-virtual-machine:~$
```

4.\$tty:- It is used to get the current terminal name.tty is short form of teletype also know as terminal.

10.0.2.26 - PuTTY

```
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ tty
/dev/pts/27
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$
```

5.\$stty:- It is used to change or print the current terminal line setting.

\$stty -all:- It is used to get all the current setting of terminal in human readable format.

10.0.2.26 - PuTTY

```
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$ stty --all
speed 38400 baud; rows 24; columns 80; line = 0;
intr = ^C; quit = ^\; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; swtch = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R;
werase = ^W; lnext = ^V; discard = ^O; min = 1; time = 0;
-parenb -parodd -cmspar cs8 -hupcl -cstopb cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpck -istrip -inlcr -igncr icrnl ixon -ixoff
-iuclc -ixany -imaxbel iutf8
opost -olcuc -ocrnl onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0 vt0 ff0
isig icanon iexten echo echoe echok -echonl -noflsh -xcase -tostop -echopr
echoctl echoke -flusho -extproc
aib@pamolil-virtual-machine:/home/aibl/home/kumar/abc$
```

6.\$ls -R:-It is used to get all the directories name including its all subdirectories.

10.0.2.26 - PuTTY

```
aib@pamolil-virtual-machine:/home/aib1/abc$ ls -R
.:
abc  ABC  abc.txt

./abc:
ana.txt  ana.txt

./abc/ana.txt:

./abc/ana.txt:

./ABC:
ab.txt
aib@pamolil-virtual-machine:/home/aib1/abc$
```

Task :- 2

- 1) create a directory patch under current directory
- 2) create three subdirectories batch, sd1, sd2 with single commands
- 3) create a directory tree Dir 1 Dir1/progs Dir/data
- 4) mkdir dir1/progs dir1/data whether it will give error or not
- 5) Try to create an already existing directory
- 6) Remove your directory which you have creates in task 3
- 7) Try /home/kumar/abc/progs

10.0.2.26 - PuTTY

```
aib@pamolil-virtual-machine:/home/aib1$ mkdir patch
aib@pamolil-virtual-machine:/home/aib1$ cd /home/aib1/patch
aib@pamolil-virtual-machine:/home/aib1/patch$ mkdir sd1 sd2 batch
aib@pamolil-virtual-machine:/home/aib1/patch$ mkdir dir1 dir1/progs dir1/data
aib@pamolil-virtual-machine:/home/aib1/patch$ mkdir dir1/progs dir1/data
mkdir: cannot create directory 'dir1/progs': File exists
mkdir: cannot create directory 'dir1/data': File exists
aib@pamolil-virtual-machine:/home/aib1/patch$ mkdir dir1
mkdir: cannot create directory 'dir1': File exists
aib@pamolil-virtual-machine:/home/aib1/patch$ rmdir dir1/progs dir1/data
aib@pamolil-virtual-machine:/home/aib1/patch$ rmdir dir1 batch sd1 sd2
aib@pamolil-virtual-machine:/home/aib1/patch$ cd ..
aib@pamolil-virtual-machine:/home/aib1$ rmdir patch
aib@pamolil-virtual-machine:/home/aib1$ /home/kumar/abc/progs
-bash: /home/kumar/abc/progs: No such file or directory
aib@pamolil-virtual-machine:/home/aib1$
```

10.0.2.26 - PuTTY

```
aib@pamolil-virtual-machine:/home/aib1$ mkdir home
aib@pamolil-virtual-machine:/home/aib1$ mkdir home/kumar
aib@pamolil-virtual-machine:/home/aib1$ mkdir home/kumar/abc
aib@pamolil-virtual-machine:/home/aib1$ mkdir home/kumar/abc/progs
aib@pamolil-virtual-machine:/home/aib1$
```


Experiment-3

Write a shell script to check whether a given number is even or odd

Program:

```
1 echo "Enter the number: "  
2 read n  
3 r='expr $n % 2'  
4 if [ $r -eq 0 ]  
5 then  
6 echo "$n is Even number"  
7 else  
8 echo "$n is odd number"  
9 fi  
10
```

Ouput:

```
(kali㉿kali)-[~/Desktop/Operating System]  
$ ./even_odd_check.sh  
Enter the number:  
3  
./even_odd_check.sh: 4: [: expr: unexpected operator  
3 is odd number
```

Experiment-4

Write a shell script to swap to number.

Program:

```
1 echo "enter first number"
2 read a
3 echo "enter second number"
4 read b
5 echo "a before swapping is $a and b is $b"
6 #swapping
7 a=$((a+b))
8 b=$((a - b))
9 a=$((a-b))
10 echo "a after swapping is $a and b is $b"
11
```

Output:

```
(kali㉿kali)-[~/Desktop/Operating System]
$ ./Swaping.sh
enter first number
2
enter second number
3
a before swapping is 2 and b is 3
a after swapping is 3 and b is 2
```

Experiment-5

Write a shell script to calculate the factorial of a given number.

Program:

```
1 echo "Enter a number"
2 read num
3 fact=1
4 while [ $num -gt 1 ]
5 do
6     fact=$((fact * num)) #fact = fact * num
7     num=$((num - 1))     #num = num - 1
8 done
9 echo $fact
10
```

Output:

```
(kali㉿kali)-[~/Desktop/Operating System]
$ ./Factorial.sh
Enter a number
4
24
```

Experiment-6

Q2. Write a shell script to check whether a given year is leap or not.

Program:

```
1 echo "LEAP YEAR SHELL SCRIPT"
2 echo -n "Enter a year:"
3 read year_checker
4 if [ `expr $year_checker % 4` -eq 0 ]
5 then
6     echo "$year_checker is a leap year"
7 else
8     echo "$year_checker is not a leap year"
9 fi
10
```

Output:

```
(kali㉿kali)-[~/Desktop/Operating System]
$ ./Leap_check.sh
LEAP YEAR SHELL SCRIPT
Enter a year:2020
2020 is a leap year

(kali㉿kali)-[~/Desktop/Operating System]
$ ./Leap_check.sh
LEAP YEAR SHELL SCRIPT
Enter a year:2022
2022 is not a leap year
```

Experiment-7

Write a program to implement FCFS scheduling algorithm

Code:

```
#include<stdio.h>
int main(){
    int bursttime[20],waitingtime[20],tat[20],i,n;
    float avgtat,avgwt;
    printf("Enter the processes: ");
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        printf("Enter the burst time for process id[%d]: ",i+1);
        scanf("%d",&bursttime[i]);
    }
    waitingtime[0]=avgwt=0;
    tat[0]=avgtat=bursttime[0];
    for(int i=1;i<n;i++){
        waitingtime[i]=waitingtime[i-1]+bursttime[i-1];
        tat[i]=tat[i-1]+bursttime[i];
        avgwt=avgwt+waitingtime[i];
        avgtat=(avgtat+tat[i]);
    }
    printf("\tProcesses\tBurst Time\tWating Time\tTurnaround Time");
    for(i=0;i<n;i++){
        printf("\n\tP%d\t\t\t%d\t\t\t%d\t\t\t%d",i,bursttime[i],waitingtime[i],tat[i]);
        printf("\nAverage waiting time: %f",avgwt/n);
        printf("\nAverage turnaround time: %f",avgtat/n);
    }
}
```

Output:

Enter the processes: 5

Enter the burst time for process id[1]: 2

Enter the burst time for process id[2]: 4

Enter the burst time for process id[3]: 3

Enter the burst time for process id[4]: 3

Enter the burst time for process id[5]: 1

Processes	Burst Time	Wating Time	Turnaround Time
P0	2	0	2
P1	4	2	6
P2	3	6	9
P3	3	9	12
P4	1	12	13

Average waiting time: 5.800000

Average turnaround time: 8.400000

Experiment – 8

Write a program to implement SJF scheduling algorithm

Code:

```
#include<stdio.h>
int main(){
    int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,pos,temp;
    float avg_wt,avg_tat;
    printf("Enter number of processes: ");
    scanf("%d",&n);
    printf("Enter Burst time: \n");
    for(i=0;i<n;i++){
        printf("P%d: ",i+1);
        scanf("%d",&bt[i]);
        p[i]=i+1;
    }
    for(i=0;i<n;i++){
        pos=i;
        for(j=i+1;j<n;j++){
            if(bt[j]<bt[pos]) pos=j;
        }
        temp=bt[i];
        bt[i]=bt[pos];
        bt[pos]=temp;
        temp=p[i];
        p[i]=p[pos];
        p[pos]=temp;
    }
    wt[0]=0;
    for(i=1;i<n;i++){
        wt[i]=0;
        for(j=0;j<i;j++) wt[i]+=bt[j];
        total+=wt[i];
    }
    avg_wt=(float)total/n;
    total=0;
    printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");
    for(i=0;i<n;i++){
        tat[i]=bt[i]+wt[i];
        total+=tat[i];
        printf("\nP%d\t %d\t\t\t %d\t\t\t\t %d",p[i],bt[i],wt[i],tat[i]);
    }
    avg_tat=(float)total/n;
    printf("\n\nAverage Waiting Time=%f",avg_wt);
    printf("\n\nAverage Turnaround Time=%f",avg_tat);
}
```

Output:

Enter number of processes: 5

Enter Burst time:

P1: 6

P2: 2

P3: 4

P4: 3

P5: 1

Process	Burst Time	Waiting Time	Turnaround Time
---------	------------	--------------	-----------------

P5	1	0	1
----	---	---	---

P2	2	1	3
----	---	---	---

P4	3	3	6
----	---	---	---

P3	4	6	10
----	---	---	----

P1	6	10	16
----	---	----	----

Average Waiting Time=4.000000

Average Turnaround Time=7.200000

Experiment-9

Write a program to implement round robin scheduling algorithm

Code:

```
#include<stdio.h>
void main(){
    int i, NOP, sum=0, count=0, y, quant, wt=0, tat=0, at[10], bt[10],
    temp[10];
    float avg_wt, avg_tat;
    printf(" Total number of process in the system: ");
    scanf("%d", &NOP);
    y = NOP;
    printf("Enter the Arrival Time of: \n");
    for(i=0; i<NOP; i++){
        printf("P[%d]: ", i+1);
        scanf("%d", &at[i]);
    }
    printf("Enter the Burst Time of: \n");
    for(i=0; i<NOP; i++){
        printf("P[%d]: ", i+1);
        scanf("%d", &bt[i]);
        temp[i] = bt[i];
    }
    printf("Enter the Time Quantum for the process: ");
    scanf("%d", &quant);
    printf("\n Process No \t\t Burst Time \t\t TAT \t\t Waiting Time ");
    for(sum=0, i = 0; y!=0; ){
        if(temp[i] <= quant && temp[i] > 0){
            sum = sum + temp[i];
            temp[i] = 0;
            count=1;
        }
        else if(temp[i] > 0){
            temp[i] = temp[i] - quant;
            sum = sum + quant;
        }
        if(temp[i]==0 && count==1){
            y--;
            printf("\nProcess No[%d] \t\t %d\t\t\t %d\t\t\t %d", i+1, bt[i],
sum-at[i], sum-at[i]-bt[i]);
            wt = wt+sum-at[i]-bt[i];
            tat = tat+sum-at[i];
            count =0;
        }
        if(i==NOP-1) i=0;
        else if(at[i+1]<=sum) i++;
        else i=0;
    }
    avg_wt = wt * 1.0/NOP;
    avg_tat = tat * 1.0/NOP;
    printf("\n Average Turn Around Time: \t%f", avg_wt);
    printf("\n Average Waiting Time: \t%f", avg_tat);}
```

Output:

Total number of process in the system: 4

Enter the Arrival Time of:

P[1]: 0

P[2]: 1

P[3]: 2

P[4]: 3

Enter the Burst Time of:

P[1]: 8

P[2]: 5

P[3]: 10

P[4]: 11

Enter the Time Quantum for the process: 6

Process No	Burst Time	TAT	Waiting Time
Process No[2]	5	10	5
Process No[1]	8	25	17
Process No[3]	10	27	17
Process No[4]	11	31	20

Average Turn Around Time: 14.750000

Average Waiting Time: 23.250000

Experiment-10

Write a program to implement Preemptive priority scheduling round robin scheduling algorithm.

Code:

```
#include<stdio.h>
struct process{
    int WT,AT,BT,TAT,PT;
};
struct process a[10];
int main(){
    int n,temp[10],t,count=0,short_p;
    float total_WT=0,total_TAT=0,Avg_WT,Avg_TAT;
    printf("Enter the number of the process\n");
    scanf("%d",&n);
    printf("Enter the arrival time , burst time and priority of the process\n");
    printf("AT BT PT\n");
    for(int i=0;i<n;i++){
        scanf("%d%d%d",&a[i].AT,&a[i].BT,&a[i].PT);
        temp[i]=a[i].BT;
    }
    a[9].PT=10000;
    for(t=0;count!=n;t++){
        short_p=9;
        for(int i=0;i<n;i++){
            if(a[short_p].PT>a[i].PT && a[i].AT<=t && a[i].BT>0) short_p=i;
        }
        a[short_p].BT=a[short_p].BT-1;
        if(a[short_p].BT==0){
            count++;
            a[short_p].WT=t+1-a[short_p].AT-temp[short_p];
            a[short_p].TAT=t+1-a[short_p].AT;
            total_WT=total_WT+a[short_p].WT;
            total_TAT=total_TAT+a[short_p].TAT;
        }
    }
    Avg_WT=total_WT/n;
    Avg_TAT=total_TAT/n;
    printf("ID WT TAT\n");
    for(int i=0;i<n;i++) printf("%d %d\t%d\n",i+1,a[i].WT,a[i].TAT);
    printf("Avg waiting time of the process is %f\n",Avg_WT);
    printf("Avg turn around time of the process is %f\n",Avg_TAT);
    return 0;
}
```

Output:

Enter the number of the process

3

Enter the arrival time , burst time and priority of the process

AT BT PT

0 7 2

2 2 1

4 1 3

ID WT TAT

1 2 9

2 0 2

3 5 6

Avg waiting time of the process is 2.333333

Avg turn around time of the process is 5.666667