Experiment7:

Write a CPU bound C program and a I/O bound C program and observe the effect of their CPU sharing using the top command and its variants.

I/O bound:-

Command: nano filename.c

Program:

```
#include<stdio.h>
#include<time.h>
int main()
{
int j,k,n;
while(1){
printf("\nEnter the any number:");
scanf("%d",&k);
printf("Enter the any number:");
scanf("%d",&j);
n=k%j;
printf("%d",n);
time_t rawtime;
struct tm * timeinfo;
time(&rawtime);
timeinfo=localtime(&rawtime);
```

printf("\n Current local time and date:%s", asctime(timeinfo));

}

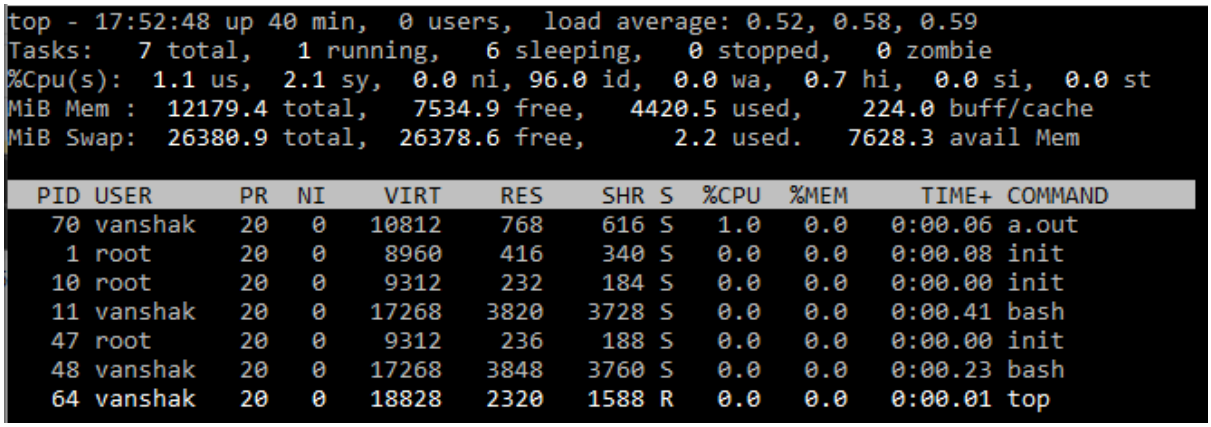
}

Terminal: gcc filename.c

./a.out

```c
#include<stdio.h>
#include<time.h>

int main()
{
int j,k,n;
while(1){

printf("\nEnter the any number:");
scanf("%d",&k);
printf("Enter the any number:");
scanf("%d",&j);
n=k%j;
printf("%d",n);

time_t rawtime;
struct tm * timeinfo;
time(&rawtime);
timeinfo=localtime(&rawtime);
printf("\n Current local time and date:%s", asctime(timeinfo));
}
}
```

```
vanshak@HP-laptop:/mnt/d$ nano exp7a.c
vanshak@HP-laptop:/mnt/d$ gcc exp7a.c
vanshak@HP-laptop:/mnt/d$ ./a.out

Enter the any number:67
Enter the any number:87
67
 Current local time and date:Fri Dec  9 17:52:36 2022

Enter the any number:787
Enter the any number:89
75
 Current local time and date:Fri Dec  9 17:52:40 2022
```



CPU bound:-

Command: nano filename.c

Program: #include<stdio.h>

```
#include<time.h>
void main() {
clock_t start, end;
double runTime;
start=clock();
int i,num=1,primes=0;
while(num<=1000000){
i=2;
while(i<=num){
if(num%i==0)
break;
i++;
}
if(i==num)
primes++;
```

printf("%d prime numbers calculated\n",primes);

num++;

}

end=clock();

runTime=(end-start)/(double) CLOCKS_PER_SEC;

printf("This machine calculated all %d primes numbers under 1000000 in %g seconds\n",primes,runTime);

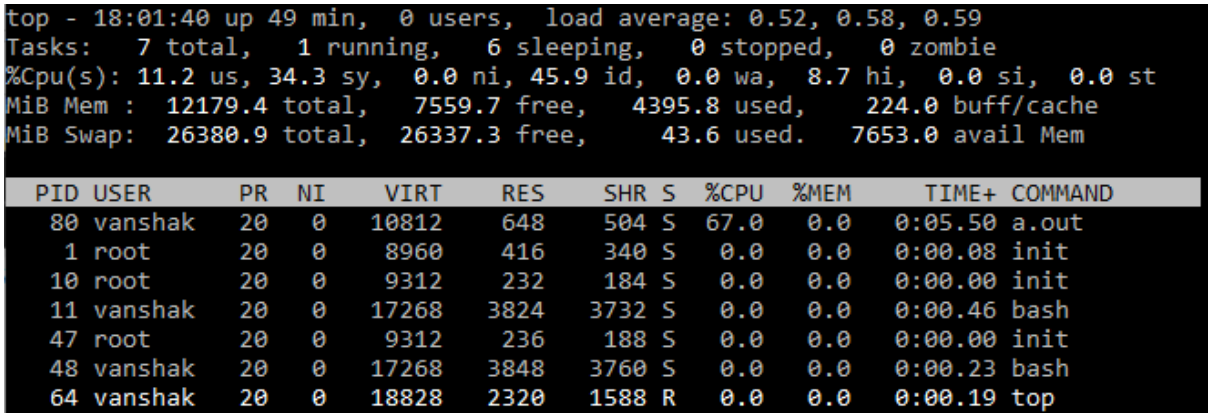
}

Terminal: gcc filename.c

./a.out

```c
#include<stdio.h>
#include<time.h>

void main() {
clock_t start, end;
double runTime;
start=clock();
int i,num=1,primes=0;

while(num<=1000000){
i=2;
while(i<=num){
if(num%i==0)
break;
i++;
}
if(i==num)
primes++;

printf("%d prime numbers calculated\n",primes);
num++;
}

end=clock();
runTime=(end-start)/(double) CLOCKS_PER_SEC;
printf("This machine calculated all %d primes numbers under 1000000 in %g seconds\n",primes,runTime);
}
```

```
78498 prime numbers calculated
78498 prime numbers calculated
78498 prime numbers calculated
78498 prime numbers calculated
78498 prime numbers calculated
This machine calculated all 78498 primes numbers under 1000000 in 211.75 seconds
```


top - 18:01:40 up 49 min, 0 users, load average: 0.52, 0.58, 0.59
Tasks: 7 total, 1 running, 6 sleeping, 0 stopped, 0 zombie
%Cpu(s): 11.2 us, 34.3 sy, 0.0 ni, 45.9 id, 0.0 wa, 8.7 hi, 0.0 si, 0.0 st
MiB Mem : 12179.4 total, 7559.7 free, 4395.8 used, 224.0 buff/cache
MiB Swap: 26380.9 total, 26337.3 free, 43.6 used. 7653.0 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
80 vanshak 20 0 10812 648 504 S 67.0 0.0 0:05.50 a.out
1 root 20 0 8960 416 340 S 0.0 0.0 0:00.08 init
10 root 20 0 9312 232 184 S 0.0 0.0 0:00.00 init
11 vanshak 20 0 17268 3824 3732 S 0.0 0.0 0:00.46 bash
47 root 20 0 9312 236 188 S 0.0 0.0 0:00.00 init
48 vanshak 20 0 17268 3848 3760 S 0.0 0.0 0:00.23 bash
64 vanshak 20 0 18828 2320 1588 R 0.0 0.0 0:00.19 top