APPLICATION USING MULTITHREADS

```
kaushik@kaushik-AcerPower-Series:~/OS $ cat thread.c
#include<pthread.h>
#include<stdio.h>
int avg,min,max;
void *worker_avg(void *p);
void *worker_min(void *p);
void *worker_max(void *p);
struct input
{
char **data;
int count;
}in;
void main(int argc,char* argv[])
{
pthread_t ptid,tid1,tid2,tid3;
pthread_attr_t attr;
pthread_attr_init(&attr);
in.data=argv;
in.count=argc;
pthread_create(&tid1,&attr,worker_avg,&in);
pthread_create(&tid2,&attr,worker_min,&in);
pthread_create(&tid3,&attr,worker_max,&in);
pthread_join(tid1,NULL);
pthread_join(tid2,NULL);
pthread_join(tid3,NULL);
printf("\nAaverage=%d",avg);
printf("\nMinimum=%d",min);
printf("\nMaximum=%d",max);
}
void *worker_avg(void *p)
{
struct input *t=(struct input*)p;
int i,sum=0;
for(i=1;i<t->count;i++)
sum+=atoi(t->data[i]);
avg=sum/(t->count-1);
pthread_exit(0);
}
void *worker_min(void *p)
{
struct input *t=(struct input*)p;
int i;
min=atoi(t->data[1]);
for(i=2;i<t->count;i++)
if(atoi(t->data[i])<min)
min=atoi(t->data[i]);
pthread_exit(0);
}
void *worker_max(void *p)
{
```

```
struct input *t=(struct input*)p;
int i;
max=atoi(t->data[1]);
for(i=2;i<t->count;i++)
if(atoi(t->data[i])>max)
max=atoi(t->data[i]);
pthread_exit(0);
}
kaushik@kaushik-AcerPower-Series:~/OS $ gcc -pthread -w -o th thread.c
kaushik@kaushik-AcerPower-Series:~/OS $ ./th 60 12 34 54 21 22

Aaverage=33
Minimum=12
Maximum=60
kaushik@kaushik-AcerPower-Series:~/OS $
```