

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

```
int main()
```

```
{
```

```
    int i,j,n,bu[10],wa[10],tat[10],ct[10],t,max;
```

```
    float awt=0,att=0,temp=0;
```

```
    printf("Enter the no. of processes -- ");
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<n;i++)
```

```
    {
```

```
        printf("\nEnter Burst Time for process %d-- ",i+1);
```

```
        scanf("%d",&bu[i]);
```

```
        ct[i]=bu[i];
```

```
    }
```

```
    printf("\nEnter the size of time slice -- ");
```

```
    scanf("%d",&t);
```

```
    max=bu[0];
```

```
    for(i=1;i<n;i++)
```

```
        if(max<bu[i])
```

```
            max=bu[i];
```

```
    for(j=0;j<(max/t)+1;j++)
```

```
        for(i=0;i<n;i++)
```

```
            if(bu[i] != 0)
```

```
                if(bu[i]<=t)
```

```
                {
```

```
                    tat[i]=temp+bu[i];
```

```
                    temp=temp+bu[i];
```

```

        bu[i]=0;
    }
    else
    {
        bu[i]=bu[i]-t;
        temp=temp+t;
    }

for(i=0;i<n;i++)
{
    wa[i]=tat[i]-ct[i];
    att+=tat[i];
    awt+=wa[i];
}

printf("\nThe Average TurnAround time is --%f\n",att/n);
printf("\nThe Average Waiting time is --%f\n",awt/n);
printf("\n\tPROCESS\t BURST TIME\t WAITING TIME\t TURNAROUND TIME\n");

for(i=0;i<n;i++)
    printf("\t%d \t %d \t\t%d \t\t%d\n",i+1,ct[i],wa[i],tat[i]);

return 0;
}

```

Mutex

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

```
#include<stdlib.h>
```

```
int mutex=1,full=0,empty=3,x=0;
```

```
void producer();
```

```
void consumer();
```

```
int wait(int);
```

```
int signal(int);
```

```
int main()
```

```
{
```

```
    int n;
```

```
    printf("\n1.Producer\n2.Consumer\n3.Exit");
```

```
    while(1)
```

```
    {
```

```
        printf("\nEnter your choice:");
```

```
        scanf("%d",&n);
```

```
        switch(n)
```

```
        {
```

```
            case 1:
```

```
                if((mutex==1) && (empty!=0))
```

```
                    producer();
```

```
                else
```

```
                    printf("Buffer is full!!");
```

```
                break;
```

```
            case 2:
```

```
                if((mutex==1) && (full!=0))
```

```
        consumer();
    else
        printf("Buffer is empty!!");
    break;
case 3:
    exit(0);
break;
}
}
return 0;
}
```

```
int wait(int s)
{
    return (--s);
}
```

```
int signal(int s)
{
    return (++s);
}
```

```
void producer()
{
    mutex=wait(mutex);
    full=signal(full);
    empty=wait(empty);
    x++;
}
```

```
    printf("\nProducer produces the item %d",x);  
    mutex=signal(mutex);  
}
```

```
void consumer()  
{  
    mutex=wait(mutex);  
    full=wait(full);  
    empty=signal(empty);  
    printf("\nConsumer consumes item %d",x);  
    x--;  
    mutex=signal(mutex);  
}
```

Peterson

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

```
#include<pthread.h>
```

```
#include<unistd.h>
```

```
int flag[2]={0, 0};
```

```
int turn =0;
```

```
void *process(void *param)
```

```
{
```

```
    int i=*(int *)param;
```

```
    int other = 1-i;
```

```
    for(int cnt=0; cnt < 5; cnt++)
```

```
    {
```

```
        flag[i]=1;
```

```
        turn =other;
```

```
        while (flag[other] && turn==other)
```

```
        {
```

```
        }
```

```
        printf("Process %d is in the critical section (iteration %d)\n",i,cnt+1);
```

```
        sleep(1);
```

```
        flag[i]=0;
```

```
        printf("Process %d is in the remainder section\n",i);
```

```
        sleep(1);
```

```
    }
```

```
    return NULL;
```

```
}
```

```
int main()
{
    pthread_t t1,t2;
    int id1=0, id2=1;
    pthread_create(&t1,NULL,process,&id1);
    pthread_create(&t2,NULL,process,&id2);
    pthread_join(t1,NULL);
    pthread_join(t2,NULL);
    printf("Both processes have finished.\n");
    return 0;
}
```