

LAB 03

- **First Come First Served - Example**
- **Shortest Job First - Example**
- Round Robin - Exercise
- Priority Scheduling - Exercise
- Earliest Deadline First - Exercise

```
#include<stdio.h>

int main()
{
    int numofProcess,burstTime[10],waitingTime[10],turnAroundTime[10],averageWaitingTime=0,i,j;
    do{
        printf("Number of processes:");
        scanf("%d",&numofProcess);
        if (numofProcess>10 || numofProcess <1)
        {
            printf("Please enter a process number between 1 and 10 !!!\n");
            printf("\n");
        }
    } while(numofProcess>10 || numofProcess <1);

    printf("\nEnter Burst Times for each Processor\n");
    for(i=0; i<numofProcess; i++)
    {
        printf("P(%d):",i+1);
        scanf("%d",&burstTime[i]);
    }
    waitingTime[0]=0;//waiting time first unit is 0
    for(i=1; i<numofProcess; i++)
    {
        waitingTime[i]=0;
        for(j=0;j<i;j++)
            waitingTime[i]+=burstTime[j];
    }

    printf("\nProcess Number          Burst Time          Waiting Time          TurnAround Time");
    for(i=0; i<numofProcess; i++)
    {
        turnAroundTime[i]= burstTime[i]+ waitingTime[i];
        averageWaitingTime += waitingTime[i];
        printf("\nP(%d)   \t\t\t%d\t\t\t%d\t\t\t%d",i+1,burstTime[i],waitingTime[i],turnAroundTime[i]);
    }

    averageWaitingTime /= i;
    printf("\n\nAverage Waiting Time: %d\n",averageWaitingTime);
    return 0;
}
```

Exercise 2. Shortest Job First Example

```
#include<stdio.h>
void main()
{
    int burstTime[10],p[10],waitingtTime[10],turnAroundTime[10];
    int i,j,k,temp,numOfProcess,total=0;
    float averageWaitingTime;
    do{
        printf("Number of processes:");
        scanf("%d",&numOfProcess);
        if (numOfProcess>10 || numOfProcess <1)
        {
            printf("Please enter a process number between 1 and 10 !!!\n");
            printf("\n");
        }
    }
    while(numOfProcess>10 || numOfProcess <1);
    for(i = 0; i < numOfProcess; i++)
    {
        printf("Enter Burst Time for per Processor P(%d): ",i+1);
        scanf("%d",&burstTime[i]);
        p[i]=i+1;
    }
    for(i=0;i<numOfProcess;i++)
    {
        k=i;
        for(j=i+1;j<numOfProcess;j++)
        {
            if(burstTime[j]<burstTime[k])
                k=j;
        }
        temp=burstTime[i];
        burstTime[i]=burstTime[k];
        burstTime[k]=temp;
        temp=p[i];
        p[i]=p[k];
        p[k]=temp;
    }
    waitingtTime[0]=0;
    for(i=1;i<numOfProcess;i++)
    {
        waitingtTime[i]=0;
        for(j=0;j<i;j++)
            waitingtTime[i]+=burstTime[j];
        total+=waitingtTime[i];
    }
    averageWaitingTime=(float)total/numOfProcess;
    total=0;
    printf("\nProcess Number\t Burst Time\t\t\t\t\t Waiting Time\t\t\t\t\t \tTurnaround Time");
    for(i=0;i<numOfProcess;i++)
    {
        turnAroundTime[i]=burstTime[i]+waitingtTime[i];
        total+=turnAroundTime[i];
        printf("\nP(%d)\t\t\t\t\t %d\t\t\t\t\t %d\t\t\t\t\t %d",p[i],burstTime[i],waitingtTime[i],turnAroundTime[i]);
    }
    printf("\n\nAverage Waiting Time=%.2f",averageWaitingTime);
    printf("\n");
}
```

Exercise 3. Write a code for

- Round Robin - Exercise
- Shortest Job First - Exercise
- Priority Scheduling - Exercise
- Earliest Deadline First - Exercise