

Program no- 5

Implementation of CPU scheduling algorithms. a) Round Robin b) SJF c) FCFS d) Priority

a) Round Robin

```
#include<stdio.h>
int main()
{
    int i, n, total = 0, temp, count= 0, qt;
    int wt[10], bt[10],tat[10], rem_bt[10];
    float awt=0, atat=0;
    printf("\nEnter Total Number of Processes:\n");
    scanf("%d", &n);
    printf("\nEnter burst time of Processes:\n");
    for(i = 0; i < n; i++)
    {
        scanf("%d", &bt[i]);
        rem_bt[i]=bt[i];
    }
    printf("\nEnter Time Quantum:\t");
    scanf("%d", &qt);

    while(1)
    {
        for(i=0,count=0;i<n;i++)
        {
            temp=qt;
            if(rem_bt[i]==0)
            {
                count++;
                continue;
            }
            if(rem_bt[i]>qt)
                rem_bt[i]= rem_bt[i]-qt;
            else
                if(rem_bt[i]>=0)
                {
                    temp=rem_bt[i];
                    rem_bt[i]=0;
                }
            total=total+temp;
            tat[i]=total;
        }
        if(n==count)
            break;
    }
}
```



```

        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("\n%d\t\t %d\t\t %d\t\t%d",p[i],bt[i],wt[i],tat[i]);
    }

    avg_tat=(float)total/n;
    printf("\nAverage Waiting Time=%f",avg_wt);
    printf("\nAverage Turnaround Time=%f\n",avg_tat);
}

```

c)FCFS

```

#include<stdio.h>

int main()

{
    int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;
    printf("Enter total number of processes(maximum 20):");
    scanf("%d",&n);

    printf("\nEnter Process Burst Time\n");
    for(i=0;i<n;i++)
    {
        printf("P[%d]:",i+1);
        scanf("%d",&bt[i]);
    }
}

```

```

    }

    wt[0]=0;

    for(i=1;i<n;i++)
    {
        wt[i]=0;
        for(j=0;j<i;j++)
            wt[i]+=bt[j];
    }

    printf("\nProcess\t\tBurst Time\tWaiting Time\tTurnaround Time");

    for(i=0;i<n;i++)
    {
        tat[i]=bt[i]+wt[i];
        avwt+=wt[i];
        avtat+=tat[i];
        printf("\nP[%d]\t\t%d\t\t%d\t\t%d",i+1,bt[i],wt[i],tat[i]);
    }

    avwt/=i;
    avtat/=i;
    printf("\nAverage Waiting Time:%d",avwt);
    printf("\nAverage Turnaround Time:%d",avtat);

    return 0;
}

```

d) Priority

```

#include<stdio.h>

int main()
{
    int bt[20],p[20],wt[20],tat[20],pr[20],i,j,n,total=0,pos,temp,avg_wt,avg_tat;
    printf("Enter Total Number of Process:");
    scanf("%d",&n);

    printf("\nEnter Burst Time and Priority\n");
    for(i=0;i<n;i++)
    {
        printf("\nP[%d]\n",i+1);
        printf("Burst Time:");
        scanf("%d",&bt[i]);
        printf("Priority:");
        scanf("%d",&pr[i]);
        p[i]=i+1;    //contains process number
    }
}

```

```

}

//sorting burst time, priority and process number in ascending order using selection sort
for(i=0;i<n;i++)
{
    pos=i;
    for(j=i+1;j<n;j++)
    {
        if(pr[j]<pr[pos])
            pos=j;
    }

    temp=pr[i];
    pr[i]=pr[pos];
    pr[pos]=temp;

    temp=bt[i];
    bt[i]=bt[pos];
    bt[pos]=temp;

    temp=p[i];
    p[i]=p[pos];
    p[pos]=temp;
}

wt[0]=0; //waiting time for first process is zero

//calculate waiting time
for(i=1;i<n;i++)
{
    wt[i]=0;
    for(j=0;j<i;j++)
        wt[i]+=bt[j];

    total+=wt[i];
}

avg_wt=total/n;    //average waiting time
total=0;

printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");
for(i=0;i<n;i++)
{
    tat[i]=bt[i]+wt[i];    //calculate turnaround time
    total+=tat[i];
    printf("\nP[%d]\t\t %d\t\t %d\t\t\t%d",p[i],bt[i],wt[i],tat[i]);
}

avg_tat=total/n;    //average turnaround time

```

```
printf("\n\nAverage Waiting Time=%d",avg_wt);  
printf("\nAverage Turnaround Time=%d\n",avg_tat);  
  
return 0;  
}
```