Ex. No.: 7 c
Date: 2.4.24

PRIORITY SCHEDULING

Aim:

To implement priority scheduling technique

Algorithm:

1. Get the number of processes from the user.
2. Read the process name, burst time and priority of process.
3. Sort based on burst time of all processes in ascending order based priority
4. Calculate the total waiting time and total turnaround time for each process
5. Display the process name & burst time for each process.
6. Display the total waiting time, average waiting time, turnaround time

Program Code:

```
#include <stdio.h>
int main(){
    ~~int bt[20]~~
    int n,i,j,pos,temp,avgwt,avgtat,total=0;
    printf(" enter ~~total~~ no.of processes:   ");
    scanf("%d", &n);
    int bt[n], p(n],wt[n],tat[n],priority[n];
    printf("\nenter Burst Time & 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", &priority[i]);

            p[i]=i+1;
    }
~~//sorting~~
```

```
// sorting bt, ~~process~~ priority, & process no. in asc using selection
```

```
for(i=0;i<n;i++){
        pos=i;
        for(j=i+1;j<n;j++){
                if(priority[j]<priority[pos])
                pos=j;
        }
        temp=priority[i];
        priority[i]=priority[pos];
        priority[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;
// calculate WT
        for(i=1;i<n;i++){
                wt[i]=0;
                for(j=0;j<i;j++){ wt[i]+=bt[j];}
                total+=wt[i];      ~~total +=wt[i]~~
                }
 avgwt=total/n;
 total=0;
 printf("\nProcess\t\t BT \t\t WT \t\t TAT");
 for(i=0;i<n;i++){
        tat[i]=bt[i]+wt[i];
        total+=tat[i];
        printf("\nP%d \t\t %d \t\t %d \t\t %d", p[i],bt[i],wt[i],tat[i]);
 }
 avgtat=total/n;
 printf("\n Avg ~~TAT~~ WT : %d", avgwt);
 printf("\n Avg TAT : %d", avgtat);
 return 0;
}
```

Output:

```
Enter Total no. of processes : 3
 Enter Burst Time & priority

P1
Burst Time: 3
 Priority: 6
P2
Burst Time: 10
 Priority:  1
P3
Burst Time: 2
 Priority : 2

 PROCESS      BT     WT     TAT
   P2         10     0      10
   P3         2      10     12
   P1         3      12     15

   Average WT   : 7 ms
   Average TAT  : 12 ms
```

RESULT :

The program has been compiled and executed successfully.

[illegible] 23/4/4 10/10