Ex. No.: 7 c
Date: 10-04-2024

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>

void swap(int *a,int *b)
{
   int temp=*a;
   *a=*b;
   *b=temp;
}

int main()
{
   int n;
   printf("Enter Number of Processes: ");
   scanf("%d",&n);

   int b[n],p[n],index[n];
   for(int i=0;i<n;i++)
   {
      printf("Enter Burst Time and Priority Value for Process %d: ",i+1);
      scanf("%d %d",&b[i],&p[i]);
      index[i]=i+1;
   }
   for(int i=0;i<n;i++)
   {
      int a=p[i],m=i;
      for(int j=i;j<n;j++)
      {
         if(p[j] > a)
         {
            a=p[j];
            m=j;
         }
      }
```

```
        swap(&p[i], &p[m]);
        swap(&b[i], &b[m]);
        swap(&index[i],&index[m]);
    }

    int t=0;
    printf("Order of process Execution is\n");
    for(int i=0;i<n;i++)
    {
        printf("P%d is executed from %d to %d\n",index[i],t,t+b[i]);
        t+=b[i];
    }
    printf("\n");
    printf("Process Id     Burst Time   Priority   Wait Time    TurnAround Time\n");
    int wait_time=0;
    for(int i=0;i<n;i++)
    {
        printf("P%d         %d         %d         %d         %d\n",index[i],b[i],p[i],wait_time,wait_ti
me + b[i]);
        wait_time += b[i];
    }
    return 0;
}
```

Output:

```
Enter Number of Processes: 4
Enter Burst Time and Priority Value for Process 1: 10 2
Enter Burst Time and Priority Value for Process 2: 4 1
Enter Burst Time and Priority Value for Process 3: 5 3
Enter Burst Time and Priority Value for Process 4: 5 0
Order of process Execution is
P3 is executed from 0 to 5
P1 is executed from 5 to 15
P2 is executed from 15 to 19
P4 is executed from 19 to 24

Process Id     Burst Time   Priority   Wait Time    TurnAround Time
P3          5          3          0          5
P1          10          2          5          15
P2          4          1          15          19
P4          5          0          19          24
```

Result:

Hence the C program to implement the priority scheduling algorithm has been successfully executed and the output is verified.