**Experiment No. 08**

**Experiment Name: Implementation of Priority Scheduling algorithm .**

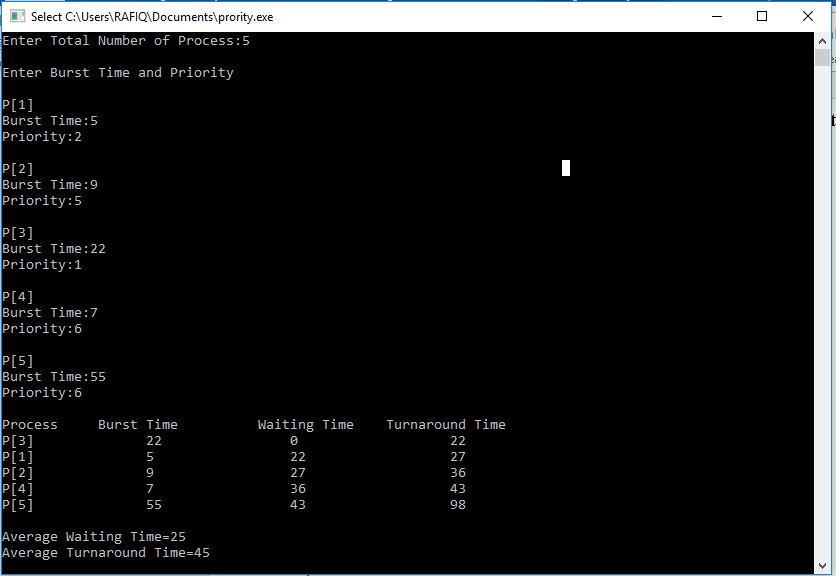
**Objectives:**

* What is Priority Scheduling algorithm.
* How to implementation in C

**Source Code:**

|  |
| --- |
| #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  }  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);  printf("\n");  return 0;  } |

**Output:**



**Conclusion:**  In priority scheduling algorithm each process has a priority associated with it and as each process hits the queue, it is stored in based on its priority so that process with higher priority are dealt with first. It should be noted that equal priority processes are scheduled in FCFS order. To prevent high priority processes from running indefinitely the scheduler may decrease the priority of the currently running process at each clock tick (i.e., at each clock interrupt).