Lab Report No : 09

Lab Report Name : Implementation priority Scheduling algorithm

Theory

In the shortest job first scheduling algorithm, the priority of a process is generally the inverse of the CPU burst time, i.e. the larger the burst time the lower is the priority of that process.

In case of priority scheduling the priority is not always set as the inverse of the CPU burst time, rather it can be internally or externally set, but yes the scheduling is done on the basis of priority of the process where the process which is most urgent is processed first, followed by the ones with lesser priority in order.

Processes with same priority are executed in FCFS manner.

The priority of process, when internally defined, can be decided based on **memory** requirements, time limits ,number of open files, ratio of I/O burst to CPU burst etc.

Whereas, external priorities are set based on criteria outside the operating system, like the importance of the process, funds paid for the computer resource use, makrte factor etc.

Types of Priority Scheduling Algorithm

Priority scheduling can be of two types:

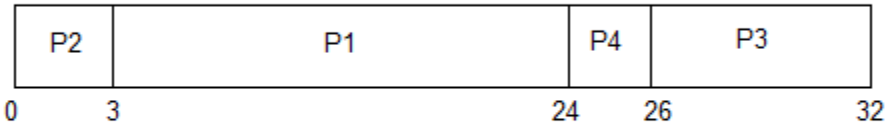
1. Preemptive Priority Scheduling: If the new process arrived at the ready queue has a higher priority than the currently running process, the CPU is preempted, which means the processing of the current process is stoped and the incoming new process with higher priority gets the CPU for its execution.
2. Non-Preemptive Priority Scheduling: In case of non-preemptive priority scheduling algorithm if a new process arrives with a higher priority than the current running process, the incoming process is put at the head of the ready queue, which means after the execution of the current process it will be processed.

Example of Priority Scheduling Algorithm

Consider the below table fo processes with their respective CPU burst times and the priorities.

PROCESS	BURST TIME	PRIORITY
P1	21	2
P2	3	1
P3	6	4
P4	2	3

The GANTT chart for following processes based on Priority scheduling will be,



The average waiting time will be, (0 + 3 + 24 + 26)/4 = 13.25 ms

Implementation priority Scheduling algorithm in c

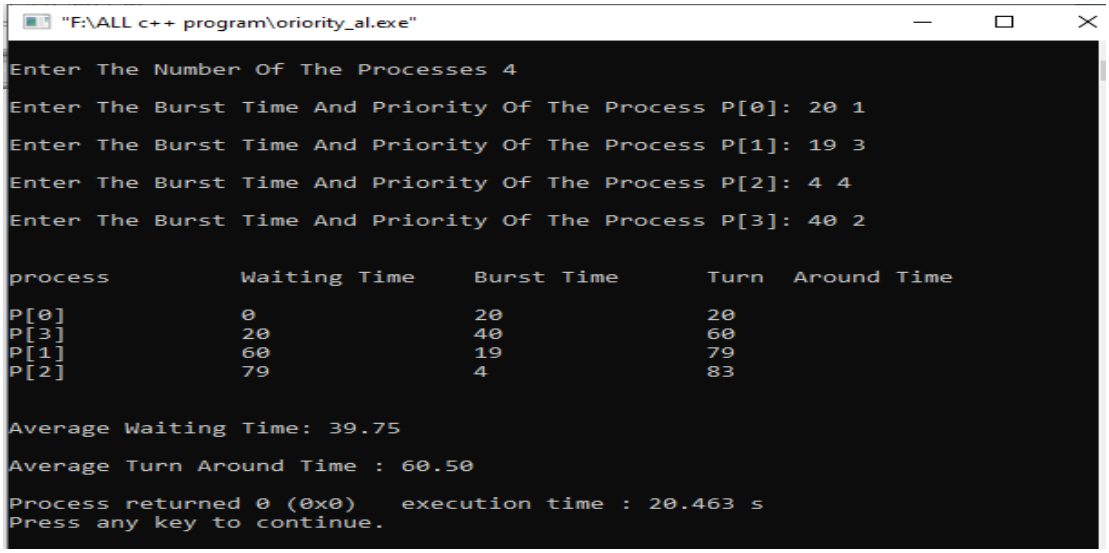
```
#include <stdio.h>

int main()

{

  float  awt =0,atat=0;
```

```
int x,tem,key,i,j;

  int tart[20],pri[20],bst[20],wat[20],p[20];



  printf("\nEnter The Number Of The Processes ");

  scanf("%d",&x);

  for(i=0;i<x;i++)

  {

    printf("\nEnter The Burst Time And Priority Of The Process P[%d]: ",i);

    scanf("%d",&bst[i]);
    scanf("%d",&pri[i]);
    p[i]=i;

  }

  for(i=0;i<x;i++)
  {
    key=i;
    for(j=i+1;j<x;j++)
    {
      if(pri[j]<pri[key])
      {
        key=j;
```

```
    }
  }
  tem=bst[i];
  bst[i]=bst[key];
  bst[key]=tem;

  tem=pri[i];
  pri[i]=pri[key];
  pri[key]=tem;

  tem=p[i];
  p[i]=p[key];
  p[key]=tem;
}



wat[0]=0;
tart[0]=bst[0];
atat=tart[0];

for(i=1;i<x;i++)
{

    wat[i]=wat[i-1]+bst[i-1];

    tart[i]=tart[i-1]+bst[i];

    awt+=wat[i];

    atat+=tart[i];
```

```
    }

    awt=awt/x;
    atat=atat/x;

    printf("\n\nprocess\t\tWaiting Time\tBurst Time\tTurn  Around Time\n");

    printf("\n");

    for(i=0;i<x;i++)
    {

        printf("P[%d]\t\t%d\t\t%d\t\t%d\n",p[i],wat[i],bst[i],tart[i]);

    }

    printf("\n\nAverage Waiting Time: %.2f",awt);
    printf("\n\nAverage Turn Around Time : %.2f\n",atat);

}
```

output:



Discussion : In this type of scheduling algorithm each process has a priority associated with it and when each process hits the queue it is stored at position based on its priority. The process with the higher priority will be dealt first. But if there are two process has same priority then they will execute according to FCFS scheduling algorithm.