**PREEMPTIVE PRIORITY SJF ALGORITHM**

**ABSTRACT:**

Preemptive Priority SJF scheduling algorithm has been developed by combining preemptive priority scheduling and SJF scheduling algorithms. This algorithm produces an optimal solution than preemptive priority scheduling .As the name implies we first pick the set of highest priority jobs then from that set we select the shortest job .In this case we presume that the burst time represent the expected execution time.

**INTRODUCTION:**

Priority Scheduling is a method of scheduling based on priority. It involves priority assignment to every process and processes with higher priorities are carried out first, whereas tasks with equal priorities are carried out on first come fist served basis. Priority scheduling can be

Preemptive scheduling

Non-preemptive scheduling

Preemptive priority scheduling may preempt the CPU in the case the priority of the freshly arrived process being greater than those that of existing processes.

Indefinite blocking otherwise called as starvation is one of the major issues concerning priority scheduling algorithms. It is a state where process is ready to be executed but faces a long wait in getting assigned to CPU.

When all processes have equal priority, the processes are assigned to CPU similar to FCFS algorithm. This is the major disadvantage of preemptive priority scheduling.

This can be overcome by our project, that is, Preemptive Priority scheduling SJF scheduling algorithm.

**AIM:**

To implement Preemptive Priority SJF scheduling algorithm.

**SOURCE CODE:**

#include<iostream>

#include<cstdio>

#include<fstream>

#include<iomanip>

using namespace std;

typedef struct prpsjf

{

int at,bt,ft,tat,wt,pr,id;

}prpsjf;

prpsjf p[20],p1[20],temp;

int ts;

int accept();

void prpsjfsort(int n);

void ganttprpsjf(int n);

int main()

{

int np;

np=accept();

prpsjfsort(np);

ganttprpsjf(np);

}

int accept() //To get the input from File

{

int np,i;

ifstream infile;

infile.open("inp.txt");

cout<<"The no of process: ";

infile>>np;

cout<<np;

for(i=1;i<=np;i++) //To get Arrival Time

{

cout<<"\nThe arrival time P"<<i<<": ";

infile>>p[i].at;

cout<<p[i].at;

p[i].id=i;

}

for(i=1;i<=np;i++) //To get Burst Time

{

cout<<"\nThe burst time P"<<i<<": ";

infile>>p[i].bt;

cout<<p[i].bt;

}

for(i=1;i<=np;i++) //To get Priority

{

cout<<"\nThe priority for the process P"<<i<<": ";

infile>>p[i].pr;

cout<<p[i].pr;

}

for(i=1;i<=np;i++)

p1[i]=p[i];

cout<<"\nThe timespan: ";

infile>>ts;

cout<<ts;

return np;

}

void prpsjfsort(int n) //To sort all the processes

{

int i,j;

for(i=2;i<=n;i++)

for(j=1;j<=i;j++)

if(p[j].at>p[i].at)

{

temp=p[i];

p[i]=p[j];

p[j]=temp;

}

else if(p[j].at==p[i].at)

if(p[j].bt>p[i].bt)

{

temp=p[i];

p[i]=p[j];

p[j]=temp;

}

}

void ganttprpsjf(int n) //To display Gant Chart

{

int i,m,min;

float a\_tt=0,a\_wt=0;0

int limit,nextval;

limit=nextval=p[1].at;

for(i=1;i<=n;i++)

limit+=p[i].bt;

cout<<"\n\n-------------Gantt Chart---------\n";

cout<<p[1].at;

do

{

min = 9999;

for(i=1;p[i].at<=nextval && i<=n ;i++)

{

if(p[i].pr<=min && p[i].bt>0)

{

if(p[i].pr<min)

{

m=i;

min=p[i].pr;

}

else if(p[i].pr==min)

{

if(p[m].bt>p[i].bt)

{

m=i;

min=p[i].pr;

}

}

}

}

if(p[m].bt>=ts)

nextval=nextval+ts;

else

nextval=nextval+p[m].bt;

cout<<"->P"<<p[m].id<<"->"<<nextval;

if(p1[m].bt>=ts)

p[m].bt=p[m].bt-ts;

else

p[m].bt=0;

if(p[m].bt==0)

{

p[m].ft=nextval;

p[m].tat=p[m].ft-p[m].at;

p[m].wt=p[m].tat-p1[m].bt;

p[0].tat+=p[m].tat;

p[0].wt+=p[m].wt;

}

}while(nextval<limit);

p[0].tat=p[0].tat/n;

p[0].wt=p[0].wt/n;

cout<<"\n\n-------------------TABLE----------------------------------\n";

cout<<"\nProcess\tAT\tBT\tFT\tPR\tTAT\tWT\n";

for(i=1;i<=n;i++)

{

printf("\nP%d\t%d\t%d\t%d\t%d\t%d\t%d\n",p[i].id,p[i].at,p1[i].bt,p[i].pr,p[i].ft,p[i].tat,p[i].wt);

cout<<"\n\n-----------------------------------------------------------\n";

}

cout<<”The average waiting time “<<(float)a\_wt/(float)n<<endl;

cout<<”The average turnaround time “<<(float)a\_tt/(float)n <<endl;

}

**OUTPUT:**

The no of process: 15

The arrival time P1: 0

The arrival time P2: 2

The arrival time P3: 4

The arrival time P4: 6

The arrival time P5: 8

The arrival time P6: 10

The arrival time P7: 12

The arrival time P8: 14

The arrival time P9: 16

The arrival time P10: 18

The arrival time P11: 20

The arrival time P12: 22

The arrival time P13: 24

The arrival time P14: 26

The arrival time P15: 28

The burst time P1: 4

The burst time P2: 10

The burst time P3: 2

The burst time P4: 20

The burst time P5: 2

The burst time P6: 12

The burst time P7: 8

The burst time P8: 14

The burst time P9: 7

The burst time P10: 2

The burst time P11: 5

The burst time P12: 15

The burst time P13: 4

The burst time P14: 13

The burst time P15: 10

The priority for the process P1: 13

The priority for the process P2: 2

The priority for the process P3: 1

The priority for the process P4: 3

The priority for the process P5: 5

The priority for the process P6: 8

The priority for the process P7: 12

The priority for the process P8: 15

The priority for the process P9: 9

The priority for the process P10: 6

The priority for the process P11: 10

The priority for the process P12: 7

The priority for the process P13: 11

The priority for the process P14: 14

The priority for the process P15: 4

The timespan: 10

-------------Gantt Chart---------

0->P1->4->P3->6->P2->16->P4->26->P4->36->P15->46->P5->48->P10->50->P12->60->P12->65->P6->75->P6->77->P9->84->P11->89->P13->93->P7->101->P14->111->P14->114->P8->124->P8->128

-------------------TABLE----------------------------------

Process AT BT FT PR TAT WT

P1 0 4 13 4 4 0

P2 2 10 2 16 14 4

P3 4 2 1 6 2 0

P4 6 20 3 36 30 10

P5 8 2 5 48 40 38

P6 10 12 8 0 0 0

P7 12 8 12 101 89 81

P8 14 14 15 0 0 0

P9 16 7 9 84 68 61

P10 18 2 6 50 32 30

P11 20 5 10 89 69 64

P12 22 15 7 0 0 0

P13 24 4 11 93 69 65

P14 26 13 14 0 0 0

P15 28 10 4 46 18 8

The average waiting time 24.0667

The average turnaround time 29

**RESULT:**

Thus the Preemptive Priority SJF scheduling algorithm has been executed successfully.