NAME : D.Nihitha Reddy

Reg.No : 11706226

Email : [nihithachinnu1215@gmail.com](mailto:nihithachinnu1215@gmail.com)

Github Link :

DESCRIPTION :

Consider a scheduling approach which is non pre-emptive similar to shortest job next in nature. The priority of each job is dependent on its estimated run time, and also the amount of time it has spent waiting. Jobs gain higher priority the longer they wait, which prevents indefinite postponement. The jobs that have spent a long time waiting compete against those estimated to have short run times. The priority can be computed as : Priority = 1+ Waiting time / Estimated run time Write a program to implement such an algorithm.

CONDITIOS :

* We interchange the given processes by their arrival times in the ascending order
* And also check weather the processes arrival time is less than the previous process burst time we interchange again in them
* we calculate waiting time from than turnaround time by turnaroundtime=waiting time + burst time
* At last we calculate the priority according to the given formula
* : Priority = 1+ Waiting time / Estimated run time

INPUT:

6

0

4

4

2

2

CODE :

#include<stdio.h>

#include<conio.h>

struct p

{

int p;

int bt;

int at;

int pr;

int wt;

int tt;

};

int main()

{

int a,i,j;

printf("enter no.of processers\n");

scanf("%d",&a);

struct p given[a];

struct p x[a];

struct p y[a];

printf("enter processer details\n1.processer number\n2.burst time\n3.arrival time\n");

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

{

given[i].p=i;

printf("processer %d \n",i);

printf("burst time\n");

scanf("%d",&given[i].bt);

if(given[i].bt<0)

exit(0);

printf("arrival time\n");

scanf("%d",&given[i].at);

if(given[i].at<0)

exit(0);

}

for(i=0;i<a-1;i++)

{

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

{

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

{

x[i]=given[i];

given[i]=given[j];

given[j]=x[i];

}

}

}

given[0].wt=0;

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

{

given[i].wt=0;

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

{

given[i].wt+=given[j].bt;

given[i].wt=given[i].wt-given[i].at;

}

}

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

{

if(given[i].wt<0)

{

given[i].wt=0;

}

}

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

{

given[i].tt=given[i].wt+given[i].bt;

}

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

{

y[i].pr=given[i].wt/given[i].bt;

y[i].pr=1+y[i].pr;

given[i].pr=y[i].pr;

}

printf("============================================================================================\n");

printf("1.processer number\t2.burst time\t3.arrival time\t4.priority\t5.waiting\t6.turnaroundtime\n");

printf("============================================================================================\n");

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

{

printf("%d\t\t\t",given[i].p);

printf("%d\t\t",given[i].bt);

printf("%d\t\t",given[i].at);

printf("%d\t\t",given[i].pr);

printf("%d\t\t",given[i].wt);

printf("%d\t\t\n",given[i].tt);

}

printf("============================================================================================");

}

OUTPUT :

