OPERATING SYSTEM ASSIGNMENT

NAME: PADAGALA SANJANA SECTION:K18JE ROLL NO: B51 GROUP: B

CPU schedules N processes which arrive at different time intervals and each process is allocated the CPU for a specific user input time unit, processes are scheduled using a preemptive round robin scheduling algorithm. Each process must be assigned a numerical priority, with a higher number indicating a higher relative priority. In addition to the processes one task has priority 0. The length of a time quantum is T units, where T is the custom time considered as time quantum for processing. If a process is preempted by a higher-priority process, the preempted process is placed at the end of the queue. Design a scheduler so that the task with priority 0 does not starve for resources and gets the CPU at some time unit to execute. Also compute waiting time, turn around.

* #include<stdlib.h>
* #include<string.h>
* #include<stdio.h>
* using namespace std;
* int main()
* {
* char a[10][5],temp[5];
* int m,l,priorityTime[10],waitingTime[10],tAt=0,pr[10],temp1,n;
* float average\_waiting\_time;
* printf("enter no of processes:");
* scanf("%d",&n);
* for(m=0;m<n;m++)
* {
* printf("enter process%d name:",m+1);
* scanf("%s",&a[m]);
* printf("enter process time:");
* scanf("%d",&priorityTime[m]);
* printf("enter priority:");
* scanf("%d",&pr[m]);
* }
* for(m=0;m<n-1;m++)
* {
* for(l=m+1;l<n;l++)
* {
* if(pr[m]>pr[l])
* {
* temp1=pr[m];
* pr[m]=pr[l];
* pr[l]=temp1;
* temp1=priorityTime[m];
* priorityTime[m]=priorityTime[l];
* priorityTime[l]=temp1;
* strcpy(temp,a[m]);
* strcpy(a[m],a[l]);
* strcpy(a[l],temp);
* }
* }
* }
* waitingTime[0]=0;
* for(m=1;m<n;m++)
* {
* waitingTime[m]=waitingTime[m-1] + priorityTime[m-1];
* tAt=tAt+waitingTime[m];
* }
* average\_waiting\_time=(float)tAt/n;
* printf("p\_name\t p\_time\t priority\t w\_time\n");
* for(m=0;m<n;m++)
* {
* printf(" %s\t %d\t %d\t %d\n" ,a[m],priorityTime[m],pr[m],waitingTime[m]);
* }
* printf("Turn Around Time =%d\n avg waiting time=%f",tAt,average\_waiting\_time);
* }

