

Program - Multi-level Queue Scheduling

Write a C program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue.

Code:

```
#include<stdio.h>

typedef struct Process{
    int id, at, bt, type, ct, tat, wt, remt;
}Process;

int ttt=0, twt=0;

void sortbyAT(Process p[], int n){
    for(int i=0;i<n-1;i++){
        for(int j=i+1;j<n;j++){
            if(p[i].at>p[j].at){
                Process temp=p[i];
                p[i]=p[j];
                p[j]=temp;
            }
        }
    }
}

void fcfs(Process p[], int n, int start){
    int curr=start;

    for(int i=0;i<n;i++){
        if(p[i].at>curr)
            curr=p[i].at;
        p[i].ct=p[i].bt+curr;
        p[i].tat=p[i].ct-p[i].at;
        p[i].wt=p[i].tat-p[i].bt;
        curr=p[i].ct;
    }
}
```

```

void printProcess(Process p[], int n, char* type){
    printf("%s Process:\n", type);
    printf("Process\tAT\tBT\tCT\tTAT\tWT\n");
    for(int i=0;i<n;i++){
        printf("P%d\t%d\t%d\t%d\t%d\t%d\n", p[i].id, p[i].at, p[i].bt, p[i].ct, p[i].tat, p[i].wt);
        ttt+=p[i].tat;
        twt+=p[i].wt;
    }
}

```

```

void roundrobin(Process p[], int n, int tq, int start){
    int comp=0, curr=start;
    int queue[20], visited[n], front=0, rear=0;

    for(int i=0;i<n;i++){
        visited[i]=0;
        p[i].remt=p[i].bt;
    }

    for(int i=0;i<n;i++){
        if(p[i].at==0){
            visited[i]=1;
            queue[rear++]=i;
        }
    }
}

```

```

while(comp<n){
    if(front==rear){
        curr++;
        for(int i=0;i<n;i++){
            if(p[i].at<=curr && !visited[i]){
                visited[i]=1;
                queue[rear++]=i;
            }
        }
        continue;
    }
}

```

```

int idx=queue[front++];
int exec_t=(p[idx].remt>tq)?tq:p[idx].remt;
curr+=exec_t;
p[idx].remt-=exec_t;
for(int i=0;i<n;i++){
    if(p[i].at<=curr && !visited[i]){
        visited[i]=1;
        queue[rear++]=i;
    }
}
if(p[idx].remt==0){
    comp++;
    p[idx].ct=curr;
    p[idx].tat=p[idx].ct-p[idx].at;
    p[idx].wt=p[idx].tat-p[idx].bt;
    ttt+=p[idx].tat;
    twt+=p[idx].wt;
}
else{
    queue[rear++]=idx;
}
}
}

int main(){
    int n, sysc=0, userc=0;
    printf("Enter the number of processes:");
    scanf("%d", &n);
    Process p[n], sysp[n], userp[n];
    printf("Enter the AT, BT and type(0-system, 1-user) for each process:\n");
    for(int i=0;i<n;i++){
        p[i].id=i+1;
        printf("Process %d:", i+1);
        scanf("%d %d %d", &p[i].at, &p[i].bt, &p[i].type);
        if(p[i].type==0){
            sysp[sysc++]=p[i];
        }
        else
            userp[userc++]=p[i];
    }
}

```

```

int c, tq;
printf("Enter:\n1. Both FCFS\n2. System is FCFS and user is RR\n3. System is RR and User
is FCFS\n");
scanf("%d", &c);

if(c==1){
    sortbyAT(syssp, sysc);
    sortbyAT(userp, userc);

    fcfs(syssp, sysc, 0);
    int lct=(sysc>0)?syssp[sysc-1].ct:0;
    fcfs(userp, userc, lct);
}

else if(c==2){
    printf("Enter the time quantum:");
    scanf("%d", &tq);
    sortbyAT(syssp, sysc);
    fcfs(syssp, sysc, 0);
    int lct=(sysc>0)?syssp[sysc-1].ct:0;
    roundrobin(userp, userc, tq,lct);
}
else{
    printf("Enter the time quantum:");
    scanf("%d", &tq);
    roundrobin(syssp, sysc, tq,0);
    int lct=(sysc>0)?syssp[sysc-1].ct:0;
    fcfs(userp, userc, lct);
}

printProcess(syssp, sysc, "System");
printProcess(userp, userc, "User");
printf("\nAverage TAT: %.2f units\n", (float)ttt/n);
printf("Average WT: %.2f units\n", (float)twt/n);
}

```

Output:

```

Enter the number of processes:4
Enter the AT, BT and type(0-system, 1-user) for each process:
Process 1:0 4 0
Process 2:0 3 0
Process 3:0 8 1
Process 4:10 5 0
System Process:
Process AT      BT      CT      TAT      WT
P1      0       4       4       4       0
P2      0       3       7       7       4
P4     10      5      15      5       0
User Process:
Process AT      BT      CT      TAT      WT
P3      0       8      23      23      15

Average TAT: 9.75 units
Average WT: 4.75 units
Process returned 0 (0x0)   execution time : 69.761 s
Press any key to continue.

```

```

Enter the number of processes:4
Enter the AT, BT and type(0-system, 1-user) for each process:
Process 1:0 4 0
Process 2:0 3 0
Process 3:0 8 1
Process 4:10 5 0
Enter:
1. Both FCFS
2. System is FCFS and user is RR
3. System is RR and User is FCFS
3
Enter the time quantum:2
System Process:
Process AT      BT      CT      TAT      WT
P1      0       4       6       6       2
P2      0       3       7       7       4
P4     10      5      15      5       0
User Process:
Process AT      BT      CT      TAT      WT
P3      0       8      23      23      15

Average TAT: 14.75 units
Average WT: 6.75 units

Process returned 0 (0x0)   execution time : 31.033 s
Press any key to continue.

```