Week -2

Write a C program to simulate a 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.

#include <stdio.h>

#include <stdlib.h>

struct process{

int id;

int burst\_time;

int is\_system; // 1 for system process, 0 for user process

};

typedef struct process Process;

void main(){

int n, sys\_count=0,user\_count=0;

printf("Enter the Number of Processes: ");

scanf("%d", &n);

if (n<=0) {

printf("No Processes to Schedule.\n");

return;

}

Process \*system\_queue[n], \*user\_queue[n];

for (int i = 0; i < n; i++) {

int type, burst;

printf("\nEnter Process Type (System: 1, User: 0): ");

scanf("%d", &type);

printf("Enter Burst Time: ");

scanf("%d", &burst);

Process \*p=(Process \*)malloc(sizeof(Process));

p->id=i + 1;

p->burst\_time=burst;

p->is\_system=type;

if(type==1){

system\_queue[sys\_count++]=p;

}else{

user\_queue[user\_count++]=p;

}

}

Process \*processes[n];

int index=0;

for (int i=0;i<sys\_count;i++){

processes[index++]=system\_queue[i];

}

for (int i = 0; i < user\_count; i++) {

processes[index++] = user\_queue[i];

}

calculate\_scheduling(processes, n);

for (int i = 0; i < sys\_count; i++) {

free(system\_queue[i]);

}

for (int i = 0; i < user\_count; i++) {

free(user\_queue[i]);

}

}

void calculate\_scheduling(Process \*processes[], int n) {

int wt[n], tat[n];

float total\_wt = 0, total\_tat = 0;

wt[0] = 0;

tat[0] = processes[0]->burst\_time;

total\_wt = wt[0];

total\_tat = tat[0];

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

wt[i] = wt[i - 1] + processes[i - 1]->burst\_time;

tat[i] = wt[i] + processes[i]->burst\_time;

total\_wt += wt[i];

total\_tat += tat[i];

}

printf("\nProcess\t Type\t Burst Time\t Waiting Time\t Turnaround Time\n");

for (int i = 0; i < n; i++) {

printf("%d\t %s\t %d\t\t %d\t\t %d\n",

processes[i]->id,

processes[i]->is\_system ? "System" : "User",

processes[i]->burst\_time,

wt[i],

tat[i]);

}

printf("\nAverage Waiting Time = %.2f", total\_wt / n);

printf("\nAverage Turnaround Time = %.2f\n", total\_tat / n);

}

