FCFS

#include<stdio.h>

#include<conio.h>

#include<curses.h>

int pn[10];

int burst[10],start[10],finish[10],tat[10],wt[10],s,i,n,temp,t;

int totwt=0,tottat=0;

int arr[10];

void process\_sort()

{

int j,i,key1,key2,key3;

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

{

key1=arr[j];

key2=pn[j];

key3=burst[j];

i=j-1;

while(i>=0 && arr[i]>key1)

{

arr[i+1]=arr[i];

pn[i+1]=pn[i];

burst[i+1]=burst[i];

i--;

}

arr[i+1]=key1;

pn[i+1]=key2;

burst[i+1]=key3;

}

}

void main()

{

printf("Enter the number of processes: ");

scanf("%d",&n);

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

{

printf("Enter the Process Name, Arrival Time & Burst Time:");

scanf("%d%d%d",&pn[i],&arr[i],&burst[i]);

}

process\_sort();

printf("Process Name\tArrival Time\tBurst Time\n");

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

{

printf("%5d\t%10d\t%10d",pn[i],arr[i],burst[i]);

printf("\n");

}

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

{

if(i==0)

{

start[i]=arr[i];

wt[i]=start[i]-arr[i];

finish[i]=start[i]+burst[i];

tat[i]=finish[i]-arr[i];

}

else

{

if(arr[i]>finish[i-1])

start[i]=arr[i];

else

start[i]=finish[i-1];

wt[i]=start[i]-arr[i];

finish[i]=start[i]+burst[i];

tat[i]=finish[i]-arr[i];

}

}

printf("PName Arrtime Bursttime Start WT TAT Finish");

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

{

printf("\n%d\t%4d\t\t%4d\t%4d\t%4d\t%4d\t%4d",pn[i],arr[i],burst[i],start[i],wt[i],tat[i],finish[i]);

totwt+=wt[i];

tottat+=tat[i];

}

printf("\nAverage Waiting time:%f",(float)totwt/n);

printf("\nAverage Turn Around Time:%f",(float)tottat/n);

}

**SJF**

#include<stdio.h>

#include<conio.h>

#include<string.h>

void main()

{

int et[20],at[10],n,i,j,temp,st[10],ft[10],wt[10],ta[10];

int totwt=0,totta=0;

float awt,ata;

char pn[10][10],t[10];

printf("Enter the number of process:");

scanf("%d",&n);

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

{

printf("Enter process name, arrival time & execution time:");

//flushall();

scanf("%s%d%d",pn[i],&at[i],&et[i]);

}

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

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

{

if(et[i]<et[j])

{

temp=at[i];

at[i]=at[j];

at[j]=temp;

temp=et[i];

et[i]=et[j];

et[j]=temp;

strcpy(t,pn[i]);

strcpy(pn[i],pn[j]);

strcpy(pn[j],t);

}

}

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

{

if(i==0)

st[i]=at[i];

else

st[i]=ft[i-1];

wt[i]=st[i]-at[i];

ft[i]=st[i]+et[i];

ta[i]=ft[i]-at[i];

totwt+=wt[i];

totta+=ta[i];

}

awt=(float)totwt/n;

ata=(float)totta/n;

printf("Pname\tarrivaltime\texecutiontime\twaitingtime\ttatime");

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

printf("\n%s\t%5d\t\t%5d\t\t%5d\t\t%5d",pn[i],at[i],et[i],wt[i],ta[i]);

printf("\nAverage waiting time is:%f",awt);

printf("\nAverage turnaroundtime is:%f",ata);

}

**Priority**

#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<curses.h>

void main()

{

int et[20],at[10],n,i,j,temp,p[10],st[10],ft[10],wt[10],ta[10];

int totwt=0,totta=0;

float awt,ata;

char pn[10][10],t[10];

printf("Enter the number of process:");

scanf("%d",&n);

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

{

printf("Enter process name,arrivaltime,execution time & priority:");

//flushall();

scanf("%s%d%d%d",pn[i],&at[i],&et[i],&p[i]);

}

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

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

{

if(at[i]<at[j])

{

temp=p[i];

p[i]=p[j];

p[j]=temp;

temp=at[i];

at[i]=at[j];

at[j]=temp;

temp=et[i];

et[i]=et[j];

et[j]=temp;

strcpy(t,pn[i]);

strcpy(pn[i],pn[j]);

strcpy(pn[j],t);

}

}

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

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

{

if(p[i]<p[j])

{

temp=p[i];

p[i]=p[j];

p[j]=temp;

temp=at[i];

at[i]=at[j];

at[j]=temp;

temp=et[i];

et[i]=et[j];

et[j]=temp;

strcpy(t,pn[i]);

strcpy(pn[i],pn[j]);

strcpy(pn[j],t);

}

}

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

{

if(i==0)

{

st[i]=at[i];

wt[i]=st[i]-at[i];

ft[i]=st[i]+et[i];

ta[i]=ft[i]-at[i];

}

else

{

st[i]=ft[i-1];

wt[i]=st[i]-at[i];

ft[i]=st[i]+et[i];

ta[i]=ft[i]-at[i];

}

totwt+=wt[i];

totta+=ta[i];

}

awt=(float)totwt/n;

ata=(float)totta/n;

printf("Pname\tarrivaltime\texecutiontime\tpriority\twaitingtime\ttatime");

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

printf("\n%s\t%5d\t\t%5d\t\t%5d\t\t%5d\t\t%5d",pn[i],at[i],et[i],p[i],wt[i],ta[i]);

printf("\nAverage waiting time is:%f",awt);

printf("\nAverage turnaroundtime is:%f",ata);

}

**ROUND ROBIN**

#include<stdio.h>

int main()

{

int i, limit, total = 0, x, counter = 0, time\_quantum;

int wait\_time = 0, turnaround\_time = 0, arrival\_time[10], burst\_time[10], temp[10];

float average\_wait\_time, average\_turnaround\_time;

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

scanf("%d", &limit);

x = limit;

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

{

printf("Enter Details of Process[%d]: ", i + 1);

printf("Arrival Time:\t");

scanf("%d", &arrival\_time[i]);

printf("Burst Time:\t");

scanf("%d", &burst\_time[i]);

temp[i] = burst\_time[i];

}

printf("Enter Time Quantum:\t");

scanf("%d", &time\_quantum);

printf("Process ID\t\tBurst Time\t Turnaround Time\t Waiting Time");

for(total = 0, i = 0; x != 0;)

{

if(temp[i] <= time\_quantum && temp[i] > 0)

{

total = total + temp[i];

temp[i] = 0;

counter = 1;

}

else if(temp[i] > 0)

{

temp[i] = temp[i] - time\_quantum;

total = total + time\_quantum;

}

if(temp[i] == 0 && counter == 1)

{

x--;

printf("\nProcess[%d]\t\t%d\t\t %d\t\t\t %d", i + 1, burst\_time[i], total - arrival\_time[i], total - arrival\_time[i] - burst\_time[i]);

wait\_time = wait\_time + total - arrival\_time[i] - burst\_time[i];

turnaround\_time = turnaround\_time + total - arrival\_time[i];

counter = 0;

}

if(i == limit - 1)

{

i = 0;

}

else if(arrival\_time[i + 1] <= total)

{

i++;

}

else

{

i = 0;

}

}

average\_wait\_time = wait\_time \* 1.0 / limit;

average\_turnaround\_time = turnaround\_time \* 1.0 / limit;

printf("\nAverage Waiting Time:\t%f", average\_wait\_time);

printf("\nAvg Turnaround Time:\t%f\n", average\_turnaround\_time);

return 0;

}

**Multi Level Queue Scheduling**

#include<stdio.h>

int main() {

int p[20], bt[20], su[20], wt[20], tat[20], i, k, n, temp;

float wtavg, tatavg;

printf("Enter the number of processes:");

scanf("%d", & n);

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

p[i] = i;

printf("Enter the Burst Time of Process %d:", i);

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

printf("System/User Process (0/1) ?");

scanf("%d", & su[i]);

}

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

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

if (su[i] > su[k]) {

temp = p[i];

p[i] = p[k];

p[k] = temp;

temp = bt[i];

bt[i] = bt[k];

bt[k] = temp;

temp = su[i];

su[i] = su[k];

su[k] = temp;

}

wtavg = wt[0] = 0;

tatavg = tat[0] = bt[0];

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

wt[i] = wt[i - 1] + bt[i - 1];

tat[i] = tat[i - 1] + bt[i];

wtavg = wtavg + wt[i];

tatavg = tatavg + tat[i];

}

printf("PROCESS\t\t SYSTEM/USER PROCESS \tBURST TIME\tWAITING TIME\tTURNAROUND TIME");

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

printf("\n%d \t\t %d \t\t %d \t\t %d \t\t %d ", p[i], su[i], bt[i], wt[i], tat[i]);

printf("\nAverage Waiting Time is --- %f", wtavg / n);

printf("\nAverage Turnaround Time is --- %f", tatavg / n);

return 0;

}