Name: Aditya Matale

Roll. No: 25

PRN: 12110037

Class: AIDS – TY (B) Batch\_01

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OS LAB**

**Scheduling Algorithms**

#] **Menu Driven Program:**

1] **First-Come, First-Served (FCFS)**

2] **Shortest Job First (SJF)**

3] **Priority Scheduling**

4] **Round Robin (RR)**

**CODE:**

**#include<stdio.h>**

**void FCFS()**

**{**

**int pid[15];**

**int bt[15];**

**int n;**

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

**scanf("%d",&n);**

**printf("Enter process id of all the processes: ");**

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

**{**

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

**}**

**printf("Enter burst time of all the processes: ");**

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

**{**

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

**}**

**int i, wt[n];**

**wt[0]=0;**

**//for calculating waiting time of each process**

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

**{**

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

**}**

**printf("Process ID Burst Time Waiting Time TurnAround Time\n");**

**float twt=0.0;**

**float tat= 0.0;**

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

**{**

**printf("%d\t\t", pid[i]);**

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

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

**//calculating and printing turnaround time of each process**

**printf("%d\t\t", bt[i]+wt[i]);**

**printf("\n");**

**//for calculating total waiting time**

**twt += wt[i];**

**//for calculating total turnaround time**

**tat += (wt[i]+bt[i]);**

**}**

**float att,awt;**

**//for calculating average waiting time**

**awt = twt/n;**

**//for calculating average turnaround time**

**att = tat/n;**

**printf("Avg. waiting time= %f\n",awt);**

**printf("Avg. turnaround time= %f",att);**

**}**

**void SJF()**

**{**

**int bt[20],p[20],wt[20],tat[20],i,j,n,total=0,totalT=0,pos,temp;**

**float avg\_wt,avg\_tat;**

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

**scanf("%d",&n);**

**printf("\nEnter Burst Time:\n");**

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

**{**

**printf("p%d:",i+1);**

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

**p[i]=i+1;**

**}**

**//sorting of burst times**

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

**{**

**pos=i;**

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

**{**

**if(bt[j]<bt[pos])**

**pos=j;**

**}**

**temp=bt[i];**

**bt[i]=bt[pos];**

**bt[pos]=temp;**

**temp=p[i];**

**p[i]=p[pos];**

**p[pos]=temp;**

**}**

**wt[0]=0;**

**//finding the waiting time of all the processes**

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

**{**

**wt[i]=0;**

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

**//individual WT by adding BT of all previous completed processes**

**wt[i]+=bt[j];**

**//total waiting time**

**total+=wt[i];**

**}**

**//average waiting time**

**avg\_wt=(float)total/n;**

**printf("\nProcess\t Burst Time \tWaiting Time\tTurnaround Time");**

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

**{**

**//turnaround time of individual processes**

**tat[i]=bt[i]+wt[i];**

**//total turnaround time**

**totalT+=tat[i];**

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

**}**

**//average turnaround time**

**avg\_tat=(float)totalT/n;**

**printf("\n\nAverage Waiting Time=%f",avg\_wt);**

**printf("\nAverage Turnaround Time=%f",avg\_tat);**

**}**

**void RR()**

**{**

**//Input no of processed**

**int n;**

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

**scanf("%d", &n);**

**int wait\_time = 0, ta\_time = 0, arr\_time[n], burst\_time[n], temp\_burst\_time[n];**

**int x = n;**

**//Input details of processes**

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

**{**

**printf("Enter Details of Process %d \n", i + 1);**

**printf("Arrival Time: ");**

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

**printf("Burst Time: ");**

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

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

**}**

**//Input time slot**

**int time\_slot;**

**printf("Enter Time Slot:");**

**scanf("%d", &time\_slot);**

**//Total indicates total time**

**//counter indicates which process is executed**

**int total = 0, counter = 0,i;**

**printf("Process ID Burst Time Turnaround Time Waiting Time\n");**

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

**{**

**// define the conditions**

**if(temp\_burst\_time[i] <= time\_slot && temp\_burst\_time[i] > 0)**

**{**

**total = total + temp\_burst\_time[i];**

**temp\_burst\_time[i] = 0;**

**counter=1;**

**}**

**else if(temp\_burst\_time[i] > 0)**

**{**

**temp\_burst\_time[i] = temp\_burst\_time[i] - time\_slot;**

**total += time\_slot;**

**}**

**if(temp\_burst\_time[i]==0 && counter==1)**

**{**

**x--; //decrement the process no.**

**printf("\nProcess No %d \t\t %d\t\t\t %d\t\t\t %d\t\t", i+1, burst\_time[i],**

**total-arr\_time[i], total-arr\_time[i]-burst\_time[i]);**

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

**ta\_time += total -arr\_time[i];**

**counter =0;**

**}**

**if(i==n-1)**

**{**

**i=0;**

**}**

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

**{**

**i++;**

**}**

**else**

**{**

**i=0;**

**}**

**}**

**float average\_wait\_time = wait\_time \* 1.0 / n;**

**float average\_turnaround\_time = ta\_time \* 1.0 / n;**

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

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

**//  return 0;**

**}**

**void Priority(){**

**int n;**

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

**scanf("%d", &n);**

**int processes[n];**

**int burst\_time[n];**

**int priority[n];**

**// Input process details**

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

**printf("Enter the details for Process %d:\n", i + 1);**

**processes[i] = i + 1;**

**printf("Burst Time: ");**

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

**printf("Priority: ");**

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

**}**

**// Sort the processes based on priority (higher priority first) using Bubble Sort**

**for (int i = 0; i < n - 1; i++) {**

**for (int j = 0; j < n - i - 1; j++) {**

**if (priority[j] > priority[j + 1]) {**

**// Swap the processes**

**int temp = processes[j];**

**processes[j] = processes[j + 1];**

**processes[j + 1] = temp;**

**temp = burst\_time[j];**

**burst\_time[j] = burst\_time[j + 1];**

**burst\_time[j + 1] = temp;**

**temp = priority[j];**

**priority[j] = priority[j + 1];**

**priority[j + 1] = temp;**

**}**

**}**

**}**

**// Calculate the waiting time and turnaround time**

**int waiting\_time[n];**

**int turnaround\_time[n];**

**waiting\_time[0] = 0;**

**turnaround\_time[0] = burst\_time[0];**

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

**waiting\_time[i] = turnaround\_time[i - 1];**

**turnaround\_time[i] = waiting\_time[i] + burst\_time[i];**

**}**

**// Calculate the average waiting time and average turnaround time**

**float avg\_waiting\_time = 0;**

**float avg\_turnaround\_time = 0;**

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

**avg\_waiting\_time += waiting\_time[i];**

**avg\_turnaround\_time += turnaround\_time[i];**

**}**

**avg\_waiting\_time /= n;**

**avg\_turnaround\_time /= n;**

**// Display the results**

**printf("Process\tBurst Time\tPriority\tWaiting Time\tTurnaround Time\n");**

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

**printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", processes[i], burst\_time[i], priority[i], waiting\_time[i], turnaround\_time[i]);**

**}**

**printf("Average Waiting Time: %.2f\n", avg\_waiting\_time);**

**printf("Average Turnaround Time: %.2f\n", avg\_turnaround\_time);**

**}**

**int main(){**

**while (1)**

**{**

**printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n1] FCFS\n2] SJF\n3] RR\n4] Priority\n5] EXIT\nScheduling Algorithm?: ");**

**int choice;**

**scanf("%d",&choice);**

**if (choice==5){**

**break;**

**}**

**switch (choice)**

**{**

**case 1:**

**FCFS();**

**break;**

**case 2:**

**SJF();**

**break;**

**case 3:**

**RR();**

**break;**

**case 4:**

**Priority();**

**break;**

**default:**

**printf("invalid choice");**

**break;**

**}**

**}**

**return 0;**

**}**

**OUTPUT:**









