**4ITRC2 Operating System Lab**

**Lab Assignment 5**

**Submitted by:** **Riya Rawat, 23I4060**

Aim :To create C programs for the different scheduling algorithms.

To Perform: Create and execute C programs for following CPU Scheduling Algorithms: 1. First Come First Serve (FCFS) 2. Shortest Job First (SJF) 3. Round Robin Scheduling

# 1. First Come First Serve (FCFS)

C Code:

#include <stdio.h>  
  
int main() {  
 int n, i;  
 printf("Enter number of processes: ");  
 scanf("%d", &n);  
  
 int bt[n], wt[n], tat[n];  
 printf("Enter burst times:\n");  
 for (i = 0; i < n; i++) {  
 printf("P%d: ", i + 1);  
 scanf("%d", &bt[i]);  
 }  
  
 wt[0] = 0;  
 for (i = 1; i < n; i++)  
 wt[i] = wt[i - 1] + bt[i - 1];  
  
 printf("\nProcess\tBT\tWT\tTAT\n");  
 for (i = 0; i < n; i++) {  
 tat[i] = bt[i] + wt[i];  
 printf("P%d\t%d\t%d\t%d\n", i + 1, bt[i], wt[i], tat[i]);  
 }  
  
 return 0;  
}

Sample Output:

Enter number of processes: 3  
Enter burst times:  
P1: 5  
P2: 8  
P3: 6  
  
Process BT WT TAT  
P1 5 0 5  
P2 8 5 13  
P3 6 13 19

# 2. Shortest Job First (SJF)

C Code:#include <stdio.h>  
  
int main() {  
 int n, i, j;  
 printf("Enter number of processes: ");  
 scanf("%d", &n);  
  
 int bt[n], p[n], wt[n], tat[n];  
 for (i = 0; i < n; i++) {  
 p[i] = i;  
 printf("Enter burst time for P%d: ", i + 1);  
 scanf("%d", &bt[i]);  
 }  
  
 for (i = 0; i < n - 1; i++)  
 for (j = i + 1; j < n; j++)  
 if (bt[i] > bt[j]) {  
 int t = bt[i]; bt[i] = bt[j]; bt[j] = t;  
 t = p[i]; p[i] = p[j]; p[j] = t;  
 }  
  
 wt[0] = 0;  
 for (i = 1; i < n; i++)  
 wt[i] = wt[i - 1] + bt[i - 1];  
  
 printf("\nProcess\tBT\tWT\tTAT\n");  
 for (i = 0; i < n; i++) {  
 tat[i] = wt[i] + bt[i];  
 printf("P%d\t%d\t%d\t%d\n", p[i] + 1, bt[i], wt[i], tat[i]);  
 }  
  
 return 0;  
}

Sample Output:

Enter number of processes: 3  
Enter burst time for P1: 6  
Enter burst time for P2: 4  
Enter burst time for P3: 2  
  
Process BT WT TAT  
P3 2 0 2  
P2 4 2 6  
P1 6 6 12

**3. Round Robin Scheduling**

C Code:  
#include <stdio.h>  
  
int main() {  
 int i, j, n, time, remain, tq;  
 printf("Enter number of processes: ");  
 scanf("%d", &n);  
  
 int bt[n], rt[n], wt[n], tat[n];  
 remain = n;  
  
 for (i = 0; i < n; i++) {  
 printf("Enter Burst Time for P%d: ", i + 1);  
 scanf("%d", &bt[i]);  
 rt[i] = bt[i];  
 }  
  
 printf("Enter Time Quantum: ");  
 scanf("%d", &tq);  
  
 int t = 0;  
 while (remain > 0) {  
 for (i = 0; i < n; i++) {  
 if (rt[i] > 0) {  
 if (rt[i] > tq) {  
 t += tq;  
 rt[i] -= tq;  
 } else {  
 t += rt[i];  
 wt[i] = t - bt[i];  
 rt[i] = 0;  
 remain--;  
 }  
 }  
 }  
 }  
  
 printf("\nProcess\tBT\tWT\tTAT\n");  
 for (i = 0; i < n; i++) {  
 tat[i] = bt[i] + wt[i];  
 printf("P%d\t%d\t%d\t%d\n", i + 1, bt[i], wt[i], tat[i]);  
 }  
 return 0;  
}

Sample Output:

Enter number of processes: 3  
Enter Burst Time for P1: 5  
Enter Burst Time for P2: 4  
Enter Burst Time for P3: 6  
Enter Time Quantum: 3  
  
Process BT WT TAT  
P1 5 9 14  
P2 4 6 10  
P3 6 8 14