Assignment 5 - CPU Scheduling Algorithms

4ITRC2 Operating System Lab  
  
Shweta Ahirwar|| Roll No. 23I4073

#include <stdio.h>  
  
**1. FCFS Scheduling Algorithm**

void fcfs() {  
 int n, i;  
 printf("\n--- First Come First Serve (FCFS) Scheduling ---\n");  
 printf("Enter number of processes: ");  
 scanf("%d", &n);  
 int bt[n], wt[n], tat[n];  
  
 printf("Enter burst time for each process:\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] = bt[i-1] + wt[i-1];  
 }  
  
 for(i = 0; i < n; i++) {  
 tat[i] = bt[i] + wt[i];  
 }  
  
 printf("\nProcess\tBT\tWT\tTAT\n");  
 for(i = 0; i < n; i++) {  
 printf("P%d\t%d\t%d\t%d\n", i+1, bt[i], wt[i], tat[i]);  
 }  
}  
  
 **2. SJF Scheduling Algorithm**

void sjf() {  
 int n, i, j;  
 printf("\n--- Shortest Job First (SJF) Scheduling ---\n");  
 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+1;  
 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 temp = bt[i]; bt[i] = bt[j]; bt[j] = temp;  
 temp = p[i]; p[i] = p[j]; p[j] = temp;  
 }  
 }  
 }  
  
 wt[0] = 0;  
 for(i = 1; i < n; i++) {  
 wt[i] = wt[i-1] + bt[i-1];  
 }  
  
 for(i = 0; i < n; i++) {  
 tat[i] = wt[i] + bt[i];  
 }  
  
 printf("\nProcess\tBT\tWT\tTAT\n");  
 for(i = 0; i < n; i++) {  
 printf("P%d\t%d\t%d\t%d\n", p[i], bt[i], wt[i], tat[i]);  
 }  
}

**3. Round Robin Scheduling Algorithm**  
void round\_robin() {  
 int n, i, tq, time = 0;  
 printf("\n--- Round Robin Scheduling ---\n");  
 printf("Enter number of processes: ");  
 scanf("%d", &n);  
 int bt[n], rem\_bt[n], wt[n], tat[n];  
  
 printf("Enter burst time for each process:\n");  
 for(i = 0; i < n; i++) {  
 printf("P%d: ", i+1);  
 scanf("%d", &bt[i]);  
 rem\_bt[i] = bt[i];  
 wt[i] = 0;  
 }  
  
 printf("Enter time quantum: ");  
 scanf("%d", &tq);  
  
 while(1) {  
 int done = 1;  
 for(i = 0; i < n; i++) {  
 if(rem\_bt[i] > 0) {  
 done = 0;  
 if(rem\_bt[i] > tq) {  
 time += tq;  
 rem\_bt[i] -= tq;  
 } else {  
 time += rem\_bt[i];  
 wt[i] = time - bt[i];  
 rem\_bt[i] = 0;  
 }  
 }  
 }  
 if(done == 1)  
 break;  
 }  
  
 for(i = 0; i < n; i++) {  
 tat[i] = bt[i] + wt[i];  
 }  
  
 printf("\nProcess\tBT\tWT\tTAT\n");  
 for(i = 0; i < n; i++) {  
 printf("P%d\t%d\t%d\t%d\n", i+1, bt[i], wt[i], tat[i]);  
 }  
}  
  
int main() {  
 int choice;  
 while(1) {  
 printf("\n\n--- CPU Scheduling Algorithms Menu ---\n");  
 printf("1. First Come First Serve (FCFS)\n");  
 printf("2. Shortest Job First (SJF)\n");  
 printf("3. Round Robin\n");  
 printf("4. Exit\n");  
 printf("Enter your choice: ");  
 scanf("%d", &choice);  
  
 switch(choice) {  
 case 1: fcfs(); break;  
 case 2: sjf(); break;  
 case 3: round\_robin(); break;  
 case 4: return 0;  
 default: printf("Invalid choice!\n");  
 }  
 }  
 return 0;  
}