**Lab Assignment 5**

**1. First Come First Serve (FCFS)**

#include <stdio.h>

int main() {

int n, i;

int bt[20], wt[20], tat[20];

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

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

scanf("%d", &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] = wt[i-1] + bt[i-1];

}

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

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

avg\_wt += wt[i];

avg\_tat += tat[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]);

}

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

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

return 0;

}

**2. Shortest Job First (Non-preemptive)**

#include <stdio.h>

int main() {

int n, i, j, pos, temp;

int bt[20], p[20], wt[20], tat[20];

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

printf("Enter number of processes: ");

scanf("%d", &n);

printf("Enter burst time:\n");

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

printf("P[%d]: ", i+1);

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

p[i] = i + 1;

}

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;

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

wt[i] = 0;

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

wt[i] += bt[j];

avg\_wt += wt[i];

}

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

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

avg\_tat += tat[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]);

}

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

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

return 0;

}

**3. Round Robin Scheduling**

#include <stdio.h>

int main() {

int i, j, n, tq, time = 0;

int bt[20], rt[20], wt[20], tat[20];

printf("Enter number of processes: ");

scanf("%d", &n);

printf("Enter burst time for each process:\n");

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

printf("P[%d]: ", i+1);

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

rt[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(rt[i] > 0) {

done = 0;

if(rt[i] > tq) {

time += tq;

rt[i] -= tq;

} else {

time += rt[i];

wt[i] = time - bt[i];

rt[i] = 0;

}

}

}

if(done) break;

}

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

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

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

avg\_wt += wt[i];

avg\_tat += tat[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]);

}

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

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

return 0;

}