**OS Assignment 5**

**Operating System Lab**

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];

}

printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");

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

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

avg\_wt += wt[i];

avg\_tat += tat[i];

printf("P[%d]\t%d\t\t%d\t\t%d\n", i + 1, bt[i], wt[i], tat[i]);

}

avg\_wt /= n;

avg\_tat /= n;

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

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

return 0;

}

2. Shortest Job First (SJF)

#include <stdio.h>

int main() {

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

int i, j, pos, temp;

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

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]);

p[i] = i + 1;

}

// Sort by burst time

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];

}

printf("\nProcess\tBurst Time\tWaiting Time\tTurnaround Time\n");

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

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

avg\_tat += tat[i];

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

}

avg\_wt /= n;

avg\_tat /= n;

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

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

return 0;

}

3. Round Robin Scheduling

#include <stdio.h>

int main() {

int i, n, time, remain, flag = 0, tq;

int wt = 0, tat = 0;

int at[10], bt[10], rt[10];

printf("Enter Total Process: ");

scanf("%d", &n);

remain = n;

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

printf("Enter Arrival Time and Burst Time for Process P[%d]: ", i + 1);

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

rt[i] = bt[i];

}

printf("Enter Time Quantum: ");

scanf("%d", &tq);

printf("\nProcess\tTurnaround Time\tWaiting Time\n");

for (time = 0, i = 0; remain != 0;) {

if (rt[i] > 0 && at[i] <= time) {

if (rt[i] <= tq) {

time += rt[i];

rt[i] = 0;

flag = 1;

} else {

rt[i] -= tq;

time += tq;

}

if (rt[i] == 0 && flag == 1) {

remain--;

printf("P[%d]\t%d\t\t%d\n", i + 1, time - at[i], time - at[i] - bt[i]);

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

tat += time - at[i];

flag = 0;

}

}

i = (i + 1) % n;

}

printf("\nAverage Waiting Time = %.2f", (float)wt / n);

printf("\nAverage Turnaround Time = %.2f\n", (float)tat / n);

return 0;

}