**Operating System**

**(4ITRC2)**

**IT IV Semester**

*Submitted by*

**Raghav Agrawal**

**23I4056**

**IT ‘A’**

*Submitted to*

**Jasneet Kaur Mam**

Institute of Engineering and Technology

Devi Ahilya Vishwavidhyalaya, Indore (M.P.) India

**(**[www.iet.dauniv.ac.in](http://www.iet.dauniv.ac.in)**)**

**Session 2023-2027**

**Assignment V**

**First Come First Served (FCFS)**

*Program-*

#include <stdio.h>

int main() {

int n, i;

printf("Enter number of processes: ");

scanf("%d", &n);

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

wt[0] = 0;

printf("Enter burst times for %d processes:\n", n);

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

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

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

}

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", i+1, bt[i], wt[i], tat[i]);

}

return 0;

}

*Output-*

Enter number of processes: 3

Enter burst times for 3 processes:

P1: 5

P2: 8

P3: 6

Process BT WT TAT

P1 5 0 5

P2 8 5 13

P3 6 13 19

**Shortest Job first (SJF) –**

*Program-*

#include <stdio.h>

int main() {

int n, i, j;

printf("Enter number of processes: ");

scanf("%d", &n);

int pid[n], bt[n], wt[n], tat[n], temp;

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

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

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

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

pid[i] = i+1;

}

// Sorting burst time in ascending order

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

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

if(bt[i] > bt[j]) {

temp = bt[i]; bt[i] = bt[j]; bt[j] = temp;

temp = pid[i]; pid[i] = pid[j]; pid[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", pid[i], bt[i], wt[i], tat[i]);

}

return 0;

}

*Output-*

Enter number of processes: 3

Enter burst times:

P1: 8

P2: 4

P3: 2

Process BT WT TAT

P3 2 0 2

P2 4 2 6

P1 8 6 14

**Round Robin Scheduling**

*Program-*

#include <stdio.h>

int main() {

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

printf("Enter number of processes: ");

scanf("%d", &n);

int bt[n], rt[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]);

rt[i] = bt[i]; // remaining time

}

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;

}

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]);

}

return 0;

}

*Output-*

Enter number of processes: 3

Enter burst times:

P1: 10

P2: 5

P3: 8

Enter time quantum: 2

Process BT WT TAT

P1 10 13 23

P2 5 6 11

P3 8 12 20

***Thank You***