## Practical 7

Practical 7

# Aim : Write a program to calculate total waiting time and turn around time of n processes with FCFS cpu scheduling algorithm .

**Code:**

#include<stdio.h> void findWaitingTime( int processes[ ] , int n , int bt[ ], int wt[ ])

{

wt[ 0 ] = 0;

for ( int i = 1 ; i < n ; i++ ) wt[ i ]

= bt[ i - 1 ] + wt[ i - 1 ] ;

}

void findTurnAroundTime ( int processes[ ], int n,

int bt[ ] , int wt [ ] , int tat[ ] )

{

for ( int i = 0 ; i < n ; i++) tat[i] = bt[i] + wt[i];

}

void findavgTime(int processes[], int n, int bt[])

{

int wt[n], tat[n], total\_wt = 0, total\_tat = 0; findWaitingTime(processes, n, bt, wt); findTurnAroundTime(processes, n, bt, wt, tat); printf("Processes Burst time Waiting time Turn around time\n"); for (int i = 0; i < n; i++)

{

total\_wt = total\_wt + wt[i]; total\_tat = total\_tat + tat[i]; printf(" %d ", (i + 1));

printf(" %d ", bt[i]);

printf(" %d", wt[i]);

printf(" %d\n", tat[i]);

}

int s = (float)total\_wt / (float)n; int t = (float)total\_tat / (float)n; printf("Average waiting time = %d", s);

printf("\n");

printf("Average turn around time = %d ", t);

}

int main()

{

// process id's int processes[] = {1, 2, 3};

## Operating Systems Batch:AB20 Practical 7

int n = sizeof processes / sizeof processes[0]; int burst\_time[] = {10, 5, 8}; findavgTime(processes, n, burst\_time); return 0;

}

# Output: