Practical-7

1. Write C Program to implement FCFS Process 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] ;

}

// Function to calculate turn around time void findTurnAroundTime( int processes[], int n, int bt[], int wt[], int tat[])

{

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

}

//Function to calculate average time void findavgTime( int processes[], int n, int bt[])

{ int wt[n], tat[n], total\_wt = 0, total\_tat =

0; findWaitingTime(processes, n, bt, wt);

//Function to find turn around time for all processes findTurnAroundTime(processes, n, bt, wt, tat); printf("Processes Burst time Waiting time Turn around time\n");

// Calculate total waiting time and total turn around time 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] );

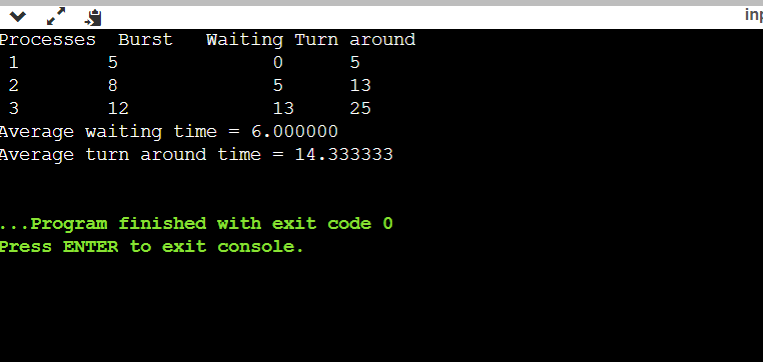
} int s=(float)total\_wt / (float)n; int t=(float)total\_tat / (float)n; printf("Average waiting time = %d",s);

printf("\n"); printf("Average turn around time = %d ",t); } int main() {

int processes[] = { 1, 2, 3}; int n = sizeof processes / sizeof

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

OUTPUT:



---------------------------------------------------------------------------------------------------------------------------------------

2.Write C Program to implement FIFO Page replacement algorithm.

CODE:

#include <stdio.h> int main() {

int referenceString[10], pageFaults = 0, m, n, s, pages, frames;

printf("\nEnter the number of Pages:\t"); scanf("%d",

&pages); printf("\nEnter reference string values:\n"); for( m = 0; m < pages; m++)

{ printf("Value No. [%d]:\t", m +

1); scanf("%d",

&referenceString[m]);

} printf("\n What are the total number of frames:\t");

{

scanf("%d", &frames);

} int temp[frames]; for(m = 0; m < frames; m++)

{ temp[m] = 1; } for(m = 0; m < pages; m++)

{ s = 0; for(n = 0; n < frames; n++)

{ if(referenceString[m] == temp[n])

{ s++; pageFaults--; } } pageFaults++; if((pageFaults <= frames) && (s == 0))

{ temp[m] = referenceString[m];

} else if(s == 0) { temp[(pageFaults - 1) % frames] = referenceString[m];

} printf("\n");

for(n = 0; n < frames; n++)

{

printf("%d\t", temp[n]);

}

} printf("\nTotal Page Faults:\t%d\n", pageFaults);

return 0; } OUTPUT:

