**Number of the Experiment :** 07

**Name of the Experiment :** Implementation of FCFS Scheduling Algorithm using C program.

**Date of Performance :**

**Date of Submission :**

**Aim and Objectives :** To learn about First Come First Serve scheduling algorithm and also know the implementation of it with a c program.

**Theory :** In the "First come first serve" scheduling algorithm, as the name suggests, the process which arrives first, gets executed first, or we can say that the process which requests the CPU first, gets the CPU allocated first.

* First Come First Serve, is just like FIFO (First in First out) Queue data structure, where the data element which is added to the queue first, is the one who leaves the queue first.
* This is used in [Batch Systems](https://www.studytonight.com/operating-system/types-of-os).
* It's easy to understand and implement programmatically, using a Queue data structure, where a new process enters through the tail of the queue, and the scheduler selects process from the head of the queue.
* A perfect real life example of FCFS scheduling is buying tickets at ticket counter.

**Source Code :**

#include<stdio.h>

int main()

{

int n,bt[20],wt[20],tat[20],avwt=0,avtat=0,i,j;

printf("Enter total number of processes(maximum 20):");

scanf("%d",&n);

printf("\nEnter Process Burst Time\n");

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

{

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

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

}

wt[0]=0; //waiting time for first process is 0

//calculating waiting time

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

{

wt[i]=0;

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

wt[i]+=bt[j];

}

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

//calculating turnaround time

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

{

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

avwt+=wt[i];

avtat+=tat[i];

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

}

avwt/=i;

avtat/=i;

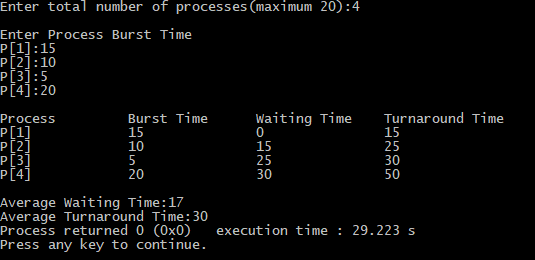
printf("\n\nAverage Waiting Time:%d",avwt);

printf("\nAverage Turnaround Time:%d",avtat);

return 0;

}

**Output :**



**Discussion :** FCFS is the simplest scheduling algorithm. FIFO simply queues processes in the order that they arrive in the ready queue. Hope this will be of great help in future work.