

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Santosh,Tangail-1902

LAB REPORT

Lab Report No : 07

Lab Report name : Implementation of FCFS Scheduling algorithm.

Course Title : Operating System Lab

Course Code : ICT-3110

Date of Performance :

Date of Submission :

Submitted by,

Student Name : Tanvir Ahmed

Student ID : IT-18043

Session : 2017-18

3rd Year 1st semester

Dept. of ICT

Submitted to,

Nazrul Islam

Assistant Professor

Dept. of ICT,

MBSTU.

Lab Report No. **07**

Lab Report Name: **Implementation of FCFS Scheduling algorithm .**

**Objectives:**

* What is FCFS Scheduling algorithm.

iiHow to implementation in C

**Theory:** First Come First Served (FCFS) is a **Non-Preemptive** scheduling algorithm.FIFO (First In First Out) strategy assigns priority to process in the order in which they request the processor. The process that requests the CPU first is allocated the CPU first. This is easily implemented with a FIFO queue for managing the tasks. As the process come in, they are put at the end of the queue. As the CPU finishes each task, it removes it from the start of the queue and heads on to the next task.

Turn Around Time = Completion Time - Arrival Time

Waiting Time = Turnaround time – Burst Time

Advantages of FCFS

Simple Easy

First come, First serv

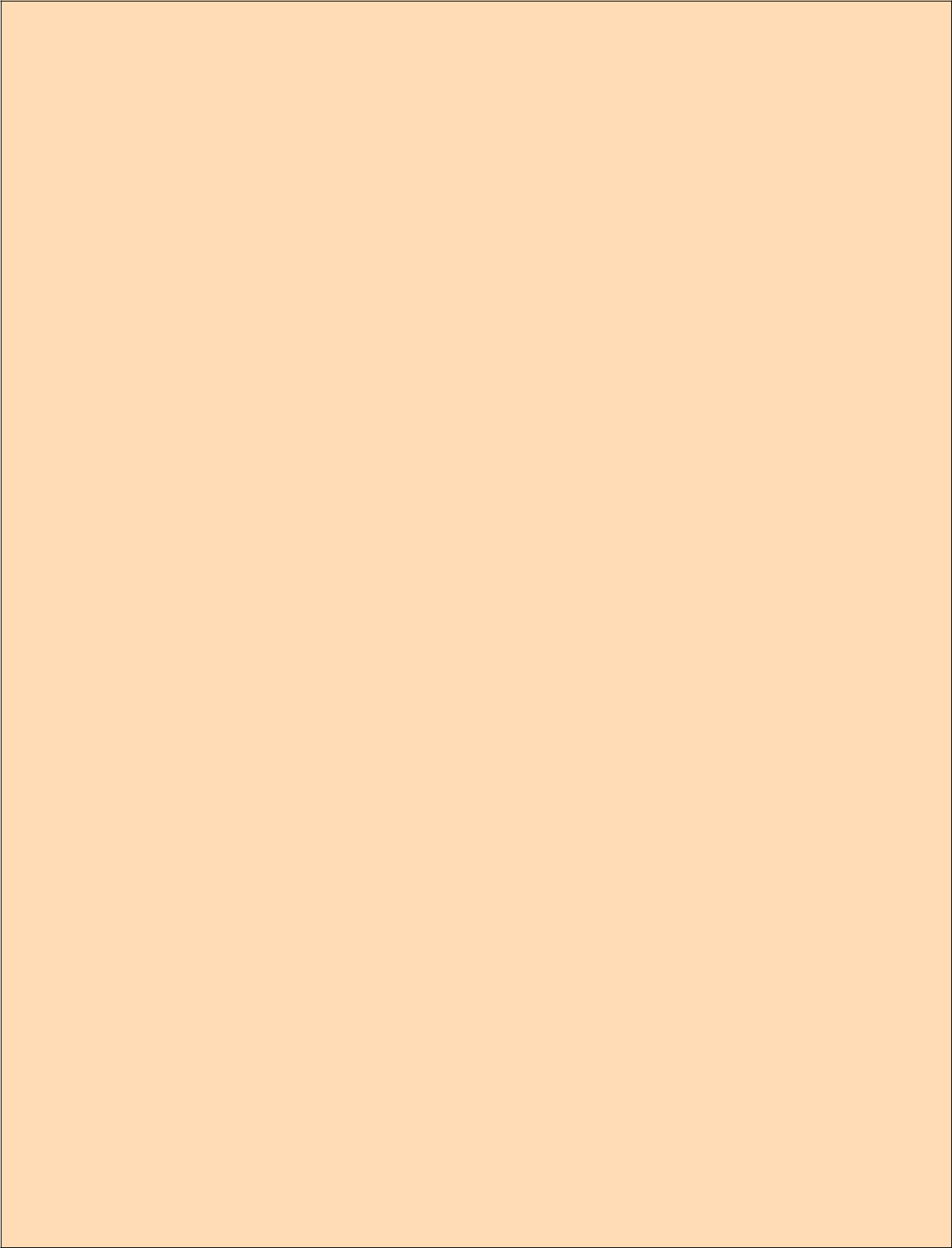
Disadvantages of FCFS

1.The scheduling method is non preemptive, the process will run to the completion.

2.Due to the non-preemptive nature of the algorithm, the problem of starvation may occur.

3.Although it is easy to implement, but it is poor in performance since the average waiting time is higher as compare to other scheduling algorithms.

**Corresponding 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: ");

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++)

{

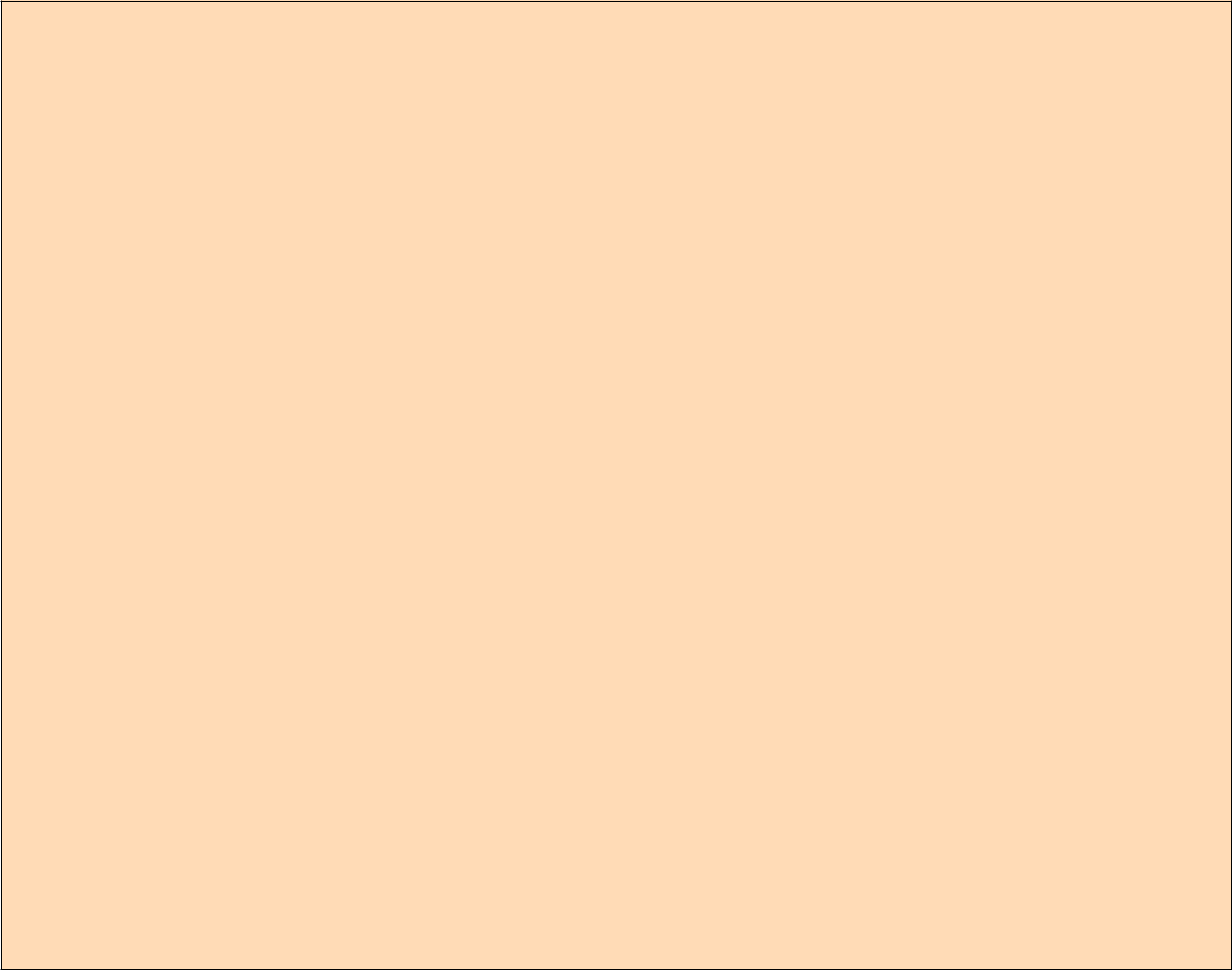
wt[i]=0;

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);

printf("\n\n");

return 0;

}

**Output:**

