

***OPERATING SYSTEM***

**Lab Project Report**

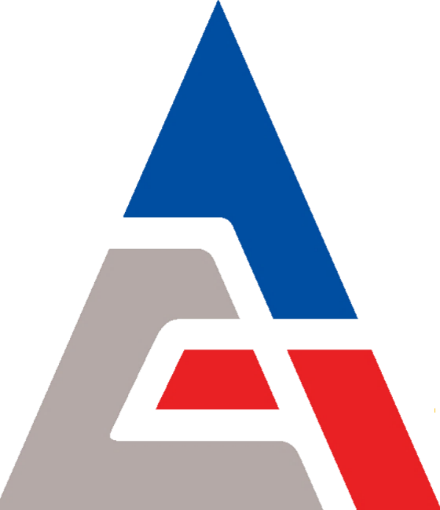
**Bachelor of Technology**

**In**

**Computer Science Engineering**

**Batch**

**(2017 – 2021)**



SUBMITTED TO:-SUBMITTED BY:-

**Er. Sanjeev Kumar Aayush Khanna (1701214)**

**Basudev Mahato (1701244)**

**Gulmehak Kaur Sandhawalia (1701260)**

**ACKNOWLEDGEMENT**

This is a humble effort to express our sincere gratitude towards those who have guided and helped us to complete this project

A project is major milestone during the study period of a student. As such this project was a challenge to us and was an opportunity to prove our caliber. We are highly grateful and obliged to each and everyone making us help out of problems being faced by us.

It would not have been possible to see through the undertaken project without the guidance of Er. Sanjeev Kumar. It was purely on the basis of their experience and knowledge that we able to clear all the theoretical and technical hurdles during the development phases of this project work.

Last but not the least we are very thankful to our Head of Department Er. Vinod sharma and all Members of Computer Science Deptt. who gave us an opportunity to face real time problems while fulfilling need of an organization by making projects for them

**DECLARATION**

We hereby declare that the project work entitled **“First Come First Serve Scheduling algorithm(Non primptive)”** is an authentic record of our work carried out as requirements of Institutional Training project for the award of degree of B.Tech(CSE), **Amritsar Group Of Colleges, Amritsar,** under the guidance of Er. Sanjeev Kumar

(Signature of student)

**Aayush Khanna (1701214)**

**Basudev Mohato (1701244)**

**Gulmehak Kaur (1701237)**

Certified that the above statement made by the student is correct to the best of our knowledge and belief.

**Faculty Coordinator**

Er. Sanjeev Kumar (Associate Professor – CSE Department)

INDEX PAGE

|  |  |  |
| --- | --- | --- |
| S.No | Content | Page no |
| 1 | Introduction to Subject |  |
| 2 | Introduction to project |  |
| 4 | Step by Step implementation of Project |  |
| 5 | Snapshot at every step |  |
|  |  |  |

###### **INTRODUCATION TO SUBJECT**

An operating system is a software that manages computer hardware. The hardware must provide appropriate mechanisms to ensure the correct operation of the computer system and to prevent user programs from interfering with the proper operation of the system.

An operating system is a program that controls the execution of application programs and acts as an interface between the user of a computer and the computer hardware. It is concerned with the allocation of resources and services such as memory, processors , devices and information.

The operating system correspondingly includes programs to manage these resources , such as traffic controller, a scheduler ,memory management module , I/O programs and a file system.

Functions of operating system:

* **Convenience:-** As OS makes a computer more convenient to use.
* **Efficiency:-** An OS allows the computer system resources to be used in an efficient manner.
* **Ability to Evolve:-** An OS should be constructed in such a way as to permit the wffective development, testing and introduction of new system functions without at the same time interfering with service.

The operating system is designed to serve two basic purposes:

* It consists the allocation and use of the computing System’s resources among the various user and tasks.
* It provides an interface between the computer hardware and the programmer that simplifies and makes feasible for coding ,creation , debugging of the application programs.

The operating system must support the following tasks:

* Provides the facilities to create , modification of programs and data files using an editor.
* Access to the compiler for translating the user program from high level language to machine language.
* Provide a loader program to move the compiled program code to the computer’s memory for execution.
* Provide routines that handles the details of I/O programming.

Types of operating systems:

* **Batch operating system:-** Sequence of jobs in a program on a computer without manual interventions.
* **Time sharing operating system:-** It allows many users to share the computer resources.
* **Distributive operating system:-** Manages a group of different computers and make appear to be a single computer.
* **Network operating system:-** computers running in different operating system can participate in common network. It is used for security purposes.
* **Real Time operating system:-** Meant applications to fix the deadlines.

**Examples of operating system:-**

* Windows(GUI based PC)
* GNU/Linux(Personal, Workstations, ISP, file and print server , Three-tier client/server)
* macOS(Macintosh),used for Apple’s personal computers and work stations(McBook,iMac).
* Android (Google Operating System for smartphones/tablets/smartwatchers).
* IOS(Apple’s OS for iphone, iPad and iPod Touch)

**INTRODUCTION TO PROJECT**

**CPU Scheduling**:-

CPU scheduling is the basis of multiprogrammed operating systems. CPU scheduling is the process which allows one process to use the CPU while the execution of another process is on hold (in waiting state) due to unavailability of any resource like I/O etc. thereby making full use of CPU. The aim of CPU scheduling is to make the system efficient, fast and fair.

Whenever the CPU becomes idle, the operating system must select one of the processes in the ready queue to be executed. The selection process is carried out by the short-term scheduler (CPU scheduler). The scheduler selects from among the processes in memory that are ready to execute, and allocates the CPU to one of them.

Types of CPU Scheduling: -

1. First Come First Serve (FCFS)

2. Shortest Job First (SJF)

3. Priority Scheduling

4. Round Robin Scheduling.

**First Come First Serve (FCFS): -**

By the far simplest CPU Scheduling algorithm is the First Come, First Served (FCFS) scheduling algorithm.

With this scheme, the process that requests the CPU first, is allocated the CPU first. The implementation of FCFS policy is easily managed with a FIFO queue.

When a process enters the ready queue, its PCB is linked onto the tail of the queue.

When the CPU is free, it is allocated to the process at the head of the queue.

The running process is then removed from the queue. The code for FCFS scheduling is simple to write and understand.

We have to find out Completion Time (CT), Turn Around Time (TAT), Waiting Time (WT):

**Arrival Time (AT): -**Time at which process arrives at the ready queue.

**Process ID(PID): -** It is the unique number for each process.

**Burst Time (BT): -** The time required by a process for CPU execution.

**Completion Time (CT): -** Time at which process completes its execution.

CT = BT+CT

**Turn Around Time (TAT): -** The amount of Time Taken to complete the process.

TAT = CT-AT

**Waiting Time (WT):-** The amount of time spent ready to run but not running.

WT = TAT-BT

**EXAPMLE**

Consider the following given table:-

|  |  |  |
| --- | --- | --- |
| Process Number | Arrival Time (AT) | Burst Time (BT) |
| P1 | 0 | 7 |
| P2 | 3 | 5 |
| P3 | 4 | 3 |
| P4 | 5 | 6 |
| P5 | 9 | 4 |

**Sol: -**

Gantt chart: -

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P5 | P3 | P1 | P4 | P2 |

0 7 12 15 21 25

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Process Number | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting Time |
| P1 | 0 | 7 | 7 | 7 | 0 |
| P2 | 3 | 5 | 12 | 9 | 4 |
| P3 | 4 | 3 | 15 | 11 | 8 |
| P4 | 5 | 6 | 21 | 16 | 10 |
| P5 | 9 | 4 | 25 | 16 | 12 |

Completion Time = Burst Time + Completion Time

Turn Around Time = Completion Time – Arrival Time

Waiting Time = Turnaround time – Burst Time

Average Turnaround time = (7+9+11+16+16)/5 = 11.8

Average Waiting Time = (0+4+8+10+12)/5 = 6.8

CODING

#include<iostream.h>

#include<stdio.h>

#include<conio.h>

int main ()

{

clrscr();

int n, i, j, k, t, c;

float avwt =0.0, avtat=0.0;

cout<<”\*\*\*\*\*\*\*\*\*First Come First Serve”\*\*\*\*\*\*\*\*\*\*”;

cout<<”Enter the number of processes to be executed”;

cin>>n;

int at[100],bt[100],ct[100],wt[100],tat[100],nat[100];

clrscr();

cout<<”Enter arrival time”<<endl;

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

{

cout<<”Processes”<<i+1<<”:”<<endl;

cin>>at[i];

}

cout<<”Enter burst time”<<endl;

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

{

cout<<”Processes”<<i+1<<”:”<<endl;

cin>>bt[i];

}

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

{

j = i-1;

k = at[i];

while ((j>=0)&&(k<at[j]))

{

at[j+1] = at[j];

at[j] = k;

t = bt[j+1];

bt[j+1] = bt[j];

bt[j] = t;

j--;

}

}

c = at[0];

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

{

ct[i] = c+bt[i];

nat[i] = c;

c = ct[i];

}

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

{

wt[i] = nat[i] – at[i];

}

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

{

tat[i] = ct[i]-at[i];

}

clrscr();

cout<<”NUMBER OF PROCESSES ENTERED”<<” “<<n<<endl;

cout<<”PROCESS”<<”\t”<<”at”<<”\t”<<”bt”<<”\t”<<”ct”<<”\t”<<”tat”<<”\t”<<”wt”<<endl;

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

{

cout<<i<<”\t”<<at[i]<<”\t”<<bt[i]<<”\t”<<ct[i]<<”\t”<<tat[i]<<”\t”<<”wt[i]”<<”\t”<<endl;

}

cout<<”Average waiting Time”;

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

{

avwt = avwt+wt[i];

}

cout<<(avwt/n)<<endl;

cout<<”Average turnaround time<<endl;

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

{

avtat = avtat[i]+tat[i];

}

cout<<(avtat/n)<<endl;

cout<<endl;

getch();

return 0;

}

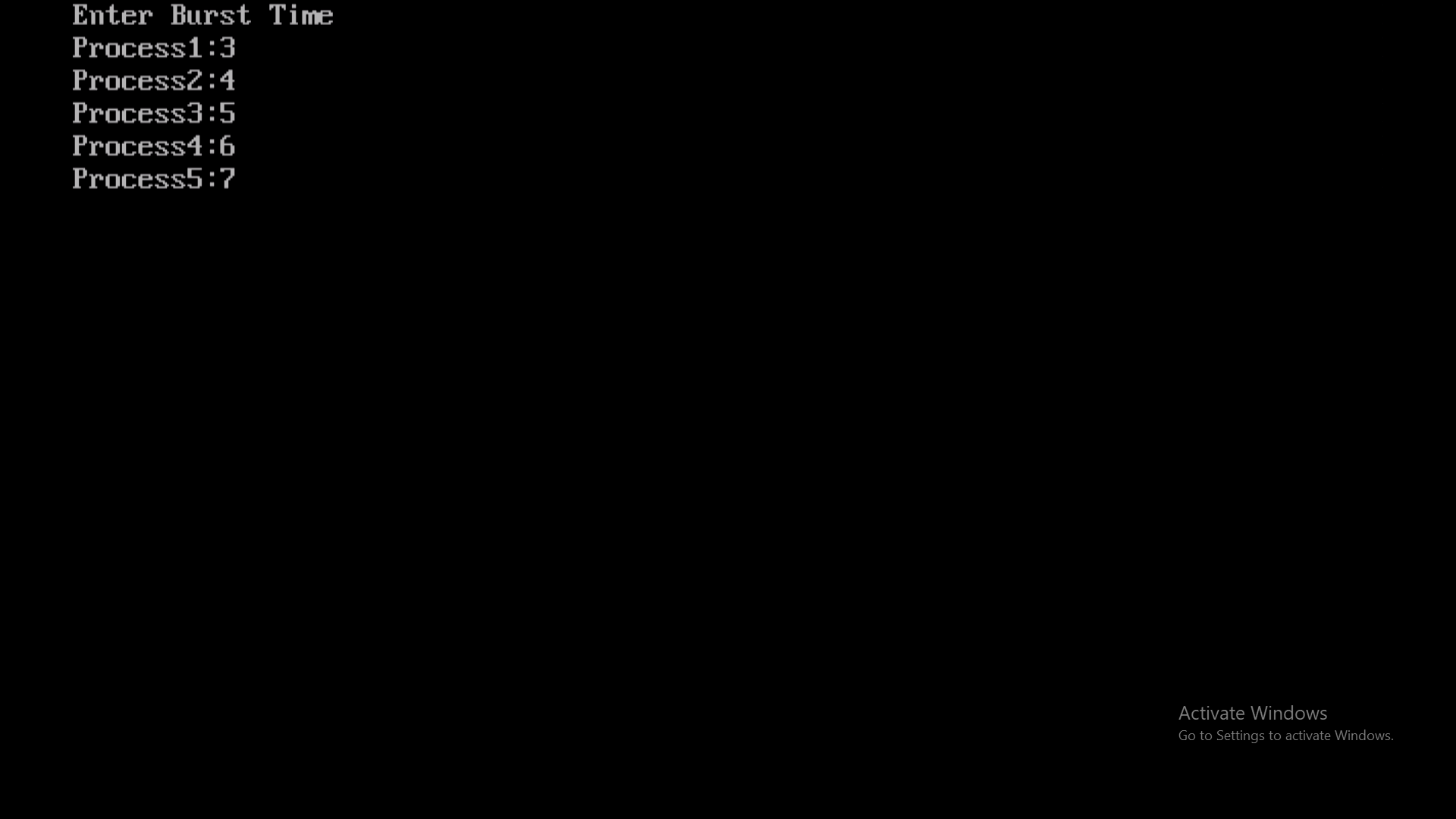
**OUTPUT**

Enter the number of processes to be executed:



Enter the arrival time of each process:

Enter burst time of each process:



Average Turnaround Time: -

Average Waiting Time: -

Table Display: -

