**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**“JnanaSangama”, Belgaum -590014, Karnataka.**



**LAB REPORT**

**on**

**Operating Systems**

**(22CS4PCOPS)**

***Submitted by:***

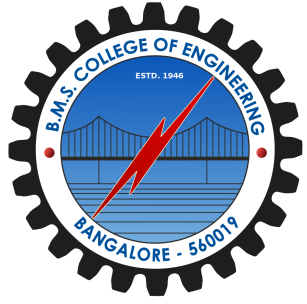
**Bhoomi Udedh (1BM23CS066)**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**



**B.M.S. COLLEGE OF ENGINEERING**

**(Autonomous Institution under VTU)**

**BENGALURU-560019**

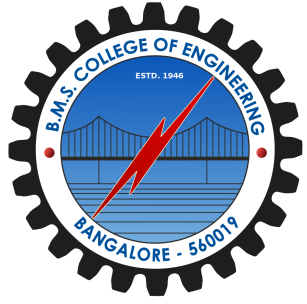
**June 2023 - August 2023**

**B. M. S. College of Engineering,**

**Bull Temple Road, Bangalore 560019**

(Affiliated To Visvesvaraya Technological University, Belgaum)

**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “**Operating Systems**” carried out by **Bhoomi Udedh (1BM23CS066),** who is bonafide student of **B. M. S. College of Engineering.** It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022-23. The Lab report has been approved as it satisfies the academic requirements in respect of **Operating Systems - (22CS4PCOPS)** work prescribed for the said degree.

**Basavaraj Jakkalli**              **Dr. Kavitha Sooda**

Associate Professor Professor and Head

Department of CSE Department of CSE

BMSCE, Bengaluru BMSCE, Bengaluru

**Table Of Contents**

| **Lab Program No.** | **Program Details** | **Page No.** |
| --- | --- | --- |
| 1 | FCFS AND SJF | 3-6 |
| 2 | PRIORITY AND ROUND ROBIN | 6-12 |
| 3 | MULTI-LEVEL QUEUE SCHEDULING | 13-16 |
| 4 | RATE-MONOTONIC AND EARLIEST DEADLINE FIRST | 17-23 |
| 5 | PRODUCER-CONSUMER PROBLEM | 24-26 |
| 6 | DINERS-PHILOSOPHERS PROBLEM | 26-29 |
| 7 | BANKERS ALGORITHM(DEADLOCK AVOIDANCE) | 30-32 |
| 8 | DEADLOCK DETECTION | 33-35 |
| 9 | CONTIGUOUS MEMORY ALLOCATION(FIRST, BEST, WORST FIT) | 36-39 |
| 10 | PAGE REPLACEMENT(FIFO, LRU, OPTIMAL) | 40-47 |
| 11 | DISK SCHEDULING ALGORITHMS(FCFS, SCAN, C-SCAN) | 48-53 |

**Course Outcomes**

**CO1:** Apply the different concepts and functionalities of Operating System.

**CO2:** Analyse various Operating system strategies and techniques.

**CO3:** Demonstrate the different functionalities of Operating System.

**CO4:** Conduct practical experiments to implement the functionalities of Operating system.

1. **Experiments**
   1. **Experiment - 1**
      1. **Question:**

**Write a C program to simulate the following non-pre-emptive CPU scheduling algorithm to find turnaround time and waiting time.**

1. **FCFS**
2. **SJF**
   * 1. **Code:**

#include<stdio.h>

int n, i, j, pos, temp, choice, Burst\_time[20], Waiting\_time[20], Turn\_around\_time[20], process[20], total=0;

float avg\_Turn\_around\_time=0, avg\_Waiting\_time=0;

int FCFS()

{

Waiting\_time[0]=0;

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

{

Waiting\_time[i]=0;

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

Waiting\_time[i]+=Burst\_time[j];

}

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

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

{

Turn\_around\_time[i]=Burst\_time[i]+Waiting\_time[i];

avg\_Waiting\_time+=Waiting\_time[i];

avg\_Turn\_around\_time+=Turn\_around\_time[i];

printf("\nP[%d]\t\t%d\t\t\t%d\t\t\t\t%d",i+1,Burst\_time[i],Waiting\_time[i],Turn\_around\_time[i]);

}

avg\_Waiting\_time =(float)(avg\_Waiting\_time)/(float)i;

avg\_Turn\_around\_time=(float)(avg\_Turn\_around\_time)/(float)i;

printf("\nAverage Waiting Time:%.2f",avg\_Waiting\_time);

printf("\nAverage Turnaround Time:%.2f\n",avg\_Turn\_around\_time);

return 0;

}

int SJF()

{

//sorting

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

{

pos=i;

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

{

if(Burst\_time[j]<Burst\_time[pos])

pos=j;

}

temp=Burst\_time[i];

Burst\_time[i]=Burst\_time[pos];

Burst\_time[pos]=temp;

temp=process[i];

process[i]=process[pos];

process[pos]=temp;

}

Waiting\_time[0]=0;

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

{

Waiting\_time[i]=0;

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

Waiting\_time[i]+=Burst\_time[j];

total+=Waiting\_time[i];

}

avg\_Waiting\_time=(float)total/n;

total=0;

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

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

{

Turn\_around\_time[i]=Burst\_time[i]+Waiting\_time[i];

total+=Turn\_around\_time[i];

printf("\nP[%d]\t\t%d\t\t\t%d\t\t\t\t%d",process[i],Burst\_time[i],Waiting\_time[i],Turn\_around\_time[i]);

}

avg\_Turn\_around\_time=(float)total/n;

printf("\n\nAverage Waiting Time=%f",avg\_Waiting\_time);

printf("\nAverage Turnaround Time=%f\n",avg\_Turn\_around\_time);

}

int main()

{

printf("Enter the total number of processes:");

scanf("%d",&n);

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

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

{

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

scanf("%d",&Burst\_time[i]);

process[i]=i+1;

}

while(1)

{ printf("\n-----MAIN MENU-----\n");

printf("1. FCFS Scheduling\n2. SJF Scheduling\n");

printf("\nEnter your choice:");

scanf("%d", &choice);

switch(choice)

{

case 1: FCFS();

break;

case 2: SJF();

break;

default: printf("Invalid Input!!!");

}

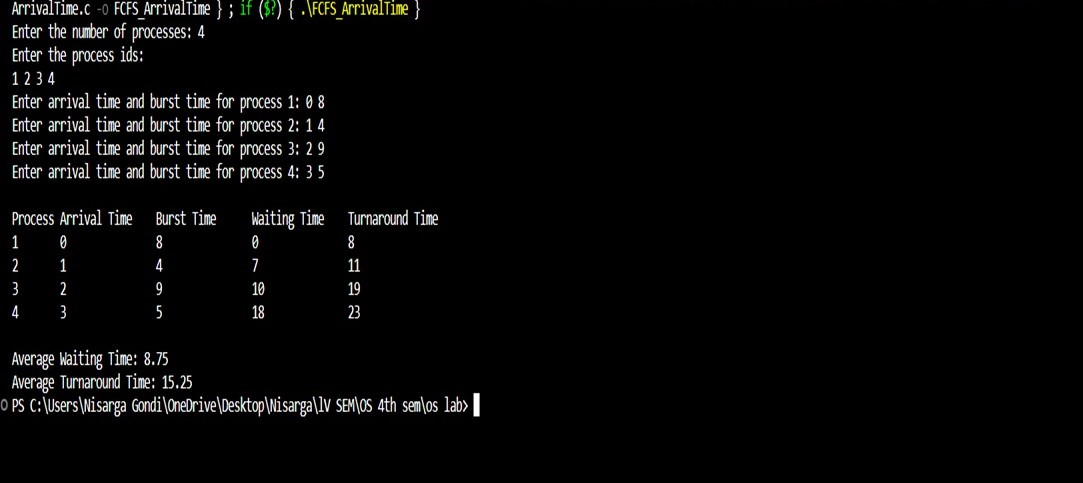
}

return 0;

}

* + 1. **Output:**

**a.**



**b.**

