An Improved Fuzzy-Based CPU Scheduling (IFCS) Algorithm for Real Time Systems

H. S. Behera, Ratikanta Pattanayak, Priyabrata Mallick

*Abstract— **Till now various types of scheduling algorithms are used for determining which process should be executed by the CPU when there are multiple no. of processes to be executed.There are many conventional approaches to schedule the tasks but no one is absolutely ideal. In this paper an improved fuzzy technique has been proposed to overcome the drawbacks of other algorithms for better CPU utilization,throughput and to minimize waiting time and turn around time.***

Index terms : Task, process, fuzzification, priority, cpu utilization,fuzzy scheduler, turnaround time,scheduling effeciency

I.INTRODUCTION

In present scenario real time systems plays an important role in the modern world without which human life can't be completed[1]. From mobile to missile we are dealing with various house hold real time devices in our daily life. Like medical imaging systems, industrial control systems, display systems, various scientific experiments etc[2].But the main constraints of real time processes is that it should be completed before a fixed timebound called deadline[3-4].

Although there are many scheduling algorithms such as FCFS,SJF,HRRN,PRIORITY Scheduling etc. are available But it's critical to schedule the real-time tasks. Let us consider the case of FCFS, In which processes are served according to their arrival time. The task which arrives first will be served first. It doesn't give guarantee to complete the execution before deadline .SJF consider the job with shortest burst time to be executed next.If a real-time process having relatively larger CPU burst then it'll leads to starvation[5]. Priority scheduling may be a better choice for real-time scheduling but it'll face the similar problem i.e low priority processes will always starved[6]. In the proposed algorithm we try to take the advantages of SJF,HRRN and PRIORITY scheduling algorithms to find a new priority by implementing fuzzy logic. In crisp or digital logic we are dealing with either 0 or 1,but in fuzzy logic varies between 0 and 1.

Manuscript Received February 09, 2012.

Ratikanta Pattanayak, Dept. of Computer Science & Engg., VSSUT, Burla, India, 09658183033., (e-mail: ratikantapattanayak@hotmail.com).

Priyabrata Mallick, Dept. of Computer Science & Engg., VSSUT, Burla, India, 09861873053., (e-mail: priyavssut@gmail.com).

H. S. Behera, Dept. of Computer Science & Engg., VSSUT, Burla, India, 9437083891, (e-mail: hsbehera_india@yahoo.com).

II.RELATED WORKS

Many Researchers have tried to implement fuzzy logic to schedule the processes. A fuzzy-based CPU scheduling algorithm is proposed by Shata J. Kadhim et. al[1].Round robbin scheduling using neuro fuzzy approach by Mr. Jeegar A Trivedi et. al[7].Soft real-time fuzzy task scheduling for multiprocessor systems by Mahdi Hamzeh et. al[8].Efficient soft real-time processing is proposed by C. Lin et. al[4].

III.SCHEDULING ALGORITHMS

CPU scheduling is nothing but selecting a single process from a bunch of processes from ready queue. This assignment is carried out by a software known as a scheduler.In order to measure the efficiency of a scheduling algorithm there are some criteria's such as Throughput(No. of processes completed per unit time),Turnaround time(The interval from the time of submission of a process to the time of completion),waiting time(The time spend by the process in the ready queue),Response time(The time interval between the submission of job until the first response is produced). Various traditional algorithms that we used earlier are :

A. FCFS:

First come first serve, is the simplest scheduling algorithm, FCFS simply arrange the processes in the order that they arrive in the ready queue. FCFS is a non preemptive scheduling algorithm. So there is less context switching overhead occurs. Throughput can be low, since long processes can hold the cpu. Turnaround time, waiting time and response time can be high for the same reason. No prioritization occurs, thus this system has trouble to meet deadlines of the processes.

B. SJF:

Shortest Job First is a scheduling policy that selects the waiting process with the smallest execution time to execute next. It's the most efficient process that we ever meet. However, it has a drawback of process starvation for process which will require a long time to complete if short processes are keep arrived continuously. It uses past behavior to indicate which process to run next, based on an estimate of its execution time . Shortest job next scheduling is rarely used outside of specialized environments because it requires accurate estimations of the runtime of all processes that are waiting to execute.

C. Priority scheduling:

The operating system assigns a fixed priority rank to every process, and the scheduler arranges the processes in the ready queue in order of their priority. Lower priority processes get interrupted by incoming higher priority processes. Overhead is not minimal, nor is it significant in this case. Waiting time and response time depend on the priority of the process. Higher priority processes have smaller waiting and response times. Deadlines can be easily met by giving higher priority to the earlier deadlined processes. Starvation of lower priority processes is possible if large no of higher priority processes keep arrived continuously.

. D. HRRN:

Highest Response Ratio Next scheduling algorithm proposed by Brinch Hansen is a to avoid limitations of SJF algorithm. It is similar to Shortest Job Next (SJN) in which the priority of each job is dependent on its estimated run time, and also the amount of time it has spent waiting in the ready queue. Jobs which gain higher priority the longer they wait, which prevents process starvation. In fact, the jobs that have spent a long time in waiting, can compete against those jobs estimated to have short run time.

$$\mathrm{Re}\, sponseRatio = \frac{WaitingTime + RunTime}{RunTime}$$

IV.PROPOSED ALORITHM

A. Overview:

Fuzzy logic is a superset of boolean logic and it deals with new aspects such as partial truth and uncertainty. Fuzzy inference is the process of formulating from a given input set to an output using fuzzy technique. The basic element of fuzzy logic are linguistic variables, Fuzzy sets and fuzzy rules. The linguistic variable values are generally adjectives such as “very high”,”high”,”medium”,”small”,”very small” and so on. In crisp logic we consider either true or false i.e a value is either belongs to the set or not. But in fuzzy logic each member have some percentage to which it belongs to the set, called membership value. This is denoted by μ. The values thus lie between 0 and 1 ie. 0 % or. 100%.We Take advantages of the well known algorithms that are shortest job first, priority and highest response ratio next scheduling algorithms.

A fuzzy set is a collection of various elements it generalize the concept of crisp set, allowing its elements to have partial membership. The degree to which an element “a” belongs to set A is characterized by membership function μ_a

.The membership function of fuzzy set corresponds to the indicator function of crisp set.It can be expressed in form of a curve that defines how each point in input is mapped to a membership value or a degree of truth between 0 and 1.

Let us consider the membership function

$\mu_x=1/(x+1)$

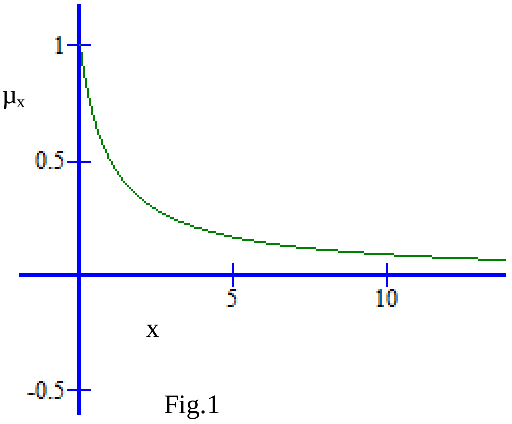


Fig.1

B. Fuzzy inference system:

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basic from which decisions can be made. The process of fuzzy inference involves all of the characteristics such as Membership Functions, Logical Operations, and If-Then rules. We can implement two types of fuzzy inference systems: Mamdani-type and Sugeno-type. These two types of inference systems vary somewhat in the way outputs are determined. In our proposed algorithm we take membership function as the ratio of actual value per (maximum value+1) present in the set.

C. Proposed architecture:

The proposed architecture shown in fig.2 is based on Mamdani-type architecture. Mamdani-type interface expect the output membership function to the fuzzy set. After the aggregation process there is afuzzy set for each output variable that needs defuzzification. In the proposed architecture burst time(BT),arrival time(AT) and priority(P) of the processes have been taken as the crisp input to the computational unit. Then membership of burst time(μ_b), priority($\mu_{p)}$ and response ratio(μ_h) are computed. These are the input to the fuzzy rule base where new priorities of processes are evaluated individualy for scheduling.

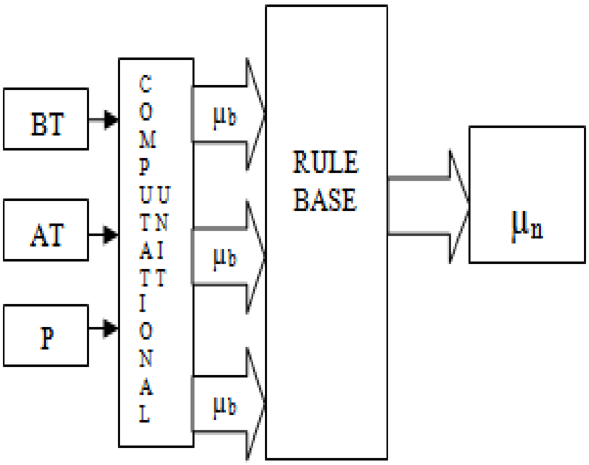
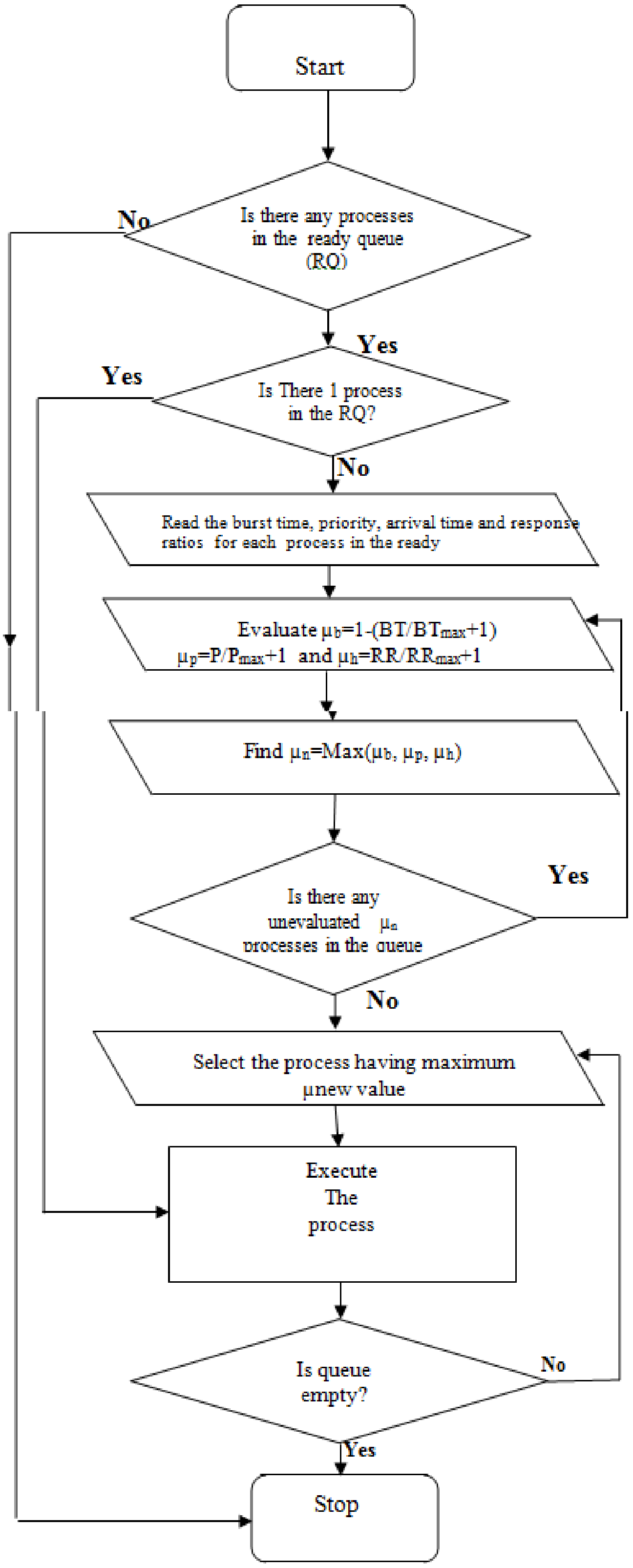


Fig.2 proposed architecture

D. Pseudo code of the proposed algorithm:

```
1.Intialize n processes with their burst time, arrival time and priorities.

2.Evaluate $\mu_p$ i.e membership value of task priority for individual processes by using the formula
   $\mu_p = \frac{\text{actual task priority}}{\text{(maximum task priority+1)}}$

3.Evaluate $\mu_b$ that is membership value of burst time for individual processes

   $\mu_b = 1 - \frac{\text{actual burst time}}{\text{(maximum burst time+1)}}$

4.Find response ratios of individual processes after each iteration.

5.Evaluate $\mu_h$ i.e membership value of response ratio

   $\mu_h = \frac{\text{actual response ratio}}{\text{(maximum response ratio+1)}}$

6.Evaluate $\mu_{ni}$: membership value of new priority after fuzzification for ith. process by the formula
$p_{ni} = \max\{ \mu_{bi}, \mu_{pi}, \mu_{hi} \}$
where 1<=i<=n

7.Apply bubble sort to get the descending order of new priorities.
for(i=1;i<n;i++)
{
   for(j=1;j<n-i;j++)
   {
     if ($p_{ni}$<$p_{ni+1}$)
     {
         Swap the two processes
     }
   }
}

8.Execute the processes in the sorted sequence

9.Stop & exit
```

E. Flowchart of the proposed algorithm:


Start
Is there any processes in the ready queue (RQ)
No
Yes
Is There 1 process in the RQ?
Yes
No
Read the burst time, priority, arrival time and response ratios for each process in the ready
Evaluate μb=1-(BT/BTmax+1)
μp=P/Pmax+1 and μh=RR/RRmax+1
Find μn=Max(μb, μp, μh)
Is there any unevaluated μn processes in the queue
Yes
No
Select the process having maximum μnew value
Execute The process
Is queue empty?
No
Yes
Stop

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

V.RESULT AND PERFORMANCE EVALUATION:

To demonstrate Proposed fuzzy algorithm we have taken different case studies and finally we compare graphically between other algorithms and Proposed fuzzy based algorithms.

A. Case Study 1:

Table 1

Processes Id	P1	P2	P3	P4	P5
Priority	2	7	5	6	1
Burst time	3	6	4	5	2
Arrival time	0	2	4	6	8
Response Ratio	1	1.167	2.25	2.4	4.5
New priority	.571	.875	.625	.75	.82

(a)Gantt chart for Priority scheduling.

P1	P2	P4	P3	P5

0 3 9 14 18 20

Waiting time : 4.8
Turn around time : 8.8

(b) Gantt chart for SJF scheduling

P1	P3	P5	P4	P2

0 3 9 11 15 20

Waiting time :3.6
Turn around time :7.6

(c) Gantt chart for HRRN scheduling

P1	P2	P3	P5	P4

0 3 9 13 15 20

Waiting time :4
Turn around time : 8

(d) Gantt chart for Fuzzy based CPU scheduling

P1	P2	P3	P4	P5

0 3 9 13 18 20

Waiting time:4.6
Turn around time:8.6

(e) Gantt chart for Improved Fuzzy based cpu scheduling

P1	P2	P5	P4	P3

0 3 9 11 16 20

Turn around time:7.8

B. Case Study 2 :

Table 2

Process Id	P1	P2	P3	P4	P5	P6	P7
Priority	1	3	2	6	5	11	7
Burst Time	18	2	1	4	3	12	13
Arrival time	0	0	0	0	0	0	0
Response ratio	1	7.33	19	7	8	3.33	4.077
New Prority	.083	.895	.95	.79	.84	.917	.58

(a) Gantt chart for Priority scheduling.

P6	P7	P4	P5	P2	P3	P1

0 12 25 29 32 34 35 53

Waiting time : 23.86
Turn around time : 31.43

(b) Gantt chart for SJF scheduling

0 1 3 6 10 22 35 53

Waiting time :11
Turn around time :18.57

(c) Gantt chart for HRRN scheduling

P1	P3	P2	P5	P4	P6	P7

0 18 19 21 24 28 40 53

Waiting time :21.43
Turn around time : 29

(d) Gantt chart for Fuzzy based cpu scheduling

P4	P5	P6	P7	P2	P3	P1

0 4 7 19 32 34 35 53

Waiting time:18.714
Turn around time:26.29

(e) Gantt chart for Improved Fuzzy based cpu scheduling

P3	P6	P2	P5	P4	P6	P7

0 1 13 15 18 22 35 53

Waiting time:14.86
Turn around time:22.43

C. CASE STUDY 3 :

Table 3

Process Id	P1	P2	P3	P4	P5	P6	P7	P8	p9
Priority	2	4	3	5	6	12	1	11	9
Burst Time	19	3	2	6	4	5	18	1	13
Arrival time	2	3	5	3	2	1	1	2	0
Response Ratio	3.63	5.33	5.5	5.17	5.25	5.4	2.83	11	1
New priority	.3	.85	.9	.7	.8	.92	.24	.95	.69

(a) Gantt chart for Priority scheduling.

P9	P6	P8	P5	P4	P2	P3	P1	P7

0 13 18 19 23 29 32 34 53 71

Waiting time : 22.44
Turn around time : 30.33

(b) Gantt chart for SJF scheduling

P9	P6	P8	P5	P4	P2	P3	P1	P7

0 13 14 16 19 23 28 34 52 71

Waiting time :20
Turn around time :27.89

(C) Gantt chart for HRRN scheduling

P9	P8	P3	P2	P5	P6	P4	P7	P1

0 13 14 16 19 23 28 34 52 71

Waiting time :20
Turn around time:27.89

(d) Gantt chart for Fuzzy based cpu scheduling

P9	P8	P3	P2	P5	P6	P4	P7	P1

0 13 14 19 23 29 32 34 52 71

Waiting time:21.89
Turn around time:29.78

(e) Gantt chart for Improved Fuzzy based cpu scheduling

P9	P8	P3	P2	P5	P6	P4	P7	P1

0 13 14 19 21 24 28 34 53 71

Waiting time:20.78
Turn around time:28.67

D. Performance Evaluation:

Below shows a comparison between proposed algorithm versus other algorithms.

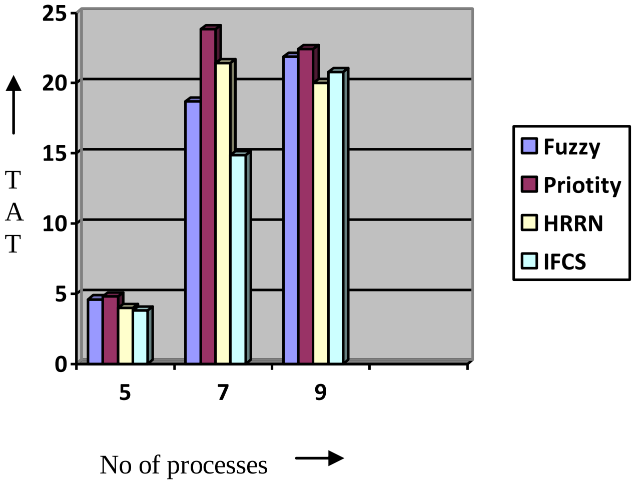


Fig.3 Waiting time comparision between various algorithms

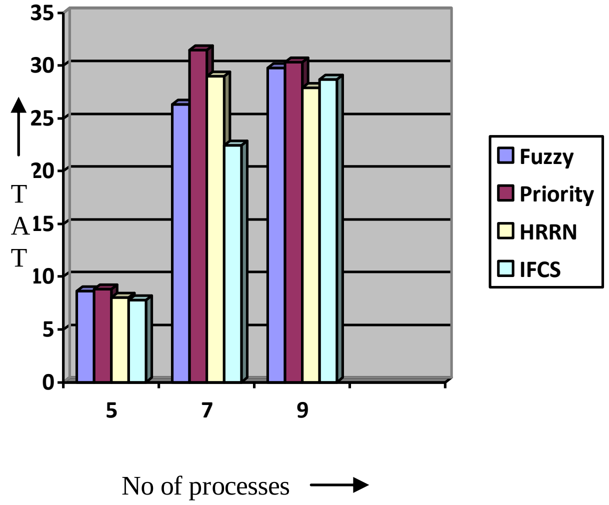


Fig.4Turn around time comparision between various algorithms

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

E.Performance Evaluation by simulation:

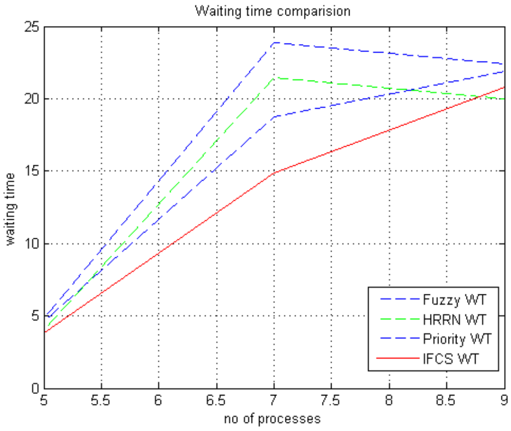


Fig.5

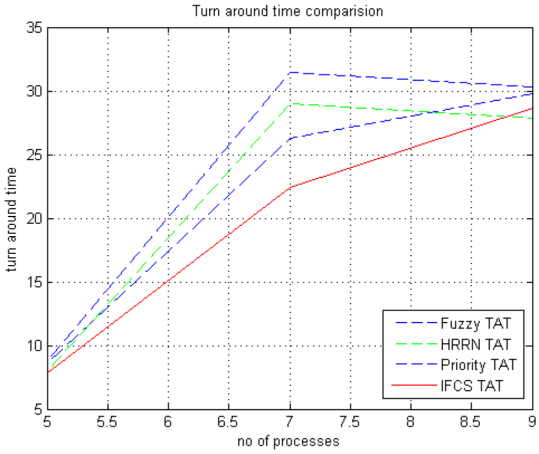


Fig.6

VI. CONCLUSION AND FUTURE WORK

The proposed improved fuzzy scheduling algorithm reduces or minimizes the waiting time and turn around time .This fuzzy schedule can be further improved by choosing more and more accurate formula for evaluating fuzzy membership value which may further reduces the waiting time and turn around time .

REFERENCES:

[1]Shata J. Kadhim , Kasim M. Al-Aubidy : ComputerEng. Dept, Al-Blaqa''Design and Evaluation of a Fuzzy Based CPU schedulilnlg Algorithm'' Applied University, Al-Salt, Jordan Computer Eng. Dept, Philadelphia University, Amman, Jordan,Springer-verlag Berlin Heidelberg 2010, CCIS 70,pp. 45-52,2010

[2] Stallings, Stallings, W.: Operating Systems Internals and Design Principles, 5th edn. Prentice-Hall,Englewood Cliffs (2004).

[3] Yaashuwanth .C, Dr. R. Ramesh: ,"Design of Real Time Scheduler Simulator and Devlopment of Modified Round Robin Architecture ",IJCSNS ,VOL.10 No.3,March (2010)

[4]C. Lin and S. A. Brandt, "Efficient soft real-time processing in an integrated system," in Proc. 25th IEEE Real-Time Systems Symp.,(2004).

[5] I. E. W. Giering and T. P. Baker, "A tool for the deterministic scheduling of real-time programs implemented as periodic Ada tasks," Ada Lett., vol. XIV, pp. 54-73, (1994).

6.Shahzad, B., Afzal, M.T.: ,"Optimized Solution to Shortest Job First by Eliminating the Starvation". In: The 6th Jordanian Inr. Electrical an Jordan (2006)

7.Mr . Jeegar A Trivedi and Dr.Priti Srinivas Sajja ,"Improving efficiency of round robin scheduling using neuro fuzzy approach " ,IJRRCS vol.2,No. 2,April 2011

8. Mahdi Hamzeh,Sied Mehdi Fakhraie and Caro Lucas ,"Soft real time fuzzy task scheduling for multiprocessor systems",world academy of science,engineering and technology 28 (2007).

AUTHOR'S BIOGRAPHY

Dr. H.S Behera is working as a faculty in Department of Computer Science and Information Technology ,Veer Surendra Sai University of Technology (VSSUT) ,Burla . His areas of interest are Distributed system, Data minning and soft computing.

Mr. Priyabrata Mallick is a final year B.Tech Student in Information Technology of Veer Surendra Sai University of Technology(VSSUT),Burla .His areas of interest are Embedded system, Microprocesser, Operating system and Artificial intelligence.

Mr. Ratikanta Pattanayak is a final year B.Tech student in Information technology of Veer Surendra Sai University of Technology,(VSSUT) Burla. His areas of interest are Soft computing, Artifical Intelligence , operating system and microprocessor.