D.S = L.C.H + T.C.S

Sr. No. _____ Date : _____

multiprocessing / multicomputing

multiprocessing

Every processor have their memory and they can also access the memory of other processor.

parallelism

(i) Instruction parallelism
(ii) Loop parallelism

parallel systems are categorized as
(i) UMA - uniform memory Access (e.g multiprocessors)
(ii) NumA — Non uniform memory Access (e.g multicomputers)
(iii) Array processors - (Global clock) may be multicomputers.

* Numa used when course grain

Sr. No. _____ Date : _____

parallel computing

Amdahl's Law

$$\text{Speed up}(S) = \frac{T_1}{T_n} = \frac{1}{F + (1-F)/n}$$

$$\text{Efficiency} = \frac{\text{speed up}}{n}$$

question Suppose 90% of the operation of a particular program or algo can be executed in parallelism what is its the best speed up? what is the best speed up with unlimited amount of processor.

answer A quad core processore could speed computer by a factore 4 but this rarely happed ~~are~~ Huge use to computer porcatage of program execution that needs to be distributed across all four ~~4~~ course.

$$\frac{1}{F + (1-F)/n}$$

$$S = \frac{1}{F + (1-F)/4}$$

$$4F + 1 - F = 1$$

$$3F = 0$$

Sr. No. _____ Date : _____

Gustafson's Law

W — Not fixed
n - fixed

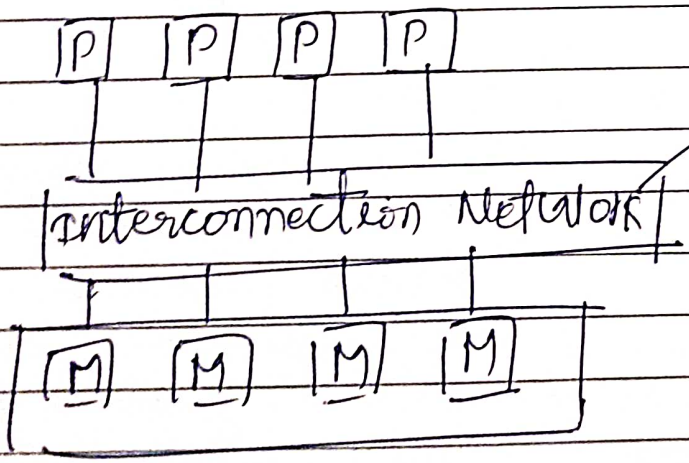
$$S \leq F + n(1-F) \Rightarrow n + (1-n)F$$

question 5% time spent on single processor
64 processor machine
compute the speed up

$$64 - 63 \times 0.05 \quad \frac{1}{20} \quad 2$$

$$64 - 32.5$$

$$= 32.5$$

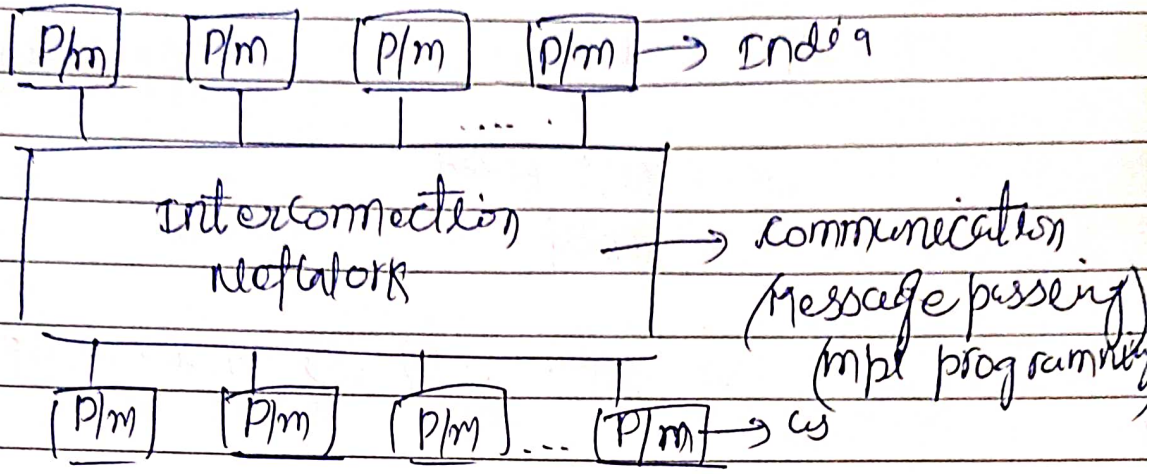


→ communication
↓
shared memory
(open mp programming)

UMA

tightly coupled

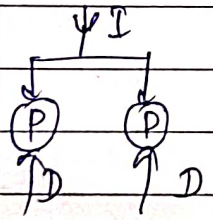
Sr. No. _____ Date : _____



NUMA

Flynn's Taxonomy

tightly coupled) SIMD - single instruction multiple data
MIMD multiple instruction multiple data
SISD - single instruction single data
tightly coupled MISD - multiple instruction single data



SIMD

H/W - LC
S/W - TC

Distributed system

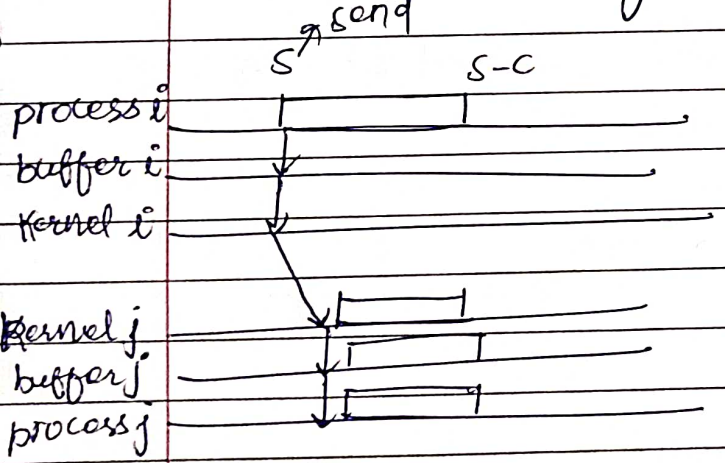
Loosely coupled Hardware and tightly coupled software

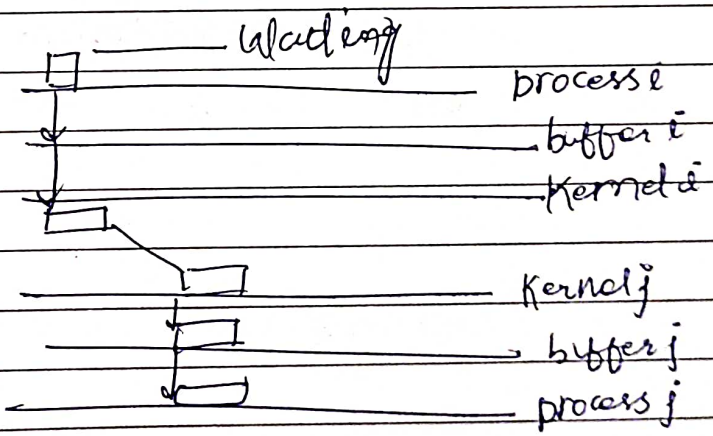
Sr. No. _____ Date : _____

1. LHW + LSW
   eg - [NOS] Network operating system.
2. LHW + TSW —— distributed system
3. THW + TSW ← eg embeded system, multiprocessor system.

Synchronous / Asynchronous systems
(blocking) (Non blocking)

synchronous systems
Advantages
(1) Reliability





Sr. No. _____ Date : _____

process synchrony - those multiprocessor systems which is using the concept of synchronization.

It indicates that all the processors execute in lock step with their clocks synchronized.

~~It does not use~~
* distributed system does not use process synchrony.

(message passing interface)
MPI is a library

(Rpc - Remote procedure call) - standard library
RMI - Remote method Invocation

Distributed system challenges

(i) Global clock is missing
(ii) synchronization of processes.
(iii) Communication
(iv) process management
(v) Naming
(vi) Consistency and Replication.

Sr. No. _____ Date : _____

CHALLEGES

→ Fault tolerance
→ Security
→ Api's
→ Scalibility and modularity
→ ***

* Scalibility with respect to distributed system (parallel system)
— scalibility is the measure of its capacity to ~~increase~~ increase the speedup in proportion to the number of processing elements.

Synchronous execution	Asynchronous execution
(i) processor synchrony	(i) No processor synchrony
(ii) ~~Message delay will be finite~~ Low drift rate bound.	(ii) Message delay will be finite but unbounded
(iii) ~~there must be upperbound~~ for execution of Anything	(iii) there is no upperbound. (iv) there is no bound on drift rate of processors

Sr. No. _____ Date : _____

अगर कोई system asynchronous execution कर रहा है

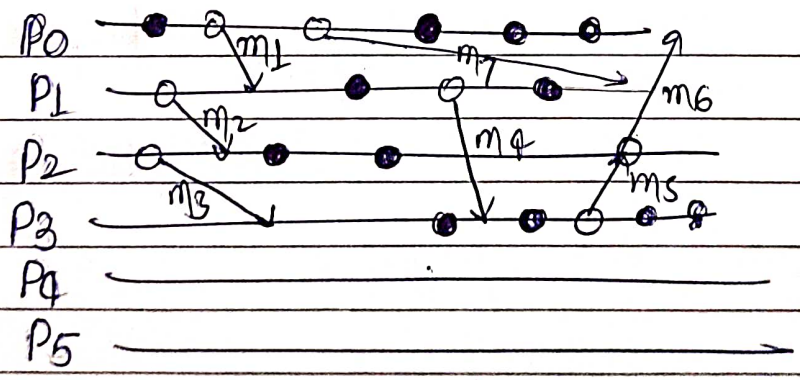
synchronize

Asynchronous execution system → coarse grain code → virtual synchrony

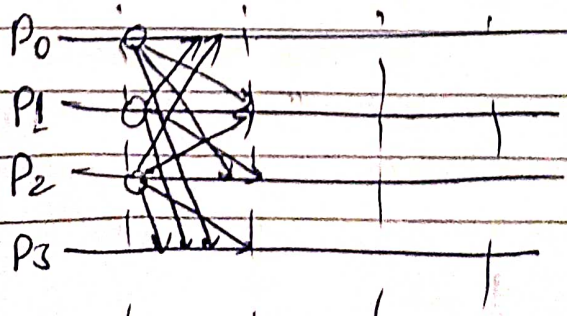
को synchronize करना चाहता है तो उसे virtual synchrony कहते हैं।

* Every process has there corresponding Logical time for execution in process synchrony.

Asynchronous execution



synchronous execution



Sr. No. _____ Date : _____

$$\text{propagation delay} = \frac{\text{distance}}{\text{transmission speed}}$$

$$\text{Transmission delay} = \frac{\text{length of packet}}{\text{transmission rate}}$$

question If the distance between the two points is ~~4 phase~~ 48000 km and propagation speed is $2.4 * 10^8$ m/s in cable ~~then~~ find out the ~~prog~~ propagation delay.

$$\frac{2 \cancel{48} \; 10}{\cancel{24} \times 10^2} = .2$$

$$\frac{2}{10}$$

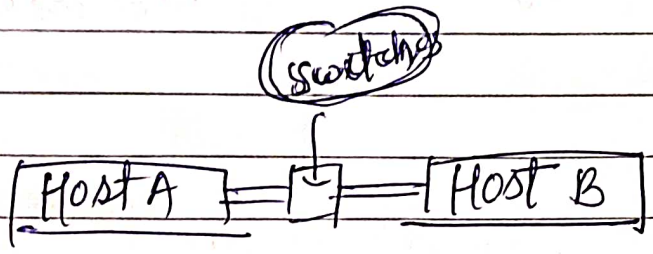
question Transmission of 1500 B (12000 bits) using transmission rate of 100 mbps will take

$$= \frac{15}{2^{20}} = \frac{120}{2^{23}}$$

Two hosts are connected via packet switch with 10^7 bps links. Each link has a propagation delay of 20 μs. The switch begins forwarding a packet 35 μs after receiving it. If 10000 bits

Sr. No. _____ Date : _____

of data are to be transmitted between the two hosts using packet size of 5000 bits. calculate the time elapsed between the transmission of the first bit of data and the reception of the last bit of data in microsecond.



speed 10^7 bps

p.d = 20μs

(1) 500 + 500 + 20 + 35 + 20 = 1075μs

(2) 1075 + 500 = 1575μs

Algorithmic challenges in distributed system.

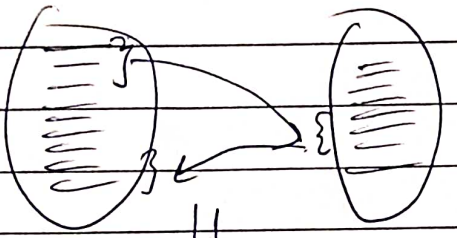
(1) Execution models (Designing)

Sr. No. _____ Data : _____

Interleaving model

↓ depends on physical clock

partial order model



↓

[logical clock]

Algorithm challanges in distributed system

(i) Designing execution model

(ii) Interleaving model
- It depends on physical clock or physical time model
- when there exists shared physical clock, we can expect total order of events in the system.

(iii) partial order model
- if there is no shared physical clock
- we can observe partial order between the events on different processor

Sr. No. _____ Date : _____

Logical time :

Logical time is relative time and eliminates the overheads of providing physical clock.

for applications where physical time is not required

Works

(i) capture the logic and inter process dependency
(ii) Track the relative progress

(iv) Automata model

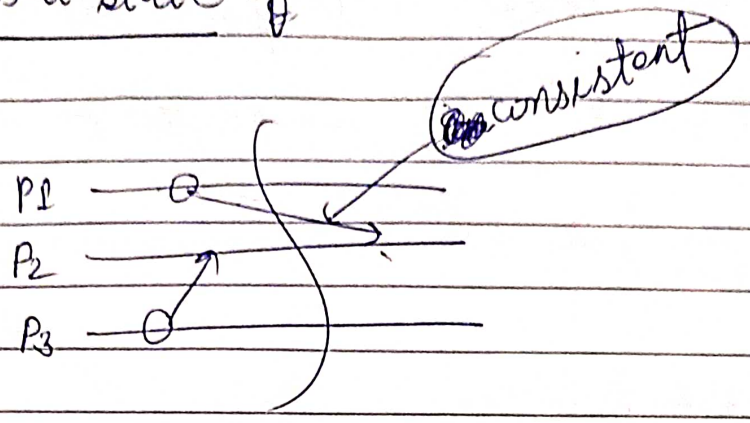
(v) TLA - Tomporal logic Action

- is a formal specification language developed by Lamport

It is used for modeling documentation designing and verification of program especially concurrent program.

Sr. No. _____ Date : _____

Global state of



Time and global state in distributed system.

→ synchronization and coordination mechanism

(i) Due to physical clock synchronization physical clock synchronization is tough

(ii) Leader election is not easy.

(iii) All process need to agree on which process will play the role of distinguish process

Leader is necessary for distributed algorithm

(i) it initiates some action.
like broadcast

(ii) it collects the state of system.

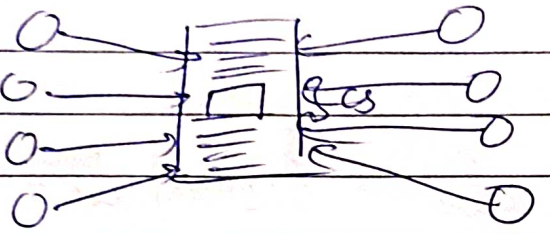
(iii) it regenerates a token that gets lost in the system.

(iv) Spanning tree

Sr. No. _____ Date : _____

(iii) mutual exclusion is very tough

(iv) Deadlock detection and resolution



→ Termination detection

find out global state of quiescence.

State of quiescence :

on distributed programs

multiple processes cooperate to perform some task and communicate via message passing to exchange information.

It is an important and well studied problem such pro in order to determine when the programs computation has completed or terminated or deadlock.

Sr. No. _____ Date: _____

this topic is challenging because each process has only local information but to show the problem it requires information about all the process.

JR language to identify (state of variance)

Group communication multicast and ordered message delivery

multicast ⇒ leader election
ordered message delivery ⇒ interleaving or total order due to lack of global clock.

→ monitoring distributed events and replicas.
→ Distributed program design or verification tools.
→ Debugging distributed programs.

→ Data Consistency.
→ Data Replicas
→ Data caching
→ World wide web (WWW)
This is the example of wide spread distributed system. with different interface

Sr. No. _____ Date : _____

→ Distributed shared memory Abstraction

A shared memory Abstraction simplifies the task of programmer because the programmer has to deal only with read and write operation and no message passing primitive.

But problem is that shared memory abstraction is very expensive

why expensive ?

(i) To design wait free algorithm

Wait free algorithm is defined as ability of the process to complete its execution irrespective of the actions of other processes, gained prominance in the design of Algorithm in order to control access to shared resources in shared memory abstraction.

→ mutual exclusion is also very challanging in distribution shared memory abstraction

→ Register construction ,,

→ Consistency model ,,

Sr. No. _____ Date : _____

Reliable and fault tolerant distributed system

- Consensus Algorithm
- Replication and Replica management
- Voting and quorum system
- Distributed database and distributed commit
- Self-stabilizing system.
- check-pointing and recovery Algorithm.
- Failure detectors
- ~~Load balancing~~

Load Balancing

performance ↑, throughput ↑

Application of distributed system

(i) mobile system
(ii) Sensor Networks
(iii) ubiquitous computing or pervasive computing
(iv) peer to peer computing
(v) multimedia
(vi) distributed datamining
(vii) Grid Computing
(viii) cloud "
(ix) Security in distributed system.

Sr. No. _____ Date _____

unit-2

clock - Atomic crystal

clocks

→ We need clocks for synchronization of processes
→ When we talk about process synchrony
- shared memory
- No global clock
- each processor has its own clock
- clock synchronization
(i) physical clock synchronization
(II) Logical clock synchronization.

पहले सूर्य के position के According समय calculate करते थे Astronomical metrics solar day
→ second clock was based on Atomic oscillator

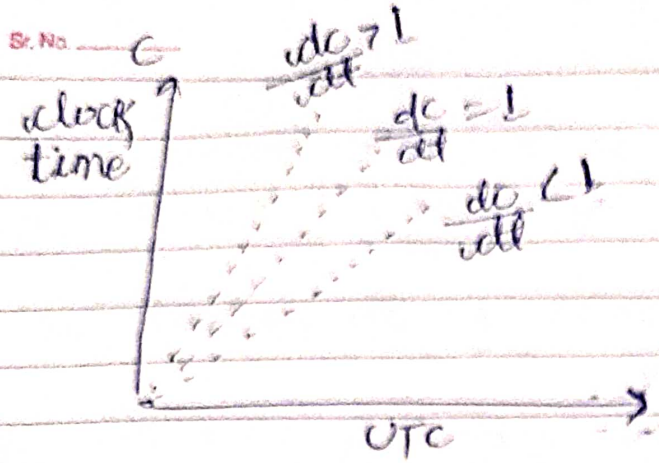
such a clock uses the property of atom and there accuracy is one part in 2013.
It is the most accurate time of clock.
→ cell phone clocks are synchronized using Atomic clock

coordinated universal time (UTC)
UTC broadcast on radio and receivers accurate 0.01 millisecond.

Drift Tolerance

$1-\rho \leq \frac{dc}{dt} \leq 1+\rho$

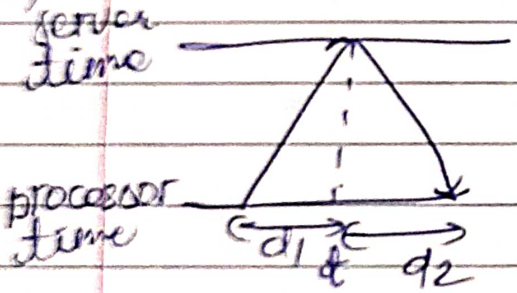
$\rho \Rightarrow$ maximum drift rate



Algorithms

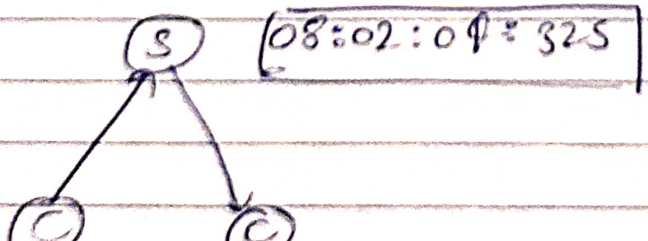
(i) Cristian Algorithm
(ii) Barkeley Algorithm
(iii) NTP (Network time protocol)
(iv) GPS (Global positioning systems)

(i)



$t + \frac{d_1+d_2}{2}$

Example



To 08:02:01:570 T1: 08:02:02:130

RTT $\simeq$ 280 ms $= \frac{d_1+d_2}{2}$

Correct time = 08:02:09:325 + 280

08:02:09:555

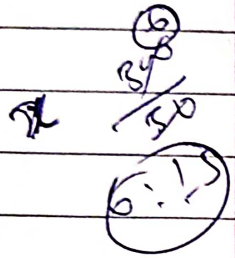
Sr. No. _____ Date : _____

Example 2

The clock in the clock tower in town A is broken it was repaired but now the clock needs to be set.

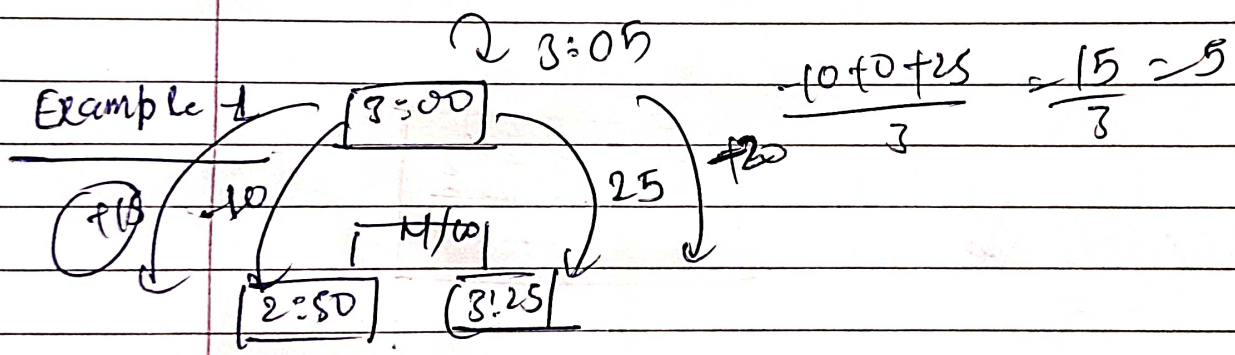
A train leaves for the Nearest town B, 150 miles away.

It returns 4 hours later with a report that the time according to the clock tower is 6:15 s = 50mph



8:15

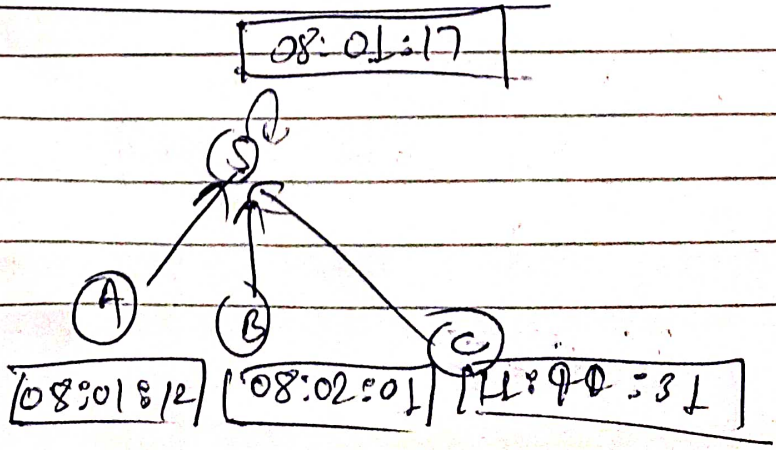
Barkeley Algorithm



$\frac{10+0+25}{3} = \frac{15}{3} = 5$

Example 2

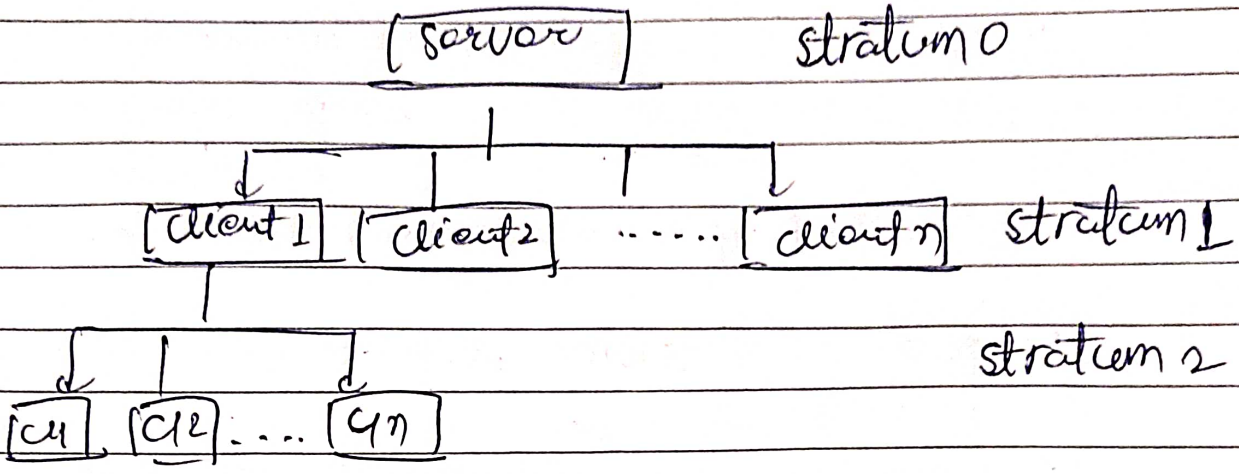
$5 + 44 + 43 + 58 \times 60 + 2 \times 60 \times 60$

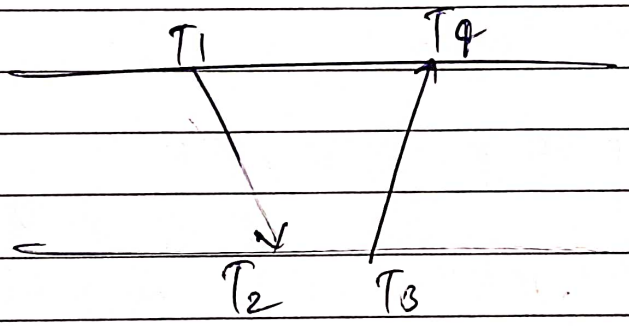


Gps - Global positioning system

Sr. No. _____ Date : _____

NTP Algorithm





$$\text{Delay} = \frac{[(T_4 - T_1) - (T_3 - T_2)]}{2}$$

$$\text{offset} = T_3 + \text{delay} - T_4 = \frac{(T_2 - T_1) + (T_3 - T_4)}{2}$$

$$\frac{2T_3 + T_4 - T_1 - T_3 + T_2 - 2T_4}{2}$$

$$\frac{T_3 + T_2 - T_1 - T_4}{2} = \frac{T_3 + T_2 - (T_1 + T_4)}{2}$$

Sr. No. _____ Date : _____

Logical clock

(i) Distributed system is loosely coupled Hardware and Tightly coupled software.

(ii) Distributed system is set of processes (No processor synchrony) connected by communication network.

(iii) Distributed system is facilitated by message passing.

(iv) Communication delay is finite and unpredictable.

(v) These processes do not have common memory and global clock.

(vi) No concept of physical clock

(vii) Communication medium may deliver messages out of order, messages may be lost, processor may fail, Communication link may go down.

program - program is a passive entity which contains the set of instructions required to perform a certain task.

process - (program under execution)
process is an active instance of the program which is started when the program is executed

Sr. No. _____ Date: _____

Distributed program is composed of a set of n asynchronous processes.

$P_1, P_2, P_3 \ldots\ldots P_n$ (by message passing over the communication Network.

Logical clock

Let c_{ij} ($C \equiv$ channel) is the communication channel

$i \leftarrow$ sender $\quad j \leftarrow$ receiver

$\downarrow \qquad\qquad \downarrow$

$P_i \qquad\qquad P_j$

m_{ij} is the message sent from P_i to P_j

Note: time $\equiv$ finite and unpredictable.

Global state of a program

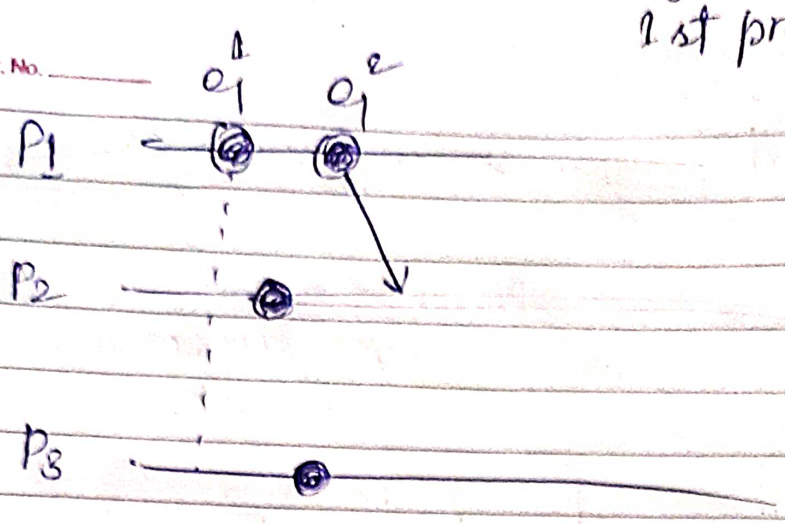
is composed of two states.

(i) state of processes
(ii) state of communication channel

Events (3 type of events) in distributed program

(i) Internal event
(ii) ~~(send event)~~ message send event
(iii) message receive event

पहला event जो भी run होगा it is considered as 1st process.



(space-time diagram)

e_1^x

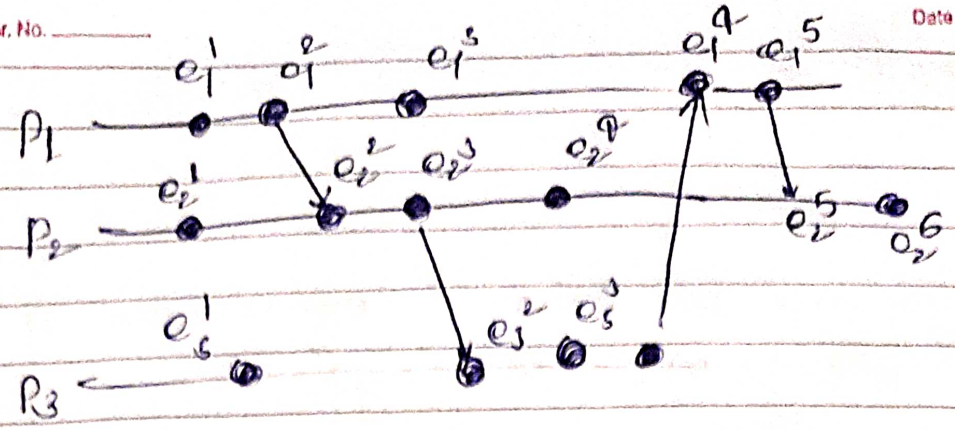
Causal precedence Relation

$\forall e_i^x, \forall e_j^y, e_i^x \longrightarrow e_j^y$ (⇒)

↓ happens before

$$\begin{cases} e_i^x \longrightarrow e_j^y \text{ i.e., } (i=j) \wedge (x<y) \\ \text{OR} \\ e_i^x \xrightarrow{msg} e_j^y \\ \text{OR} \\ \exists e_k^z \in H : e_i^x \longrightarrow e_k^z \wedge e_k^z \longrightarrow e_j^y \end{cases}$$

Sr. No. _____ Date : _____



$\forall e_i^x, \forall e_j^y \in H$,

$e_i^x \rightarrow e_j^y \iff$

- $e_i^x \rightarrow e_j^y$ i.e $(i=j) \wedge (x \wedge y)$
- OR
- $e_i^x \xrightarrow{msg} e_j^y$
- OR
- $\exists e_k^z \in H$:

$e_i^x \rightarrow e_k^z \wedge e_k^z \rightarrow e_j$

question 1

$e_1^1 \rightarrow e_3^3$ ✓

question 4

$e_2^4 \rightarrow e_3^1$ ✗

question 2

$e_5^3 \rightarrow e_2^6$ ✓

question 5

$e_3^3 \rightarrow e_2^4$ ✗

question 3

$e_1^3 \rightarrow e_2^3$ ✗

Sr. No. _____ Date : _____

question 6

Causal ordering (CO)

For any two messages m_{ij} and m_{kj}, if

send (m_{ij}) → send (m_{kj})

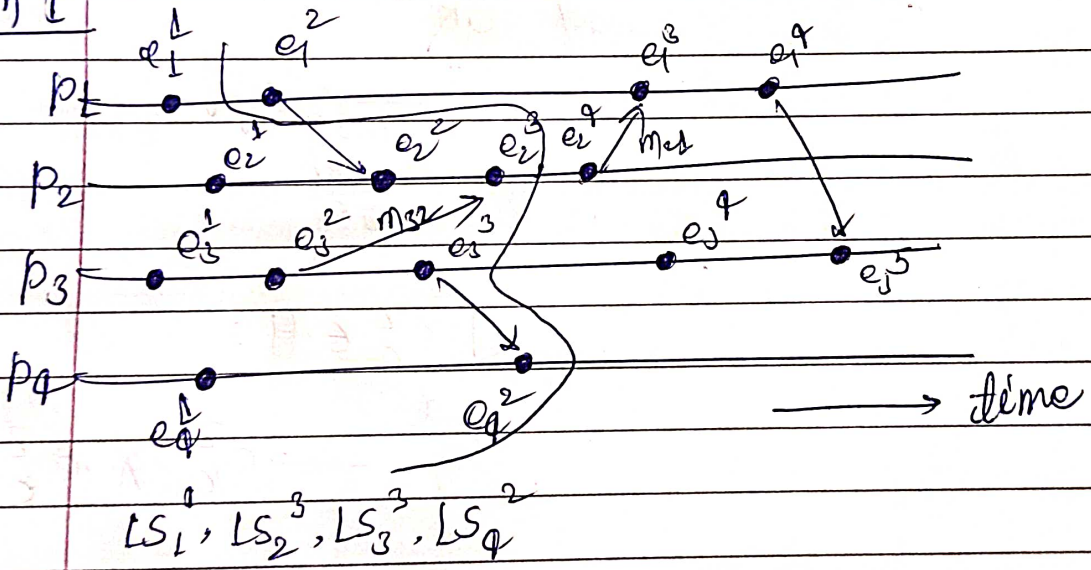
then

rec (m_{ij}) → rec (m_{kj})

CO ⊂ fifo ⊂ Non-fifo

CO ⊂ fifo ⊂ Non-fifo

question 1



Sr. No. _____ Date : _____

Global state

→ A message cannot be received if it was not sent. It is known as consistent state.

→ A message was received but it wasn't sent. It is known as inconsistent state.

$$GS = \left\{ \bigcup_i LS_i^{x}, \bigcup_{j,k} SC_{jk}^{y_j, z_k} \right\}$$

LS ≡ Local state
SC ≡ state of channel

will be consistent if and only if $\forall m_{ij}$ such that send of $(m_{ij}) \notin LS_i^x \Rightarrow m_{ij} \notin SC_{ij}^{x_i, y_j} \wedge receive(m_{ij}) \notin LS_j^{y_i})$

question 1:

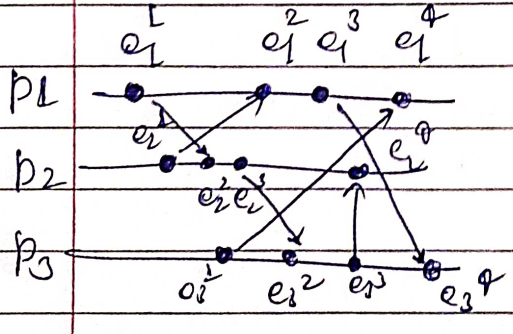
$\{LS_1^1, LS_2^3, LS_3^3, LS_4^2\}$

- Inconsistent

question 2:

$\{LS_1^2, LS_2^4, LS_3^4, LS_4^2\}$

- Consistant

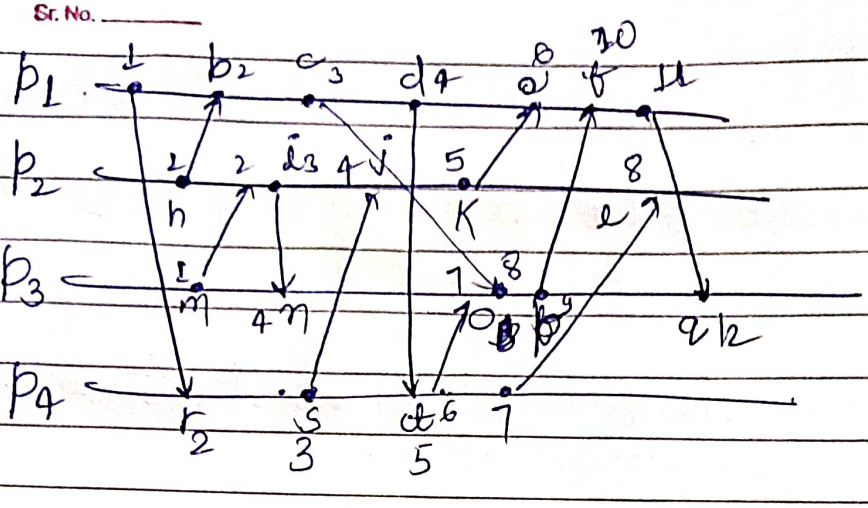


(i) LS_1^1, LS_2^2, LS_3^1 - consistant

(ii) LS_1^2, LS_2^3, LS_3^2 - Consistant

(iii) LS_1^3, LS_2^4, LS_3^2 - Inconsistant

Sr. No. _____ Date : _____



Logical time

$C : H \longrightarrow T$

H ≡ set of events

clock time stamp of e_i before e_j

$e_i \longrightarrow e_j \Rightarrow C(e_i) \longrightarrow C(e_j)$

↓ happens before

Rules of logical time

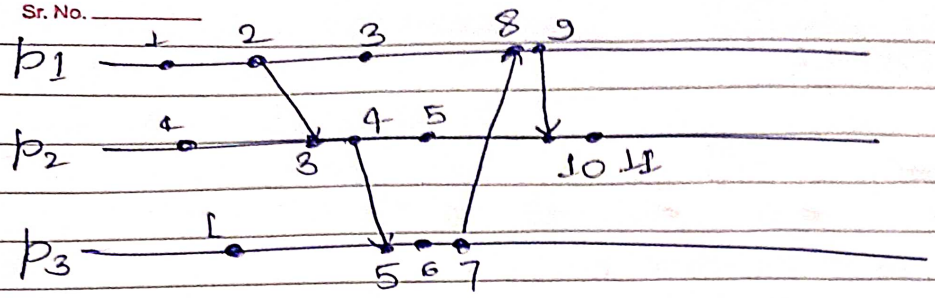
(i) Governs how the local logical clock is updated by a process when it executes an event

(ii) It governs how a process updates its global logical clock to update the view of global time and global process.

Rule 1 $C_i := C_i + d$ where $d > 0$

Lamport's clock

Sr. No. _____ Date : _____



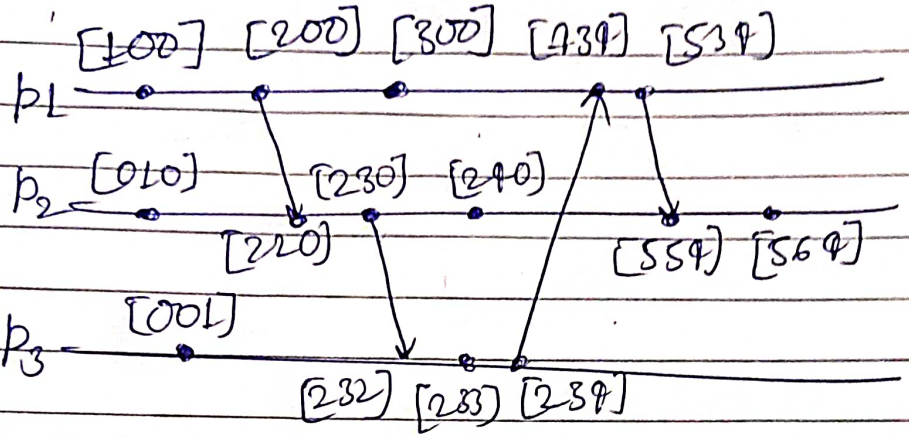
Rules

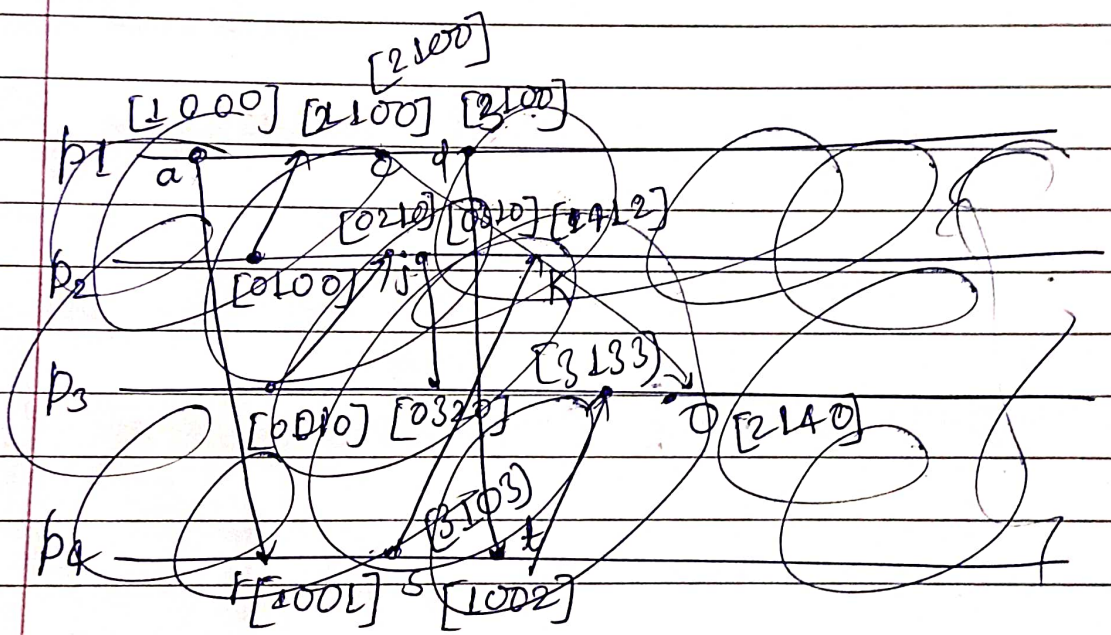
(i) $C_j := max(C_j, C_{msg})$
(ii) execute R_1 (rule 1)
(iii) deliver the message

[3100]

Sr. No. _____ Date : _____

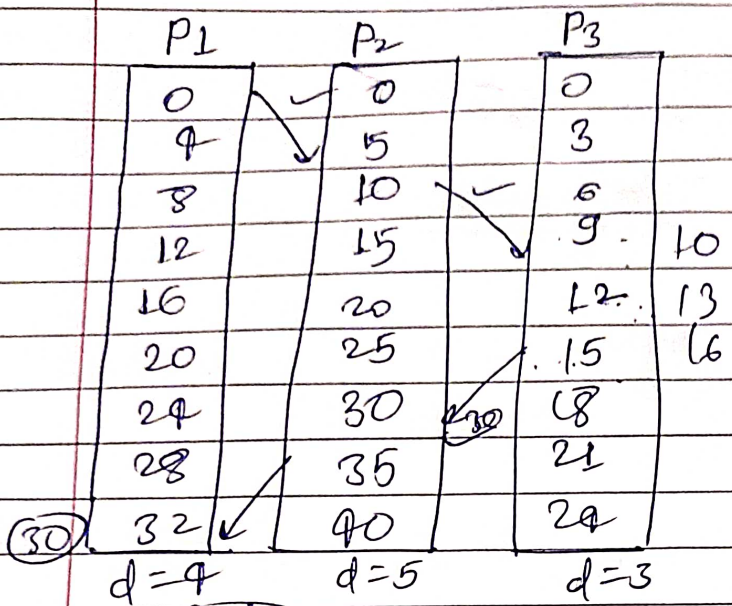
vector clock



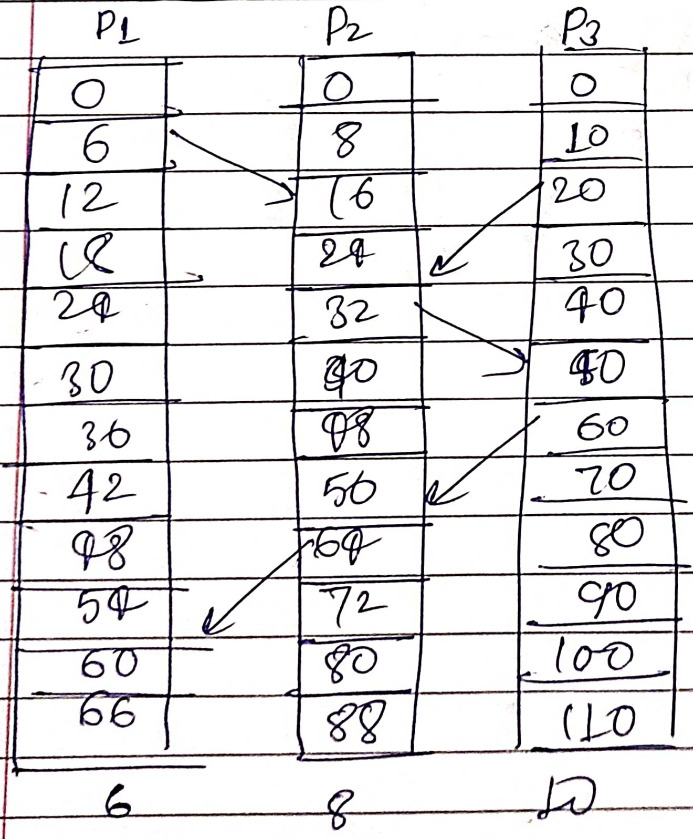


जब P_1 P_2 P_3 का d अलग होगा तो receive वाले में d Add कर देंगे

Sr. No. _____ Date _____



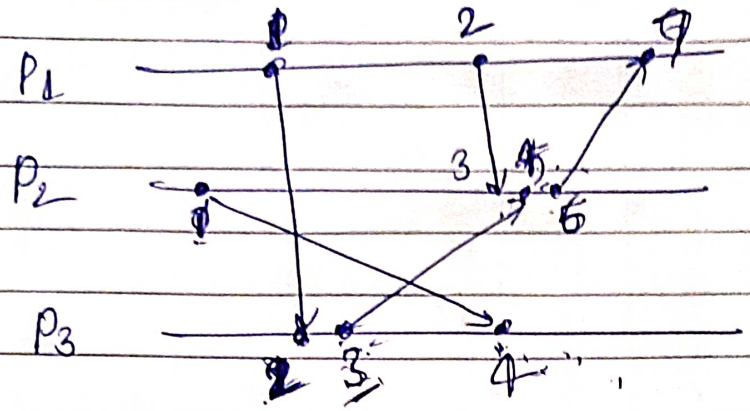
vector



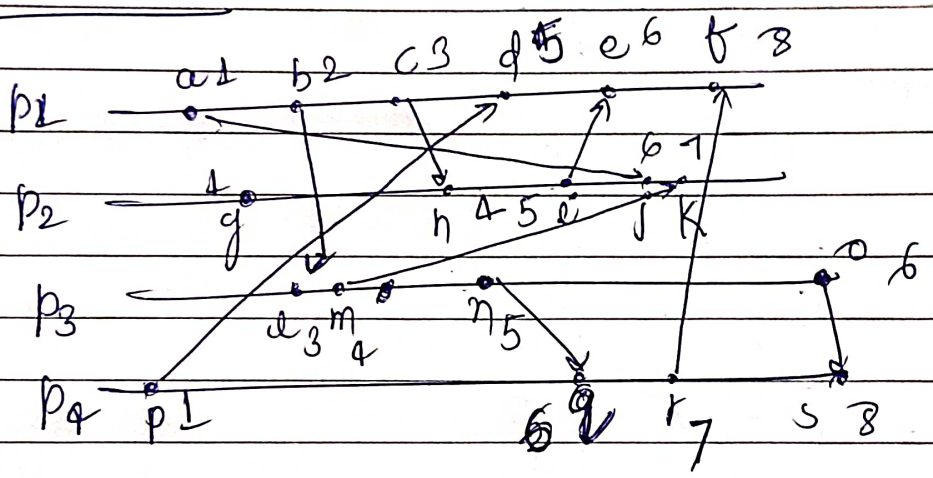
(68)

Sr. No. _____ Date : _____

question 3



question 4



Sr. No. _____ Lamport's ~~Lamport~~ clock Date : _____

Rule R_1 — local logical time

This rule governs how the logical clock is updated by a process when it execute internal event.

Rule R_2 — global logical time

This rule governs how a process update its view of global time and global progress

Algo

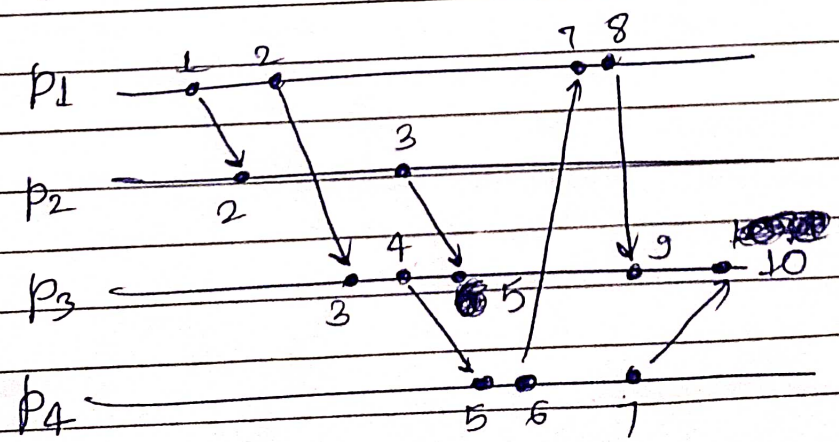
Before executing an event process p_i executes

$\Rightarrow R_1 \Rightarrow$ $C_i := C_i + d$ where $(d > 0)$

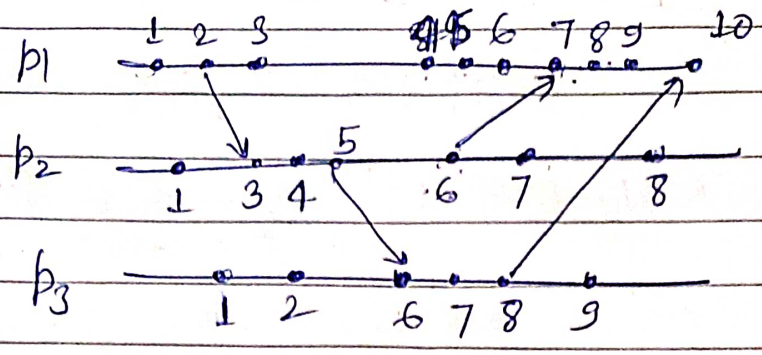
Each message piggybacks the clock value of its sender at sending time when a process p_i receives a message with timestamp C_{msg} it will execute

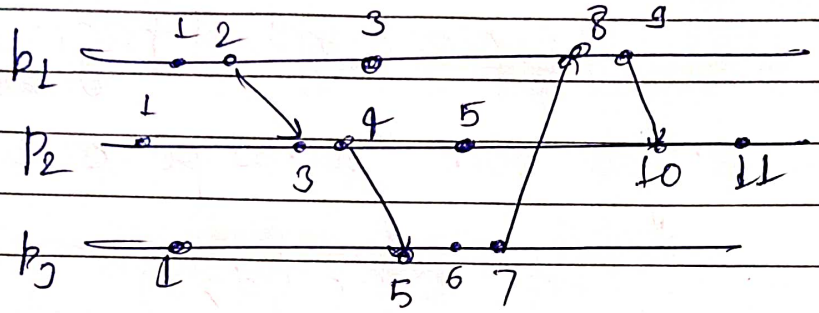
R_2 : 1. $C_i := \max(C_i, C_{msg})$
2. execute R_1
3. deliver

Ex



Sr. No. _____ Date : _____





properties of Lamport's clock

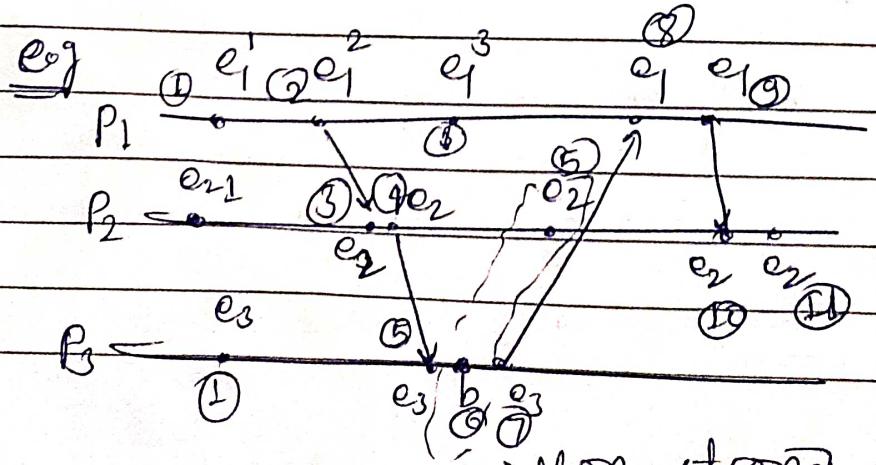
(i) Consistency property

for e_i and e_j

if $e_i \rightarrow e_j \Rightarrow C(e_i) < C(e_j)$

But Not strong consistant

$C(e_i) < C(e_j) \Rightarrow e_i \rightarrow e_j$



→ Non strong consistency

Sr. No. _____ Date : _____

(ii) Total ordering → Total ordering is fulfilled in Lamport clock.

If total order is consistent with the causality relation.

(iii) Event counting

(iv) No strong consistency

Vector clock - fulfill the strong consistancy

R1: Before executing an events process p_i updates its logical clock as follows:

$$Vt_i[i] := Vt_i[i] + d \quad (d > 0)$$

R2:

1. update the global logical time as follows

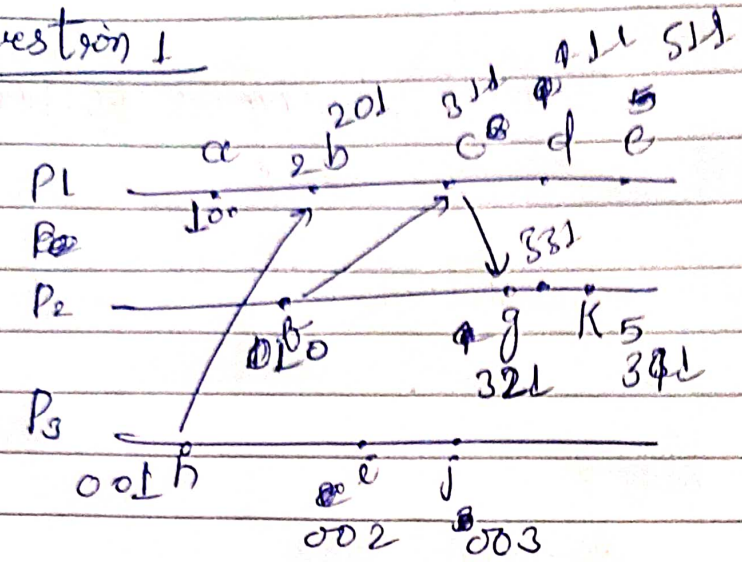
$1 \leq K \leq n$ such that $Vt_i[K] := max(Vt_i[K], Vt[K]$

2. Execute R_1

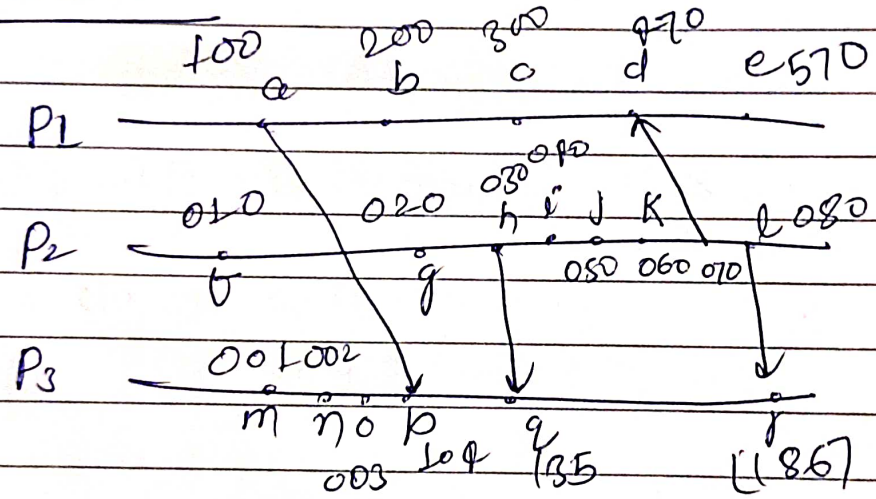
3. Deliver the message

Sr. No. _____ Date : _____

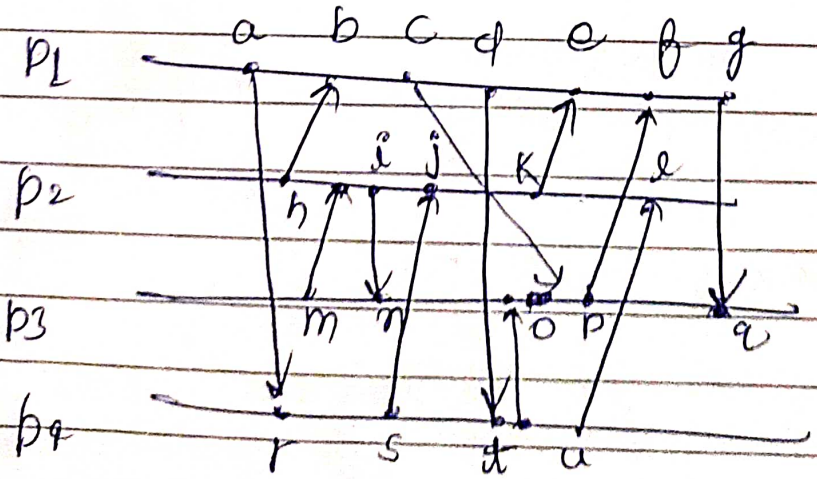
question 1



question 2

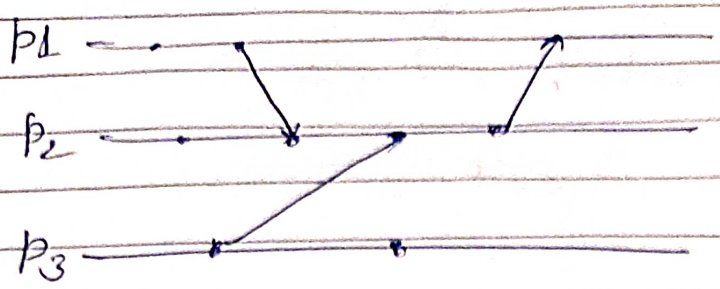


question 3

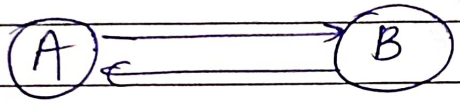


Sr. No. _____ Date : _____

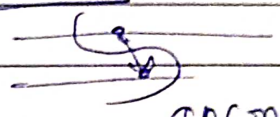
Q4:



Global state

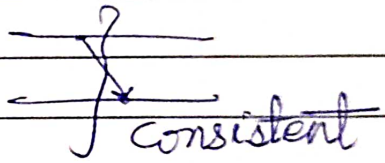


Case 1



Inconsistant

Case 2

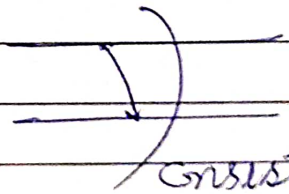


Consistent

Q: If the state is consistent but not transitless

Q: If the state is transitless — not possible

Case 3



Consistent & Transitless

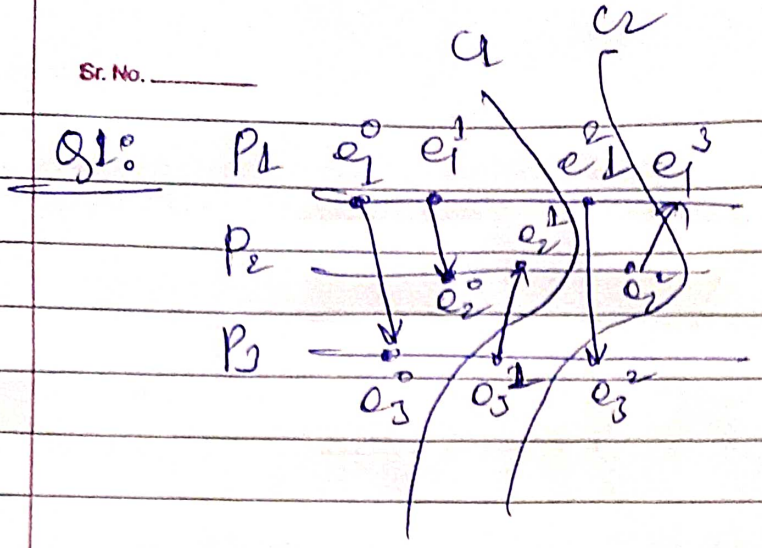
All the consistent transitless are consistant but not all the consistent are transitless.

Snap shot Algorithm

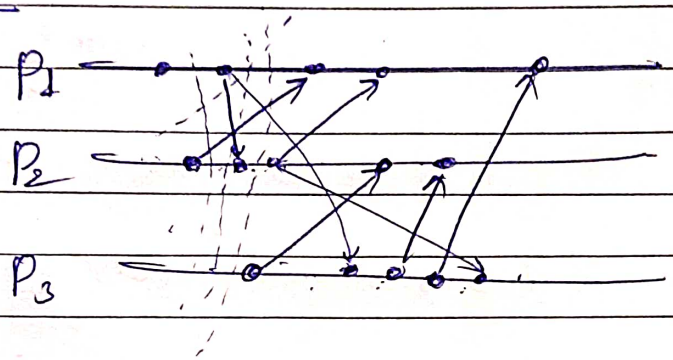
Chandy-Lamport algorithm

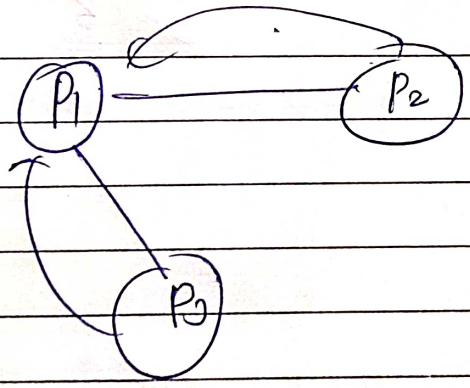
Sr. No. _____ Date : _____

Q1:



Q2:





Sr. No. _____ Date : _____

Chandy Lamport Algorithm

marker Receiving rule for process p_i

on p_i's receipt of a marker message over channel c
if (p_i has not yet recorded its state)
    it records its process state now;
    it records the state of channel c as empty;
    turns on recording of messages arriving over other incoming channel;
else
{ p_i records the state of c as the set of messages it has received over c since it saved its state
}

marker sending rule for process p_i

After p_i has recorded its state for each outgoing channel c;
p_i sends one marker message over c (before it sends any other message over c);

Ex. No. _____ Date: _____

Marker sending rule for process i

- process i records its state
- for each outgoing channel c on which a marker has not been sent, i sends a marker along c before i sends any message along c.

Marker Receiving rule for process j

on receiving a marker along c

if j has not recorded its state

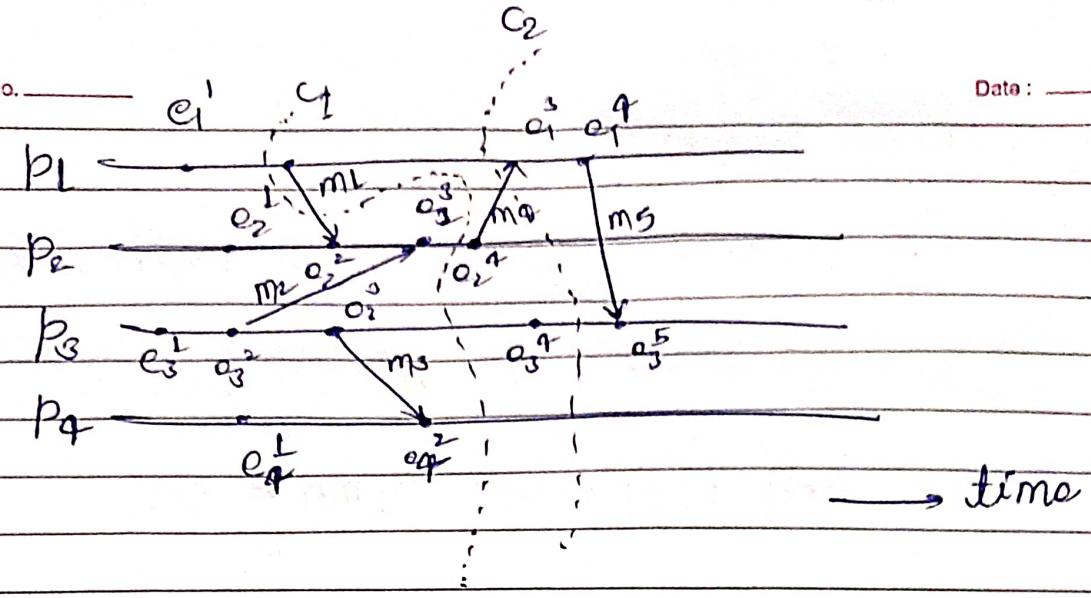
{
Record the state of c as the empty set
follow the marker sending rule
}

else

{
Record the state of c as the set of messages received along c after j's state was recorded before j received the marker.
}

Hello Hello

Sr. No. _____ Date : _____



There are two issues in this problem

(i) How to distinguish between the messages to be recorded in the snapshot from those that is not to be recorded.

Notes:

Any message that is sent by a process before recording its snapshot must be recorded in the global snapshot.

Any message that is sent by a process after recording its snapshot must not be recorded in global snapshot.

(ii) How to determine the instant when the process takes its snapshots

Note: A process p_i must record its snapshot before processing a message m_{ij} that was sent by process p_i after recording its snapshot.

Sr. No. _____ Date : _____

$GS = \{ \cup_i LS_i, \cup_{ij} SC_{ij} \}$

is consistent (in terms of mathomical form)

(1) if and only if it satisfy two conditions

C1: $send(m_{ij}) \in LS_i \Rightarrow m_{ij} \in SC_{ij}$

$\oplus$

$rec(m_{ij}) \in LS_j$

C2: $send(m_{ij}) \notin LS_i \Rightarrow m_{ij} \notin SC_{ij}$

$\wedge$

$rec(m_{ij}) \notin LS_j$

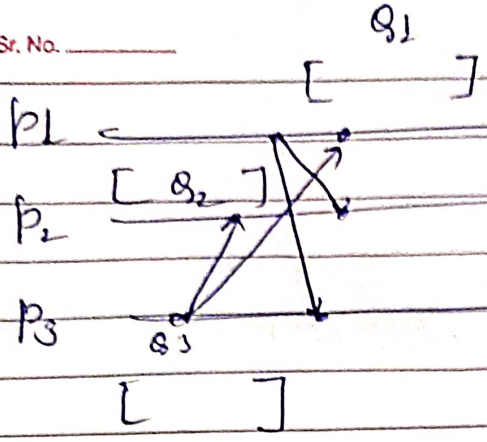
mutual exclusion

The system with processor synchrony ← uses semaphore or monitor to get the critical section ← [shared resource] [P1] [P2]

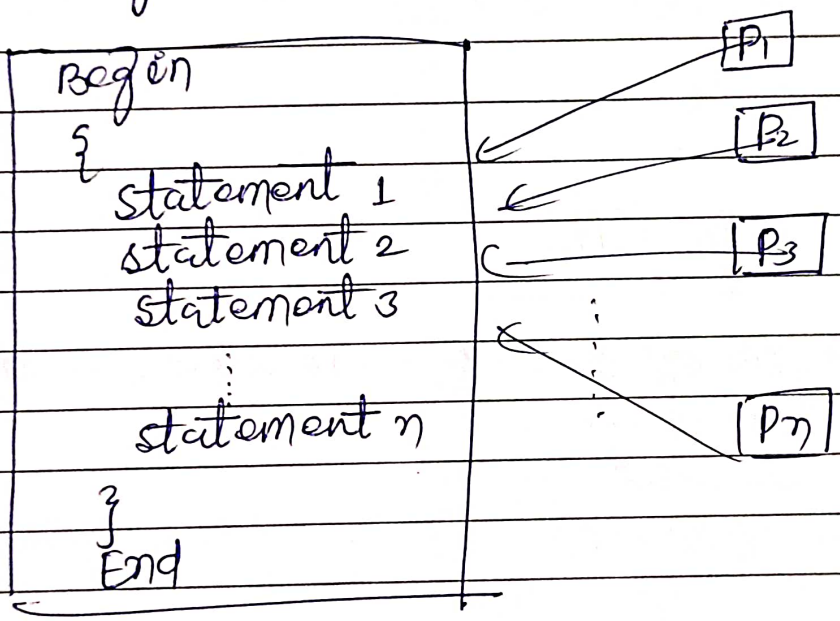
Distributed systems ← ? ? ← [NO shared Resource] [P1] [P2] [Pn]

process synchrony
clock + memory
sharable

Sr. No. _____ Date _____



program



3 types of messages

- Request
- Reply
- Release

Sr. No. _____ Date : _____

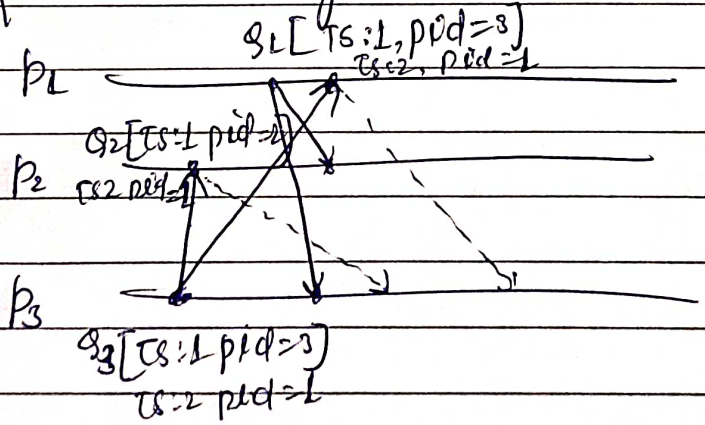
Two types of algorithm

(i) Token based
- Suzuki - Kasami algorithm
- Raymond's tree algorithm

(ii) Non-Token based
- Lamport algorithm
- Ricart - Agrawala algorithm
- Maekawa's Algorithm

Lamport Algorithm

Three types of messages are used in this Algo.
(i) Request message
(ii) Reply message
(iii) Release message



Suppose P_3 wants to get the critical section. P_3 will wait until it doesn't get any reply from other processor to get the critical section.

Sr. No. _____ Date : _____

before getting the reply P_1 also want to get the critical section.

The processor which has less time stamp will have greater priority to get the critical section

when all the processor send the reply message all processor get to know which processors is having the less time stamp.

After completing the execution the processor which have got the critical section will release the critical section.

Total messages = $3(n-1)$

Requirements to enter the critical system

(i) Its time stamp is on the top of the queue
(ii) It should receive Reply message from other processes.

Due to unnecessarily a lot of messages to be broadcasted the algorithm suffers from $3(n-1)$ message complexity.

that which is solving by next algorithm Ricart algorithm).

Sr. No. _____ Date : _____

Ricart Agarawala Algorithm

Algorithm

Step 1: When a site s_i wants to enter CS, it sends a time-stamped Request messages to all other sites.

Step 2: When site s_j receives Request message from site s_i it sends Reply to site s_i

if and only if

(i) site s_j is neither requesting nor executing critical section.

(ii) In case site s_j is requesting time stamping of s_i is smaller than s_j otherwise request is deferred by site s_j

Step 3

Execution of CS

↳ site s_i enters CS if it has received Reply message from all other sites

Release of CS

↳ upon exiting site s_i sends Reply message to all the deferred request

Sr. No. _____ Date : _____

Complexity

$2(N-1)$

Maekawa's Algorithm for mutual exclusion

Quorum based Approach

What is quorum?

→ if a site want to enter into the critical section there is no need of permission from every site taking other site. but it needs to take permission from a subset of site.

↳ Non time stamped Approach

↳ Three types of messages are used

(i) Request
(ii) Reply
(iii) Release

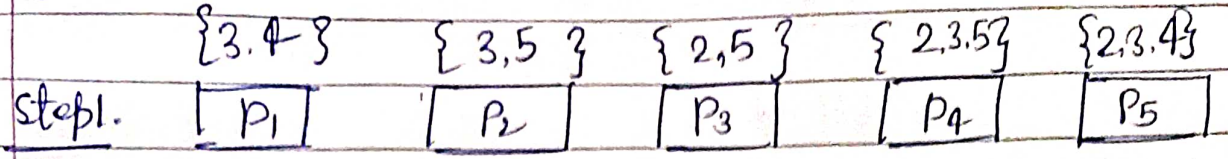
Assumptions

(i) every process should belong to some group
(ii) if any process wants into the critical section it will sent the request message to its member only
(iii) (i) (suppose group i) $G_i \cap G_j \neq \phi$
(ii) The process should be part of any group
(iii) single process cannot be considered as a group

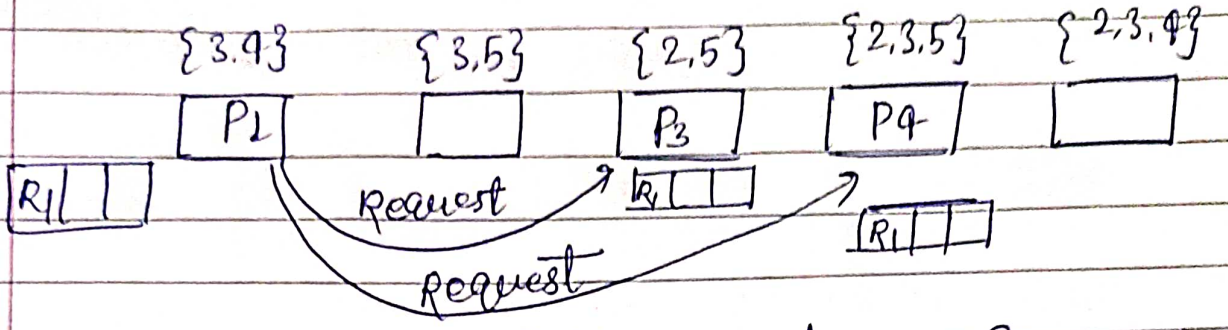
Request set

Sr. No. _____ Date : _____

Ex



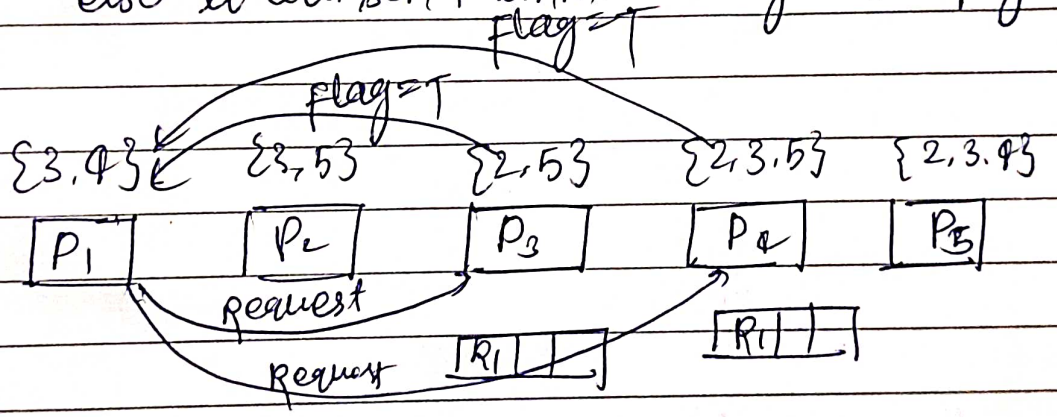
Step2. suppose P_1 wants to enter critical section



Step3: After getting the request message P_3 and P_4 updates and check their queues already

If there are other messages in queues, it will not send reply.

else it will send immediately the reply

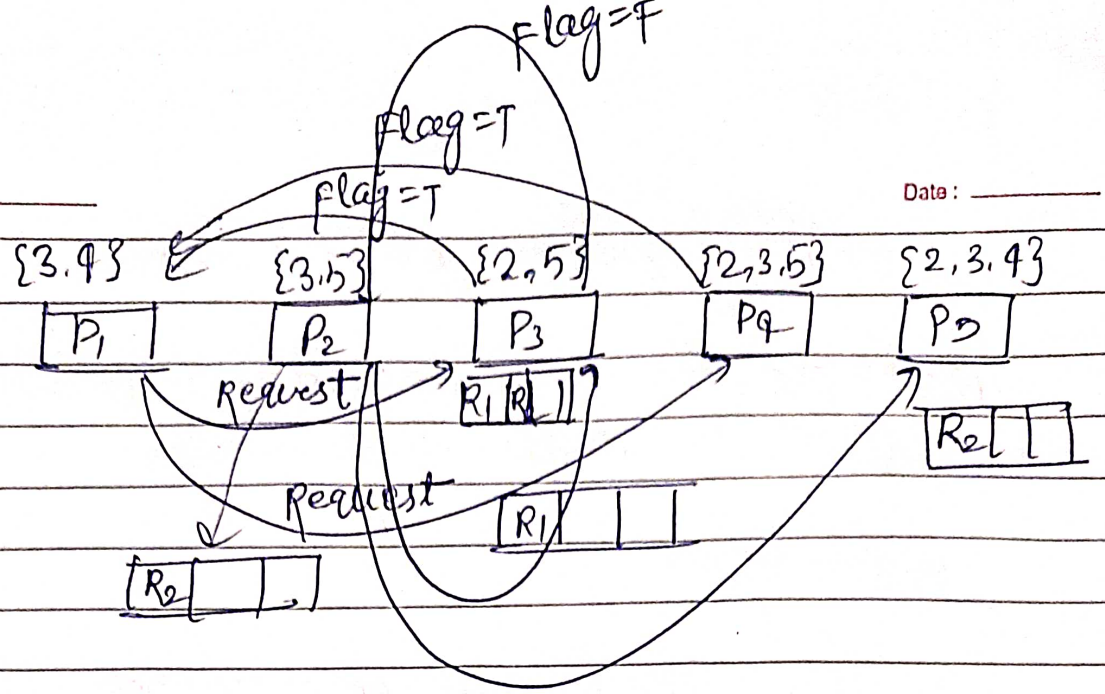


Step4: when P_1 gets flag = True Reply from other members of the group it will immediately enter into the critical section

Step 5: suppose P_2 wants to enter the critical section

Pradeep

Sr. No. _____ Date : _____



P_2 will not get enter into the critical section because P_3 has not sent reply section because there is already a message request in the queue.

After Execution of P_1. P_1 will sent message to its group members.

P_2 will enter into critical section when P_3 return Flag = T to the P_1

Requesting the critical section

(i) A site requests access to critical section by sending request messages to all its group members

(ii) on receipt of request messages the site members will sent a reply if it has not sent a reply to any other member OR it has not got the release message

~~Executing~~

Sr. No. _____ Date : _____

Executing critical section

(I) A site enters into CS only if it receives a reply from other member of its group.

Releasing CS

(I) After the execution of CS is over, site sends Release(0) message to all its group members.

(II) When a site receives Release message, it sends a reply to the first pending request and delete the entry from its request queue.