Qtext :

Consider the space-time diagram shown in Fig. 1. The scalar clock values of the events ***a***, ***b***, ***c*** and ***d*** are 3, 2, 2 and 4 respectively. Identify the scalar timestamps of the rest of the events and the scalar timestamps of the messages exchanged among the 4 processes. Use **R1** and **R2** for updating the scalar clocks. Assume increment value d = 2. [10]

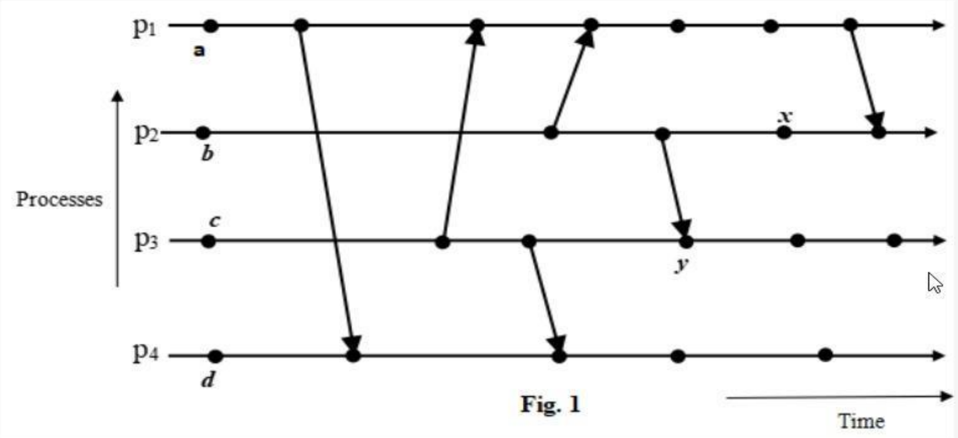


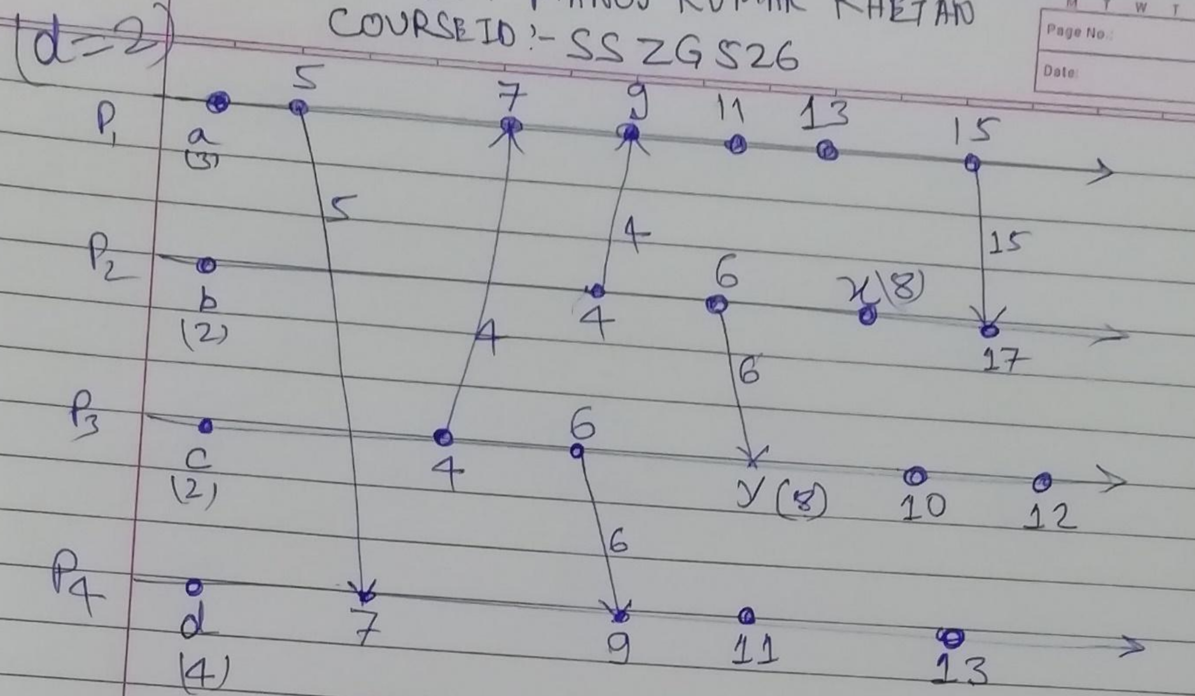
Fig. 1

Are the scalar timestamps of events ***x*** and ***y*** same? If yes, then with proper justification, show how you can totally order ***x*** and ***y***? [1 + 1 = 2]

Ans :

BITS ID : 2021MT13294
NAME :- MANOJ KUMAR KHETAN
COURSEID :- SS ZG526

Mid term exa



Scalar clocks of each event is shown above.

⇒ Yes timestamp of x and Y is same as 8.

⇒ The total ordering relation ≺ on two event X and Y with timestamp (h, i) and (k, j)

$X \prec Y \iff (h < k \text{ or } (h = k \text{ and } i < j))$

$h = 8$

$k = 8$

$i = 2$

$j = 3$

$x \prec Y$ as we has $(2 < 3)$ as second condition applied.

Barsha Ma'am feedback : 1st part answer is correct, 2nd part answer is not clear. You have to write the order in tuple form.

Correction for part 2
x has time stamp as (8, 2) and y as time stamp as (8, 3). As (8, 2) < (8, 3) since 2 < 3.
Hence x < y.

Qtext :

Consider the timing diagram of Fig. 3. The timing diagram shows 2 processes P1 and P2 of a distributed system. C_{12} is the communication channel from P1 to P2 and C_{21} is the communication channel from P2 to P1. P1 has 2 local variables A and B. P2 has 2 local variables X and Y. The dashed vertical lines denote the different time instants of this timing diagram. Initially, at time instant t_1, the values of the local variables of P1 are – A = 30, B = 9 and the values of the local variables at P2 are – X = 50, Y = 20. Between time instants t_1 and t_2, P1 sends message m1 to P2. The content of m1 is "change X to 60". P2 receives m1 between t_2 and t_3 and accordingly changes the value of X. P2 sends message m2 to P1 between t_1 and t_2. The content of m2 is "change B to 29" and P1 receives this message between t_2 and t_3. P1 changes the value of B accordingly. Between t_3 and t_4, P1 sends message m3 to P2. The content of m3 is "change Y to 90". P2 receives m3 between t_4 and t_5 and accordingly changes the value of Y. The messages m1, m2 and m3 are shown using bold arrows in the diagram. Assume that the Chandy-Lamport algorithm is executed on this distributed system for recording the global snapshot. P1 initiates the snapshot recording algorithm by sending a marker to P2 after time instant t_4. P2 receives this marker after time instant t_5 and sends a marker to P1. P1 receives this marker after t_6. The sending and the receiving of markers are shown using dotted arrows in the Figure. Describe how the Chandy-Lamport algorithm records the local states of P1 and P2 and the states of the channels C_{12} and C_{21}. Note that the values of the individual local variables constitute the state of each process. [6]

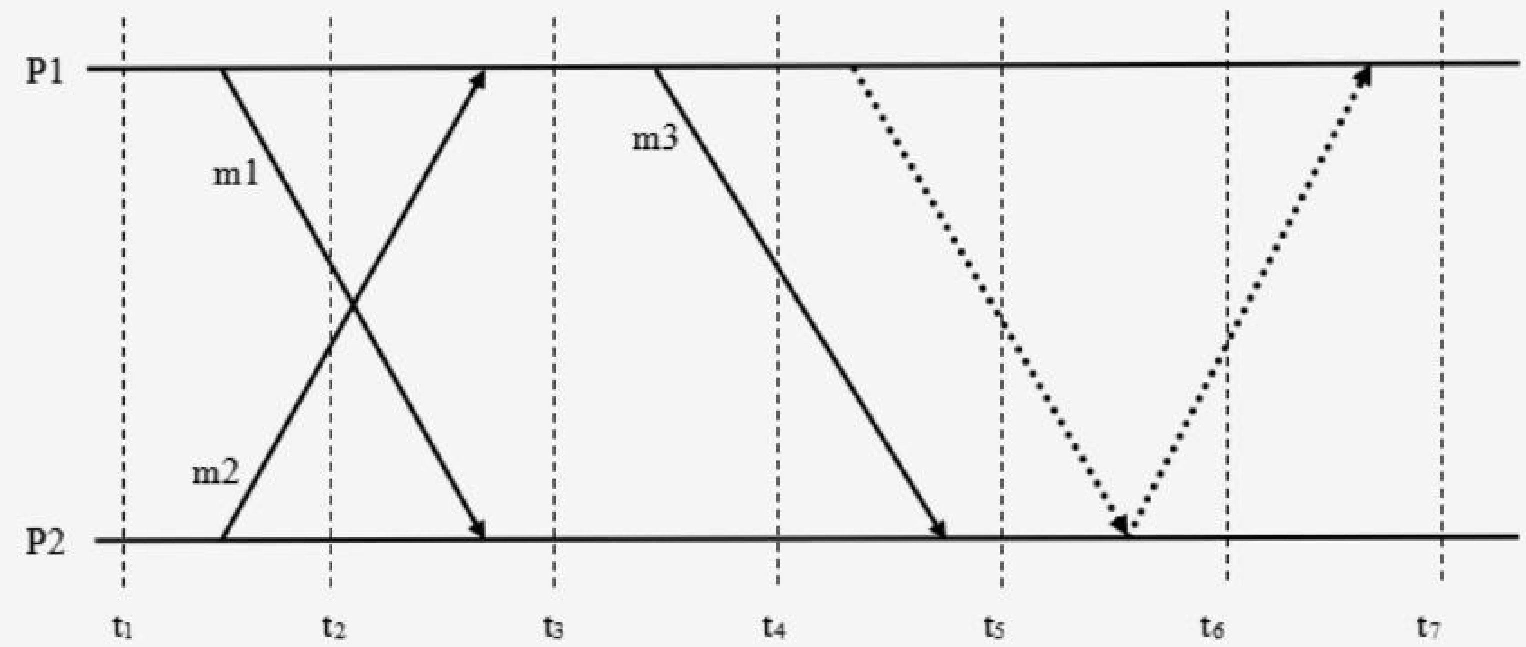
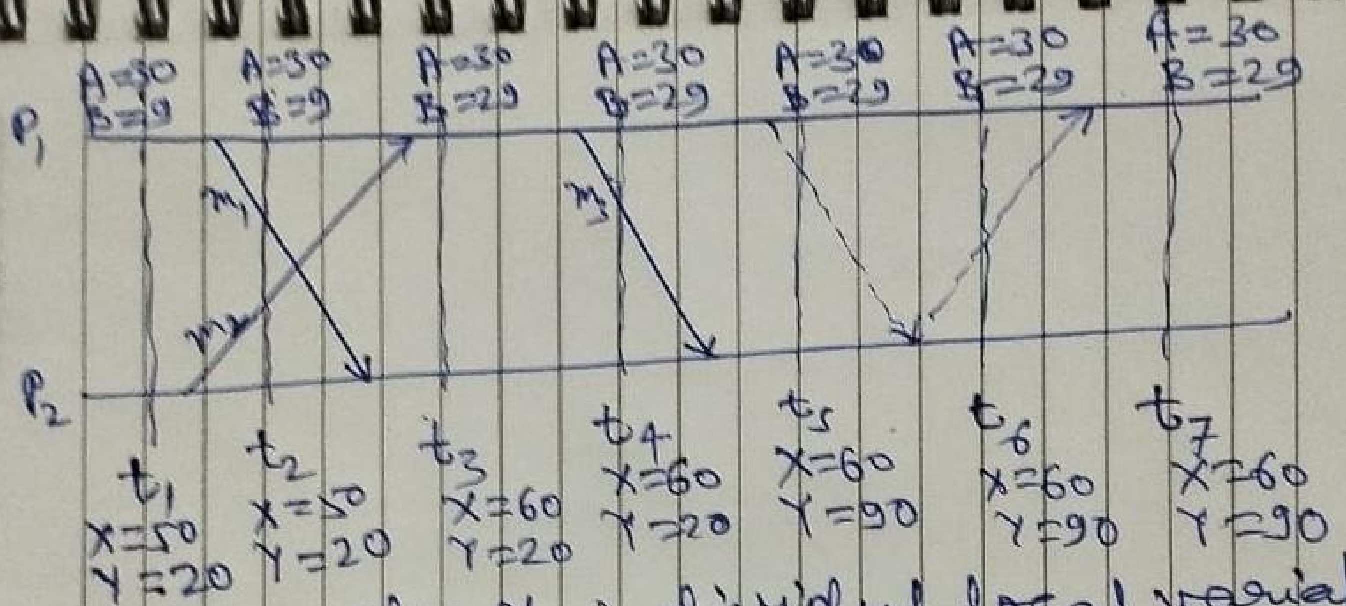


Fig. 3

Ans :

Barsha Ma'am feedback : The sum of the variables are not to be computed. The state of each process is the values

Redone the solutions of problem as below:



⇒ As per given value of individual local variable constitute the state of each process.

⇒ As per Chandy Lamport Algorithm, process P_1 record the state/~~of~~ snapshot as LSP_1 $\{A=30, B=29\}$ before sending marker message, later sends marker to P_2

⇒ On receiving marker process P_2 records the state/snapshot later sends marker to P_1. $LSP_2 = \{X=60, Y=90\}$

⇒ State of channel C_{12} as $SC_{12} = \emptyset$ → as there both messages m_1, m_3 were already reached P_2

⇒ State of channel C_{21} as $SC_{21} = \emptyset$ → as m_2 already reached to P_1

Qtext :

Consider a distributed system consisting of 5 processes as depicted in Fig. 4. The sites have not been shown in the diagram. You are not required to consider any site for answering the questions.

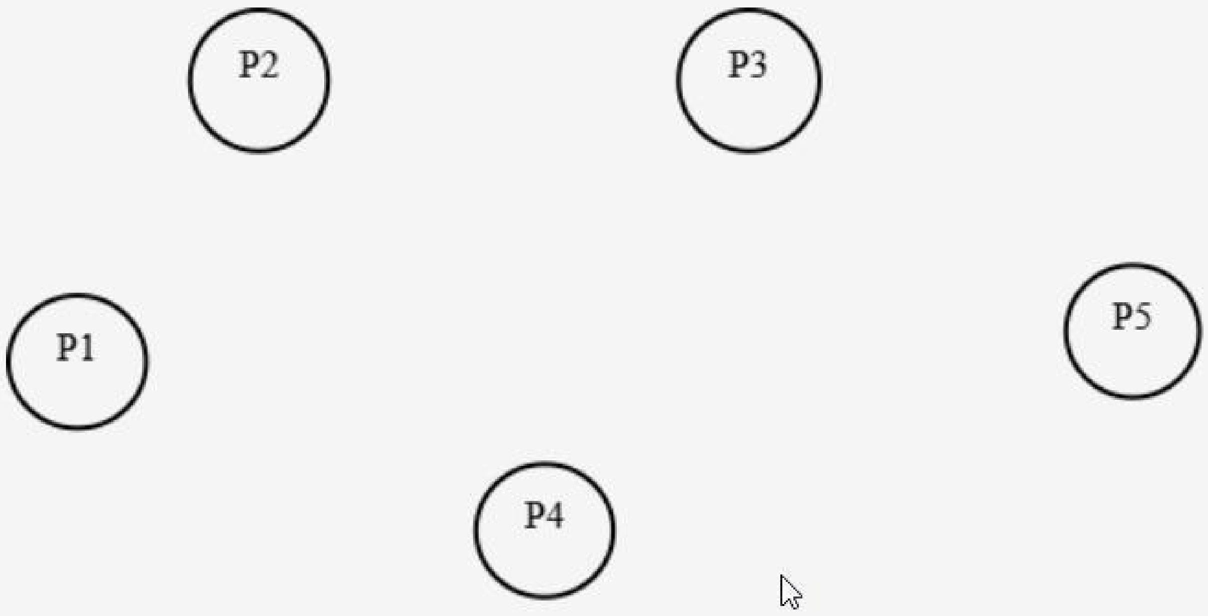


Fig. 4

The following dependencies exist among the processes:

- P1 is blocked and is waiting for P4 to release some resource
- P4 is blocked and is waiting for P3 to release some resource
- P3 is blocked and is waiting for P2 to release some resource
- P2 is blocked and is waiting for P1 to release some resource
- P5 is blocked and is waiting for P3 to release some resource

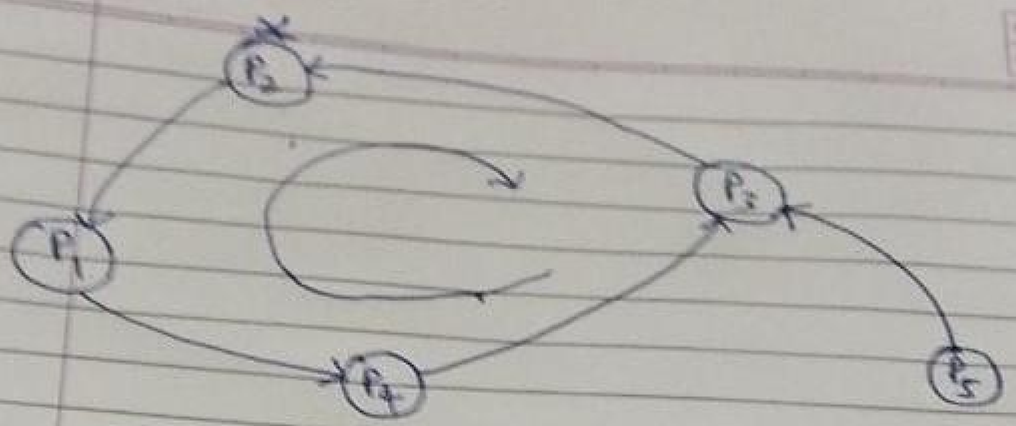
Draw the WFG for the above scenario. [1]

Execute Chandy-Misra-Haas algorithm for the OR model on this distributed system to determine whether any deadlock exists in the system. Assume P2 initiates the algorithm. Note that the deadlock is to be detected by using the algorithm and not by any other means. [9]

Ans :

Barsha Ma'am feedback : You have to write the steps of the CMH algorithm. You have to mention the messages exchanged in text, not in diagram. Follow the lecture content.

Redone the solutions of problem as below:



Since process $P_2 \to P_1 \to P_4 \to P_3$ cycle is present as well as process P_5 is dependent on P_3. So Knot is present in the WFG.

presence of Knot represents a deadlock in ~~pee~~ CMH OR model.

Step1 Since P_2 is initiating the CMH OR Algorithm

$Num_{P_2} = 1$

$|DS_{P_2}| = 1$

$wait_{P_2} = T$

Query Sent $\cancel{q(P_2, P_2, P_1)}$ $q(2,2,1)$ as engaged query

Step2 at P_1

$Num_{P_1} = 1$

$|DS_{P_1}| = 1$

$wait_{P_1} = T$

Query Sent $q(2,1,4)$

Step3 at P_4

$Num_{P_4} = 1$

$|DS_{P_4}| = 1$

$wait_{P_4} = T$

Engaged Query Sent $q(2,4,3)$

Step4: Since P_5 is not part of knot, So it will not send engaged Query

Step5: at P_3

$Num_{P_5} = 1$

$|DS_{P_5}| = 1$

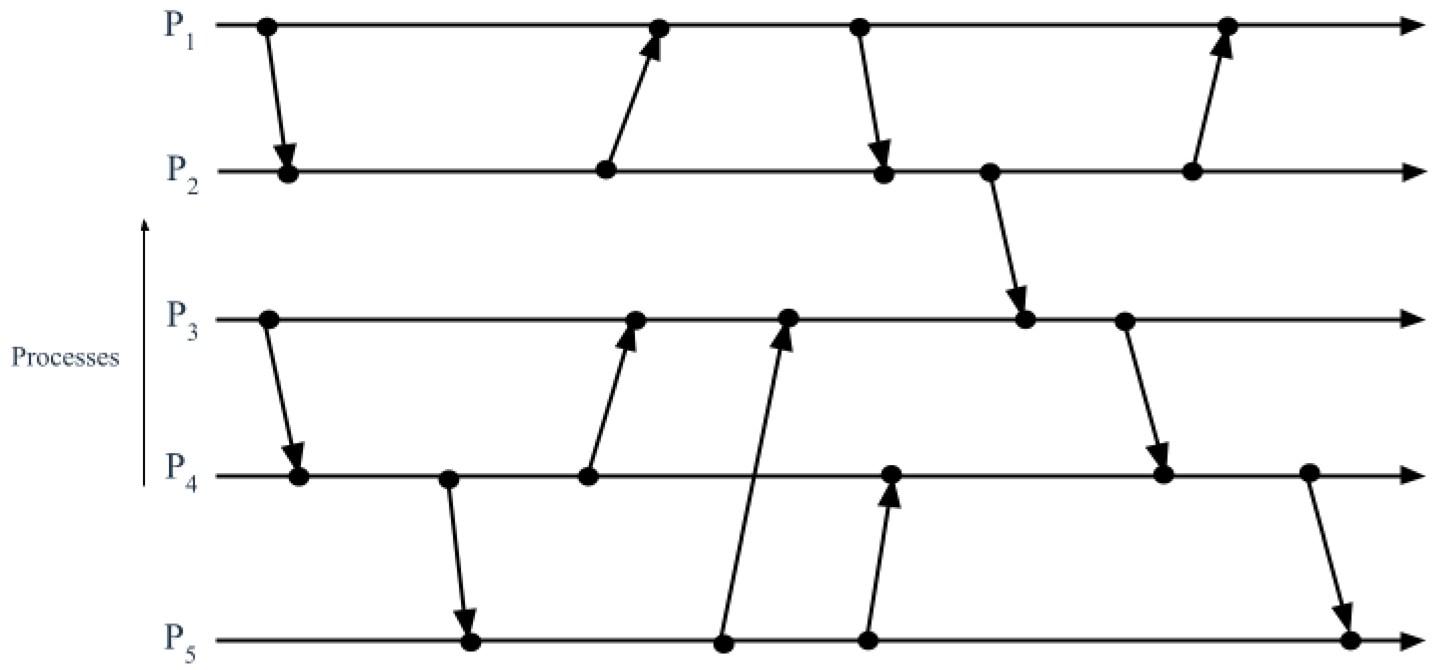
$wait_{P_5} = T$

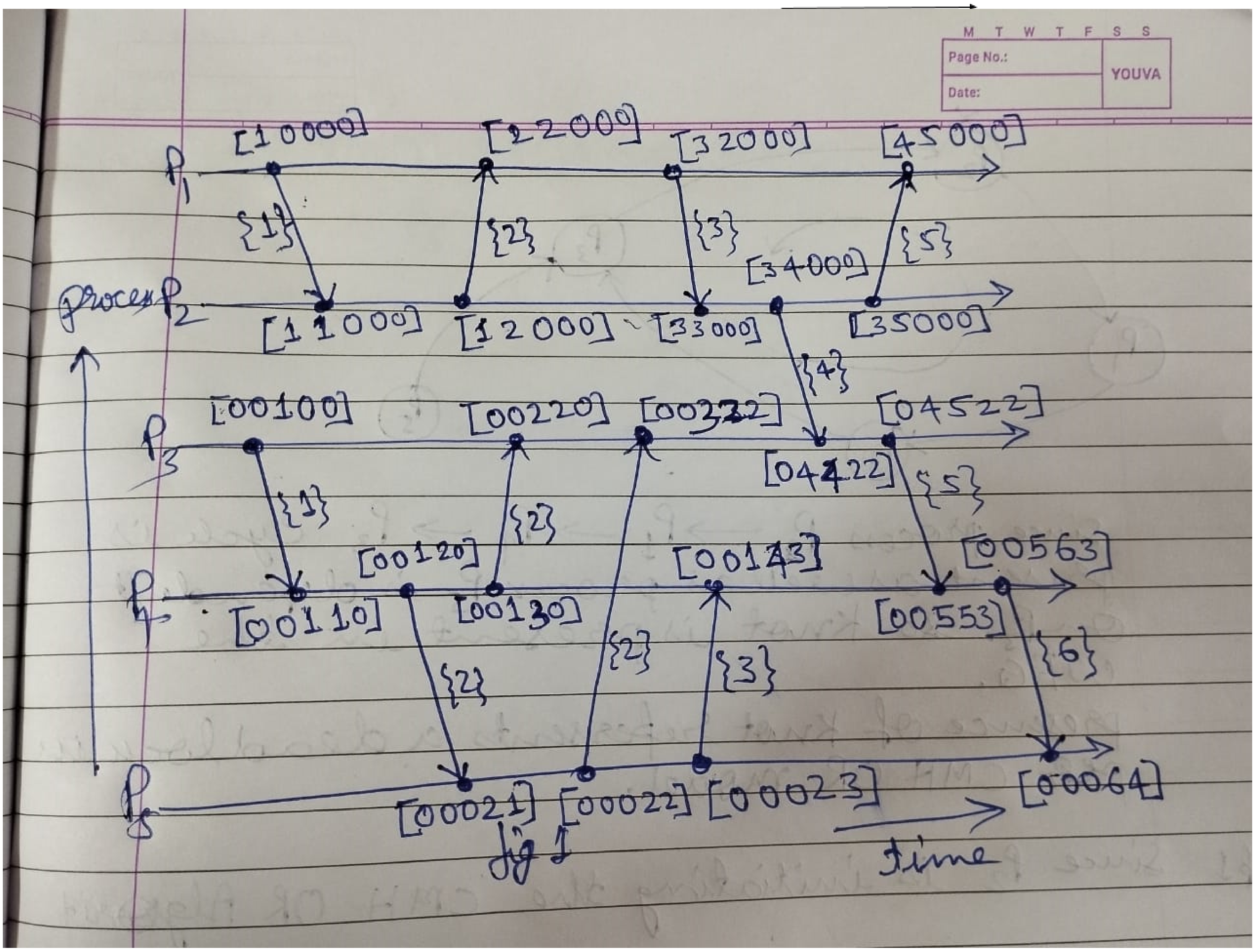
Query Sent $q(2,3,2)$ Engaged query.

Step 6: P_2 will send reply message to P_3 as it send Engaged Query

Q.1Set.(A)

Apply Fowler-Zwaenepoel's direct-dependency technique to determine the progress of vector clocks at each of the 5 processes shown in Fig. 1 and also the contents of the messages exchanged among them. Assume that initially the vector clocks of each of the 5 processes is [0 0 0 0 0] and the increment value d = 1. [9]





Q.2Set.(A) Suppose a distributed system consists of 8 processes – P1, P2, P3, P4, P5, P6, P7 and P8. Consider the following 4 events –

- e_1 has occurred on P2 and has a height of 7
- e_2 has occurred on P5 and has a height of 10
- e_3 has occurred on P6 and has a height of 7
- e_4 has occurred on P8 and has a height of 5

You do not have to consider the other events of this distributed system.

Calculate the scalar timestamps of e_2 and e_4. Assume d = 1 and the scalar clocks of each of the processes before the occurrence of any events in the distributed system was 0.

How will you totally order e_1 and e_3? Clearly mention the timestamp representations of the 2 events and the reason for your manner of total ordering. [1 + 2 = 3]

Ans:

e2 has height as 10 and value of d =1 , so scalar timestamp : 10 +1 = 11
e4 has height as 5 and value of d =1 , so scalar timestamp : 5 +1 = 6

e1 has height as 7 and value of d =1 , so scalar timestamp : 7 +1 = 8
e3 has height as 7 and value of d =1 , so scalar timestamp : 7 +1 = 8

So time stamp for e1 is (8, 1) and e3 is (8, 3).

Since (8, 1) < (8, 3) as per total ordering second as below:

$$x \prec y \Leftrightarrow (h < k \textbf{ or } (h = k \textbf{ and } i < j))$$

So e1 < e3

Q 3Set(A) Consider the spanning tree shown in Fig. 2. Each node of this tree stores an integer value. For each node, the node label corresponds to the stored integer value. The objective is to determine the sum of all the integer values stored in the tree. Describe the step-by-step procedure by which the root node notifies all the nodes of the spanning tree to initiate the procedure for computation of the sum of all integers and how the sum is computed over all the integers. Write the steps of the algorithm as is applicable to the given scenario. DO NOT ANNOTATE IN THE DIAGRAM. If you annotate in the diagram without writing the steps of the algorithm, NO MARKS WILL BE AWARDED.

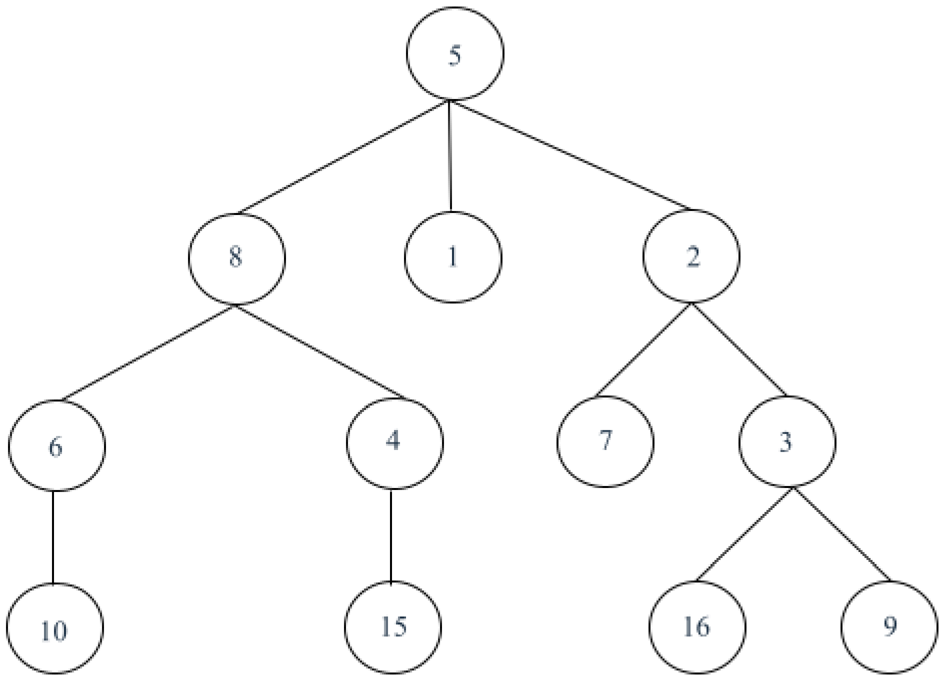
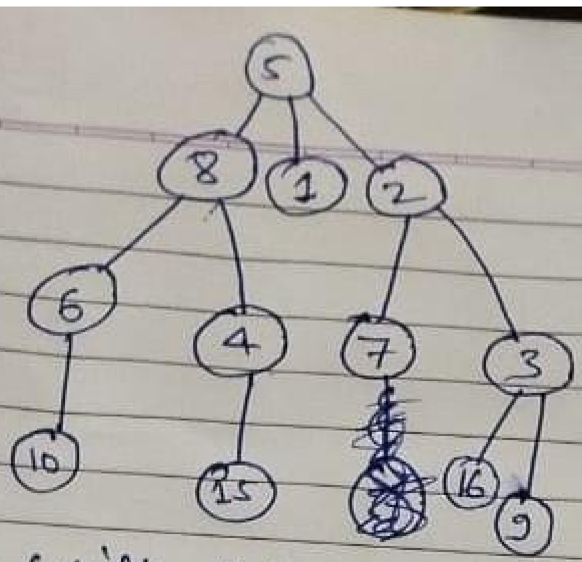


Fig. 2



Broadcast

root node is with stored value as 5 & labelled as 5

⇒ root node 5 will send notify to children (to compute sum) 8, 1, 2. Node 8, 2 has further children, so will send notification.

⇒ 8 node send notificatication (to compute sum) to node 6, 4, node 6 further sends notification to 10. node 4 send further to node 15.

⇒ 2 node send notification to 7 and 3. node 3 has children, so send notification further.

⇒ 3 node send notification to node 16 and 9.

Converge cast

⇒ node 16 and 9 sends calculated sum as 16 and 9 to parent 3. node 3 calculates sum as 3+16+9 =28 and send values to parent 2.

⇒ node 2 get values from 7 as value 7 & 3 as 28 Calculates sum as 2+28+7=37 and sends to parent 5.

⇒ node 6 gets value as 10 from 10 & sends 6+10=16 value to parent node 8.

⇒ node 4 gets values as 15 from 15 & sends 15+4=19 value to parent node 8.

⇒ node 8 gets values as 19 from 4 and 16 from 6 & send value 8+19+16=43 to parent 5.

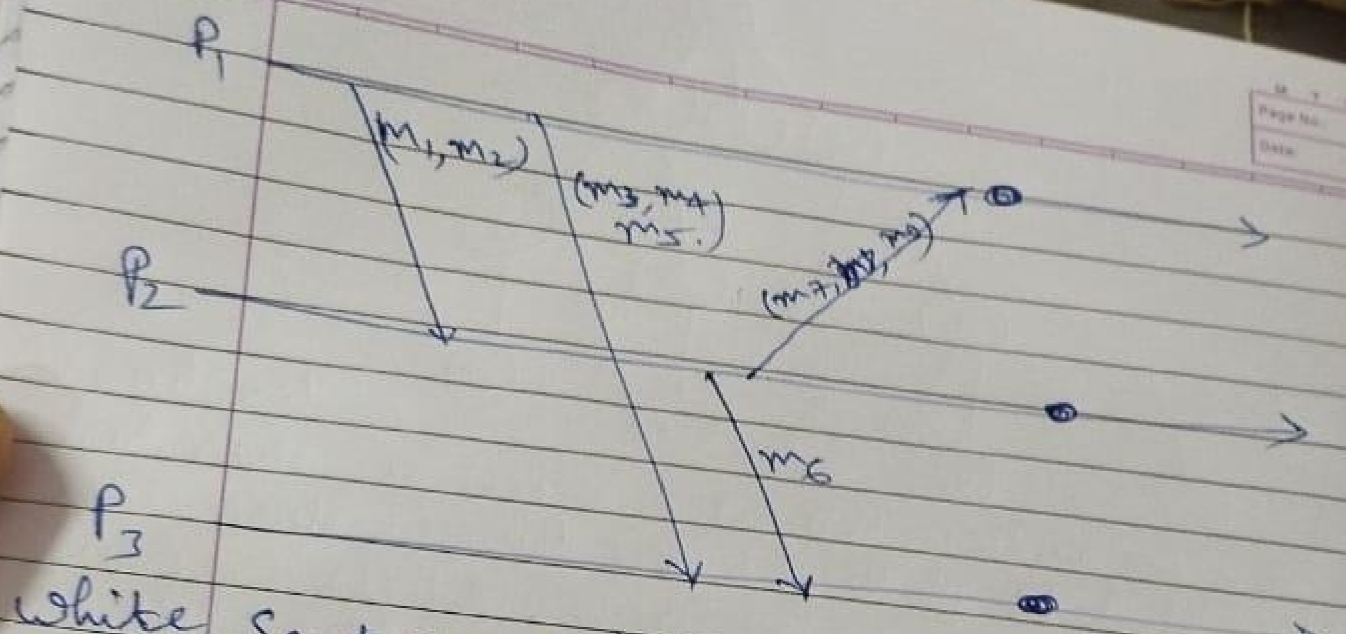
root node 5 receives values as 43 from node 8 and 1 from 1, 37 from 2.

root node calculates final sum as $5+43+1+37 = 86$.

Q.4Set.(A) Consider a distributed system consisting of 3 processes – P1, P2 and P3. The communication channel from P1 to P2 is denoted as C_{12}, the communication channel from P2 to P1 is denoted as C_{21}, the communication channel from P1 to P3 is denoted as C_{13}, the communication channel from P3 to P1 is denoted as C_{31}, the communication channel from P2 to P3 is denoted as C_{23}, and the communication channel from P3 to P2 is denoted as C_{32}. Now consider the following history of white messages exchanged.

- P1 has sent m_1 and m_2 on C_{12}
- P1 has sent m_3, m_4 and m_5 on C_{13}
- P2 has sent m_6 on C_{23}
- P2 has sent m_7, m_8 and m_9 on C_{21}
- P2 has received m_2 on C_{12}
- P3 has received m_4 and m_5 on C_{13}
- P3 has received m_6 on C_{23}
- P1 has received m_9 on C_{21}

Determine the states of the channels C_{12}, C_{13}, C_{23} and C_{21} using your knowledge of the Lai-Yang algorithm for global snapshot recording. You have to properly show how you are determining each of the channel states along with appropriate justification. Assume that P1, P2 and P3 have turned red and have recorded their respective local states. Note that all the messages mentioned above are white messages and were exchanged among the processes before they turned red. The communication channels are non-FIFO in nature.



white messages Sent on C_{12} by $P_1 = (m_1, m_2)$

white message received on C_{12} by $P_2 = (m_2)$

$$SC_{12} = (m_1, m_2) - (m_2) = (m_1)$$

white messages Sent on C_{13} by $P_1 = \{m_3, m_4, m_5\}$

white messages received C_{13} by $P_3 = \{m_4, m_5\}$

$$SC_{13} = \{m_3, m_4, m_5\} - \{m_4, m_5\} = \{m_3\}$$

white message Sent on C_{23} by $P_2 = \{m_6\}$

white message received on C_{23} by $P_3 = \{m_6\}$

$$SC_{23} = \{m_6\} - \{m_6\} = \emptyset$$

white message Sent on C_{21} by $P_2 = \{m_7, m_8, m_9\}$

white messages received on C_{21} by $P_1 = \{m_9\}$

$$SC_{21} = \{m_7, m_8, m_9\} - \{m_9\} = \{m_7, m_8\}$$

Q.5Set.(A) For a synchronous system consisting of 5 Byzantine/faulty processes, what is the minimum total number of processes that should be present in the system so that the Byzantine agreement problem can be solved? Do not simply write the answer. Show your calculations.

If the recursive Byzantine Agreement Tree algorithm is executed on the above synchronous distributed system, then calculate the number of rounds/phases for which the algorithm will execute. Do not simply write the answer. Show your calculations.

Based upon the minimum total number of processes that you had calculated above, determine the total number of messages exchanged in round number 3. [1 + 1 + 1 = 3]

Ans :

Page No.:
Date:

⇒ No. of faulty/ Byzantine processes = 5
formula for min Total nodes $N = 3m + 1$ (where m is faulty/Byzan Proc)

minimum no. of process $(N) = 3 \times 5 + 1 = 16$.

⇒ for 5 faulty/Byzantine processes recursive calls will be OM(5) to OM(0)
So, the number of rounds $= 5 + 1 = 6$

⇒ Total number of messages exchange in round 3 $= (N-1)(N-2)(N-3)$ messages
$= (16-1)(16-2)(16-3)$
$= 15 \times 14 \times 13 = 2730$ messages

Q.6Set.(A) Consider a distributed system consisting of 11 processes as depicted in Fig. 3.

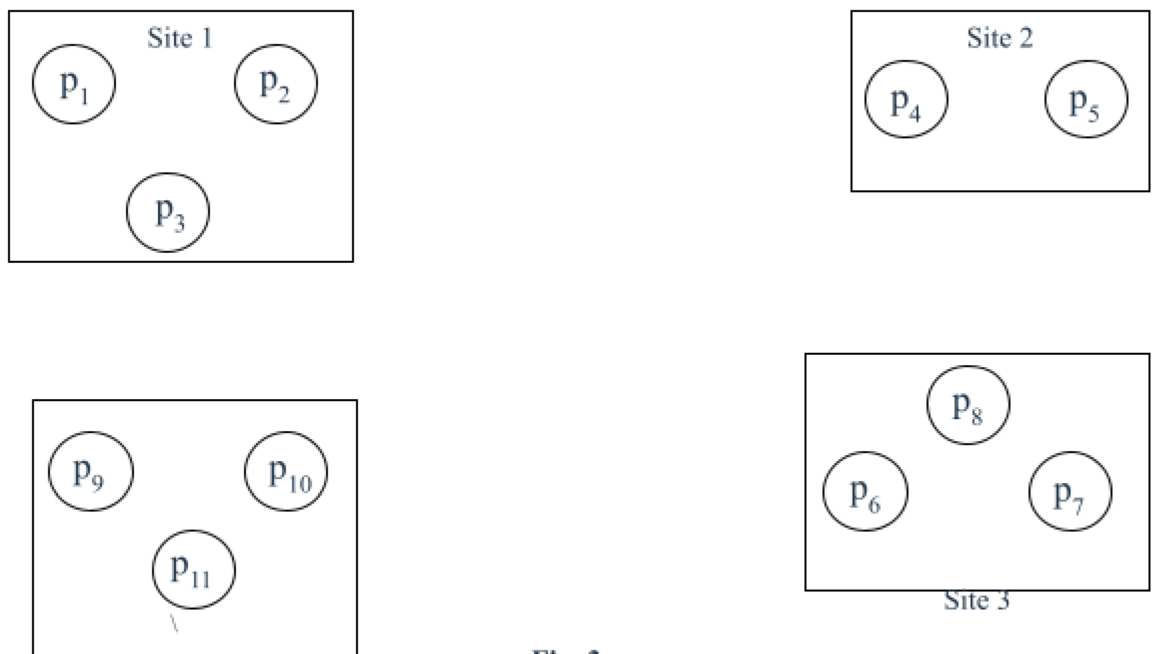


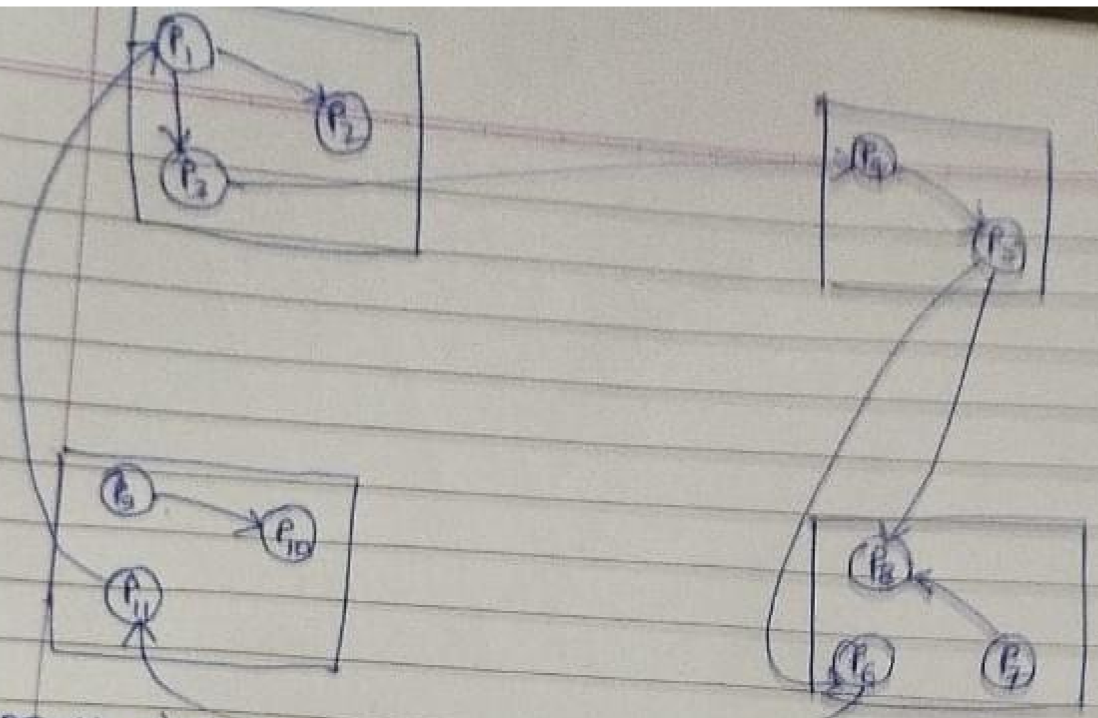
Fig. 3

The following dependencies exist among the processes:

- p_1 is blocked and is waiting for p_2 and p_3 to release some resources
- p_3 is blocked and is waiting for p_4 to release some resource
- p_4 is blocked and is waiting for p_5 to release some resource
- p_5 is blocked and is waiting for p_6 and p_8 to release some resources
- p_7 is blocked and is waiting for p_8 to release some resource
- p_6 is blocked and is waiting for p_{11} to release some resource
- p_9 is blocked and is waiting for p_{10} to release some resource
- p_{11} is blocked and is waiting for p_1 to release some resource

Draw the WFG for the above scenario.

Execute Chandy-Misra-Haas algorithm for the AND model on this distributed system to determine whether any deadlock exists in the system. Assume p_1 initiates the deadlock detection and initially, the dependent arrays of all the processes contain false as values. Also, assume that all nodes of the WFG obtained above are AND nodes. Note that if a deadlock exists, it is to be detected by using the algorithm and not by any other means. Write the steps of the algorithm as is applicable to the scenario given above. DO NOT ANNOTATE IN THE DIAGRAM. If you annotate in the diagram without writing the steps of the algorithm or detect deadlock without applying the algorithm, NO MARKS WILL BE AWARDED.



P_1 process initiates CMH AND model.

step1: Since P_1 is not self dependent process, so further sending of probemessage will be there.

dependent$_1$(2) = True

dependent$_1$(3) = True

probe message P_1, P_3 → (1, 1, 3)

probe message P_1, P_2 → (1, 1, 2)

step 2: Since P_2 is active process, so, it will discard received probe message (1, 1, 2) from P_1

step3: P_3 is blocked process, will further send probe m

dependent$_3$(4) = True

probe message P_3, P_4 → (1, 3, 4)

Step4: P_4 is blocked process, will send probe message furth

dependent$_4$(5) = True

probemessage P_4, P_5 = (1, 4, 5)

Step5: P_5 is blocked process, will send probemessage fu

dependent$_5$(8) = True

dependent$_5$(6) = True

probe message P_5, P_6 = (1, 5, 6)

probe message P_5, P_8 = (1, 5, 8)

Step 6: P_8 is active process, so will discard all probe message it will rec

Step 7: P_7 is not part of global WFG, so will not send probe message

Step 8: P_6 is blocked process and send probe message further

$dependent_6(11) = T$

probe message $P_6, P_{11} = (1, 6, 11)$

Step 9: P_9 and P_{10} are not part of global WFG, so will not send probe message

Step 10: P_{11} is blocked and send probe message further

$dependent_{11}(1) = True$

probe message $P_{11}, P_1 = (1, 11, 1)$

Step 11: On receiving probe message as (1, 11, 1) from P_{11} to P_1, since $k = i$ (1=1), P_1 will declare deadlock

Q.7Set.(A) Consider a synchronous distributed system consisting of 8 processes – P_1, P_2, P_3, P_4, P_5, P_6, P_7 and P_8 . Every process puts forth an initial integer value. The initial values put forth by P_1, P_2, P_3, P_4, P_5, P_6, P_7 and P_8 are 1, 3, 4, 4, 2, 3, 1 and 6 respectively. Out of the 8 processes, P_2, P_5 and P_7 are faulty processes and the non-faulty processes are aware of this fact. Using your knowledge of the interactive consistency problem, determine the 8 values of the final agreement array. Assume that for a faulty process, the agreed upon value by all the non-faulty processes is 0. You do not need to consider how many maximum faulty processes can be tolerated by the system. [4]

⇒ Defined processes as faulty with doing ~~circle on it.~~

P_1 — 1
P_2 — 3
P_3 — 4
P_4 — 4
P_5 — 2
P_6 — 3
P_7 — 1
P_8 — 6

with 3 processes as faulty minimum no. of total nodes/processes expected as $3 \times 3 + 1 = 10$ processes for consensus.

Since except P_2, P_5, P_7 all processes are healthy, so will continue same values further since agreed values for faulty processes for ~~1, 4,~~ non-faulty are 0.

final values as $(1, 0, 4, 4, 0, 3, 0, 6)$ P_7 will
assume P_2 will send $\$$ value, P_5 will send $\#$, send $\&$

P_1 will get $(1, \$, 4, 4, \#, 3, \&, 6)$
P_2 will get $(1, \$, 4, 4, \#, 3, \&, 6)$
P_3 will get $(1, \$, 4, 4, \#, 3, \&, 6)$
P_4 will get $(1, \$, 4, 4, \#, 3, \&, 6)$
P_5 will get $(1, \$, 4, 4, \#, 3, \&, 6)$
P_6 will get $(1, \$, 4, 4, \#, 3, \&, 6)$
P_7 will get $(1, \$, 4, 4, \#, 3, \&, 6)$
P_8 will get $(1, \$, 4, 4, \#, 3, \&, 6)$

final values as $(1, 0, 4, 4, 0, 3, 0, 6)$
as final agreed values for faulty processes to non-faulty process as

Q.8Set.(A) For a Chord ring of m = 6, the nodes as shown in Fig. 4 are present.

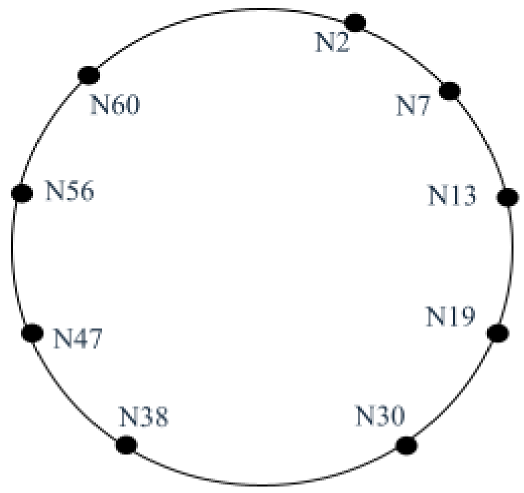
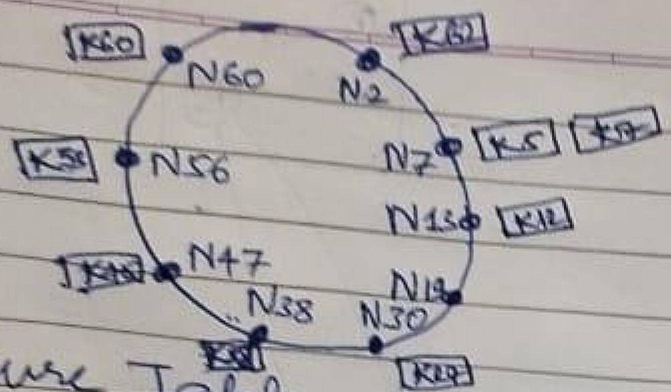


Fig. 4

Consider the following 9 keys – K5, K7, K12, K27, K38, K45, K53, K60 and K62. Identify on which node each of the above mentioned keys will be placed for this Chord ring. You should clearly mention the reason for placing each key on a particular node.

Determine the entries of the finger tables of N7, N19 and N47. [4.5 + 4.5 = 9]

Chord Ring m=6



Fingure Table

N17

17+1=18 → N19

17+2=19 → N19

17+4=21 → N30

17+8=25 → N30

17+16=33 → N38

17+32=49 → N56

N19

19+1=20 → N30

19+2=21 → N30

19+4=23 → N30

19+8=27 → N30

19+16=35 → N38

19+32=51 → N56

N47

47+1=48 → N56

47+2=49 → N56

47+4=51 → N56

47+8=55 → N56

47+16=63 → N2

47+32=79 → N19

Keys are as below

K5, K7, K12, K27, K38, K45, K53, K60 and K62.

m=6

Succ(5) = N7 ← K5

Succ(7) = N7 ← K7

Succ(12) = N13 ← K12

Succ(27) = N30 ← K27

Succ(38) = N38 ← K38

Succ(45) = N47 ← K45

Succ(53) = N56 ← K53

Succ(60) = N60 ← K60

Succ(62) = N2 ← K62

Qtext :

Suppose a synchronous system consists of 10 processes out of which 3 processes are faulty. Assume that the Byzantine agreement tree algorithm is executed on this system.

What will be the total number of rounds for the algorithm? [1]

Each message will be sent to how many nodes in the last round? [1]

Ans: As number of faulty processes are 3, so the number of rounds will be 3 + 1 = 4
Number of messages in last round = (10-1) * (10-2) * (10-3)* (10-4) = 9 * 8 * 7 * 6 = 3024

Qtext :

A Chord ring with m = 8 contains the following nodes – N10, N25, N42, N56, N80, N101, N137, N159, N210, N242, and N251.

The chord ring contains the following keys – K10, K57, K201 and K230. Identify on which node each of the above-mentioned keys will be placed for this Chord ring. You should clearly mention the reason for placing each key on a particular node. You are not required to draw any diagram for answering this question. [4]

Suppose for this Chord ring, a lookup for K230 is initiated at N101. Describe the step-by-step procedure by which K230 can be located using the simple lookup algorithm. Write the steps of the algorithm clearly in text. You are not required to draw any diagram. If you answer this question by only drawing a diagram, no marks will be awarded. [4]

Ans :
m = 8;

Successor (10) = N10 ==> K10
Successor (57) = N80 ==> K57
Successor (201) = N210 ==> K201
Successor (230) = N242 ==> K230

Simple Lookup
(variable)
integer: successor ← initial value;
1) i: Locate_Successor (Key), where key is not at i;
1a) if Key ∈ (i, successor) then
return (successor)
else return (successor.Locate_Successor(key))
key → K230, look up position N101
Step 0 → check that key K230 located at N101 → False
Step1 230 ∈ [101, 137] → False
Step2 230 ∈ [137, 159] → False
Step3 230 ∈ [159, 210] → False
Step4 230 ∈ [210, 242] → True
Successor is 242 → N242