SL.NO:CAA 212853

SRI SIVASUBRAMANIYA NADAR COLLEGE OF ENGINEERING

(An Autonomous Institution, Affiliated to Anna University, Chennai)
Rajiv Gandhi Salai (OMR), Kalavakkam – 603 110

THEORY EXAMINATIONS

Register Number	205001085		
Name of the Student	Sabaarivasan. V		
Degree and Branch	BE CSE	Semester	VII
Subject Code and Name	UCS1701 - Distributed Systems		
Assessment Test No.	III	Date	3/11/2023

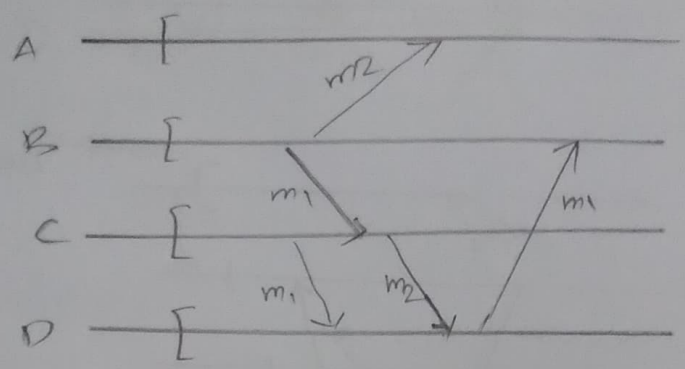
Details of Marks Obtained									
Part A		Part B				Part C			
Question No.	Marks	Question No.	(a) Marks	(b) Marks	Total Marks	Question No.	(a) Marks	(b) Marks	Total Marks
1	2	7	5			10	8		
2	2								
3	2	8	6			11			
4	2					12			
5	2	9	6			13	10		
6	2								
Total (A)		Total (B)				Total (C)			
Grand Total (A+B+C)					Marks (in words)				
Signature of Faculty									

~~PART - A~~

~~(1) Effect of consensus for faulty source~~

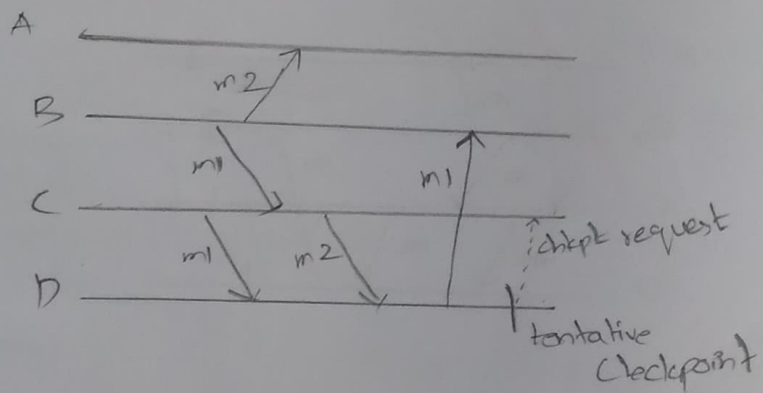
PART - C

(10)



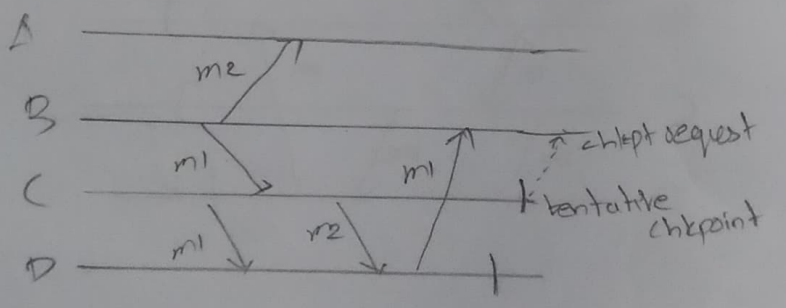
$Ckpt\_cohort = \{ y \mid last\_label\_recvd_x[y] \geq 1 \}$

$last\_label\_recvd_x[y] \geq last\_label\_send_y[x] \geq 1$



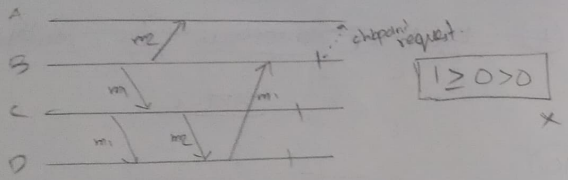
✓

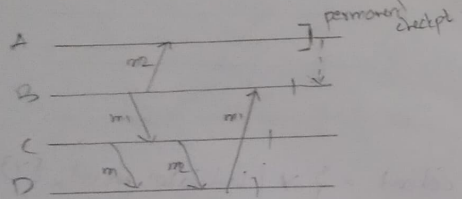
$2 \geq 1 > 0$

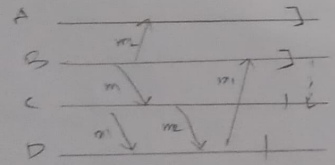


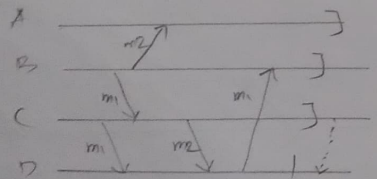
✓

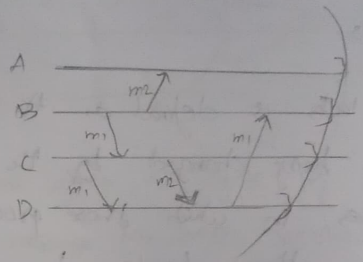
$1 \geq 1 > 0$


A
B
C
D
m2
m1
m1
m1
m2
checkpoint request
1 ≥ 0 > 0
x


A
B
C
D
m2
m1
m1
m1
m2
permanent checkpt

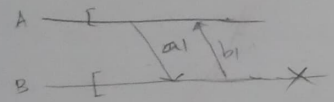

A
B
C
D
m2
m1
m1
m1
m2


A
B
C
D
m2
m1
m1
m1
m2


A
B
C
D
m2
m1
m1
m1
m2
globally consistent

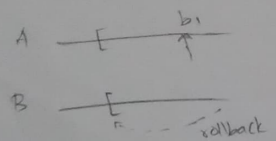
(13) Live locks

⇒ Live locks is defined as the state of a process is being changed by the state of other processes in which those process being active do nothing, leading to waste of time and resources.

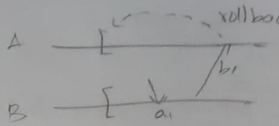


⇒ Let us consider that process B gets crashed after sending b_1 message from B to A.

⇒ So Process B is rollbacked to its checkpoint

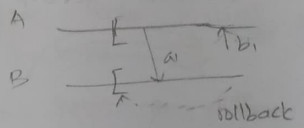


⇒ But since b_1 message is already sent before rollback, process A also needs to get roll backed.

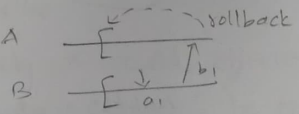


⇒ Even after a cycle of rollback is done, still a_1 message has been received by B from A.

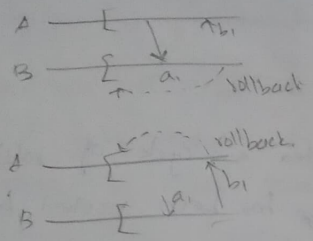
⇒ So again B has to rollback.



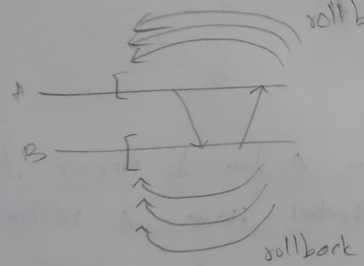
⇒ Now A has b_1 message from B which has rollbacked. Hence A rollbacks again.



So the process happens again and another set of rollbacks takes place ~~continuous~~ continuously which leads to live lock.



Overall the rollback scenario is,



(8) PART - B

Distributed Indexing of P2P systems

⇒ P2P systems is a type of decentralised distributed system where there won't be a centralised server.

⇒ Multiple nodes called peers are present in these systems.

⇒ Peers help in either sending data or request data from other peers which means the peer itself acts as a server and a client.

⇒ Scalability is well efficient as nodes can be easily added to the system.

⇒ Anonymity is maintained in which no data about the users is known.

⇒ P2P systems are autonomous in which consistency and performance of the nodes are well maintained.

⇒ Distributed indexing is followed in P2P systems in which nodes are indexed for easy routing and object storage.

⇒ P2P systems communication happen among the nodes by means of routing algorithms with the help of distributed indexing.

⇒ Otherwise flooding or random walk techniques are followed where indexing of nodes are not followed.

⇒ A node or peer can do both operations at the same time (i.e) it can provide data to a peer and request data from another peer. Operations happening in a peer are not interrupted.

⇒ Examples of P2P systems are Gnutella, Bit Torrent.

(9) Lamport's Bakery Algorithm

⇒ Lamport's Bakery algorithm is used to maintain mutual exclusion principle among different processes doing different activities at the same time.

```
lock(p):
{
    num[i] = max [num[0], num[1], ... num[p]) + 1
    for (I=0 to p)
        while ( num[i] ≠ 0 and num[i] < num[p]);
}
```

```
release(p):
{   num[i] = 0 }
```

Example

⇒ Consider 5 processes P_1 P_2 P_3 P_4 P_5.

⇒ Now P_2 starts an activity, so lock is applied to it.

⇒ Lock is applied by increasing its current state value by 1.

P_1	P_2	P_3	P_4	P_5
0	0	0	0	0

⇒

P_1	P_2	P_3	P_4	P_5
0	1	0	0	0

⇒ Similarly, next P_4 starts an activity and next P_5 starts an activity.

So,

P_1	P_2	P_3	P_4	P_5
0	1	0	2	3

⇒ Now we must enter into the critical state to perform the operations.

⇒ The num(i) value which has the least will be first entered into the critical state and perform its operations.

⇒ While leaving from critical state, num(i) is assigned to zero which is considered as a release.

Next P_5 in critical state.

	P_1	P_2	P_3	P_4	P_5
⇒	0	0	0	2	3

→

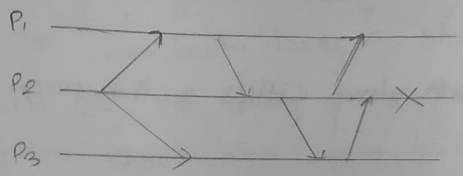
P_1	P_2	P_3	P_4	P_5
0	0	0	2	3

↓

P_1	P_2	P_3	P_4	P_5
0	0	0	0	0

(7)

Asynchronous Recovery Algorithm



Algorithm

If Process i is recovering after failure,

then $CkPt(i)$ = latest event in stable storage.

else

$CkPt(i)$ = latest event that took place.

for (k=1 to N) do

begin

for each neighbour process j do

send ROLLBACK (I, $SENT_{ij}$, $CkPt(i)$);

wait for ROLLBACK message from neighbour.

If $(RCVD_{i,j}(CkPt(i)) \leq SEND_{j \to i}(CkPt(j)))$

then proceed with ROLLBACK.

Else

add temp CkPt(j) to that process j

end

⇒ The algorithm is used to recovery from a crashed state in asynchronous mode of checkpoints maintained in distributed systems.

PART-A

(1)

⇒ Consensus algorithm states that if a source is non-faulty having a simple value, then all the non-faulty process present in that system should also agree to that same value.

⇒ If source becomes faulty, then the values which are present on other non-faulty processes is shared with the source to maintain consistency.

(2) Orphan messages:

Orphan messages are defined as during a recovery state or rollback, the message will have been received but the sender information is not available.

(3)

P2P System	Client Server System
Decentralised system	Centralised system
Nodes acts as both server and client.	Server receives requests and sends responses which are received by clients
High fault tolerance.	Single point of failure
Scalability is efficient	Scalability is little complicated
Eg: Bit Torrent.	Eg: Webservers

(4) Overlay networks

Overlay networks are defined as a virtual network which is present on another network which has purpose for handling routing algorithms and for object storing in decentralised distributed systems.

(5) Applications of P2P systems

→ Bit Torrent
→ Gnutella
→ Bitcoin (Cryptocurrency)
→ File sharing

(6) ⇒ Local indexing is defined as indexing is applied on nodes or local network for easy routing in P2P systems.

⇒ Optimistic overlay uses local indexing.