COMPUTER ENGINEERING DEPARTMENT

ASSIGNMENT NO. 1

Subject: Distributed Computing

COURSE: B.E Year: 2021-2022 Semester: VIII
DEPT: Computer Engineering
SUBJECT CODE: CSC802 DUE DATE: 11/03/2022

Roll No.: 50 Name: Amey Mahendra Thakur

Class: BE-Comps B Date of Submission: 11/03/2022

DC Assignment - 1

Sr. No.	Questions
1	Discuss the common issues with which the designer of a heterogeneous distributed system must deal.
2	Discuss different models of middleware.
3	Discuss the call semantics of RPC.
4	Discuss the stream-oriented communication in detail.
5	Explain Berkeley Algorithm and Cristian Algorithm along with advantages and disadvantages.
6	Discuss Meakawa Algorithm with an example along with deadlock handling messages.

Student Signature:

AMEY THAKUR B-50 Amey.

Q1. Discuss the common issues with which the designer of a heterogeneous distributed system must deal.

Ans:

Common Design Issues:

① Transparency
- User should not be aware of hidden facts of the systems.
- Types of transparency are:
  - Location
  - Access
  - Migration
  - Relocation
  - Replication
  - Failure
  - Concurrency
  - Persistence

② Reliability
- The system should be fault tolerant to handle failures in DS.

③ Flexibility
- The system should be flexible enough to adopt new changes i.e. ease of enhancement.

④ Scalability
- Increasing number of devices should not hamper basic working of the system.

⑤ Performance
- The performance of DS should be at least as good as centralized system.

AMEY THAKUR B-50 Amey.

⑥ Heterogeneous Environment
- Since the Ds is forming a heterogeneous environment, there is a need to break into consideration heterogenecity.

⑦ Emulation of Existing Software!
- Architecture should support all software supported by centralized system.

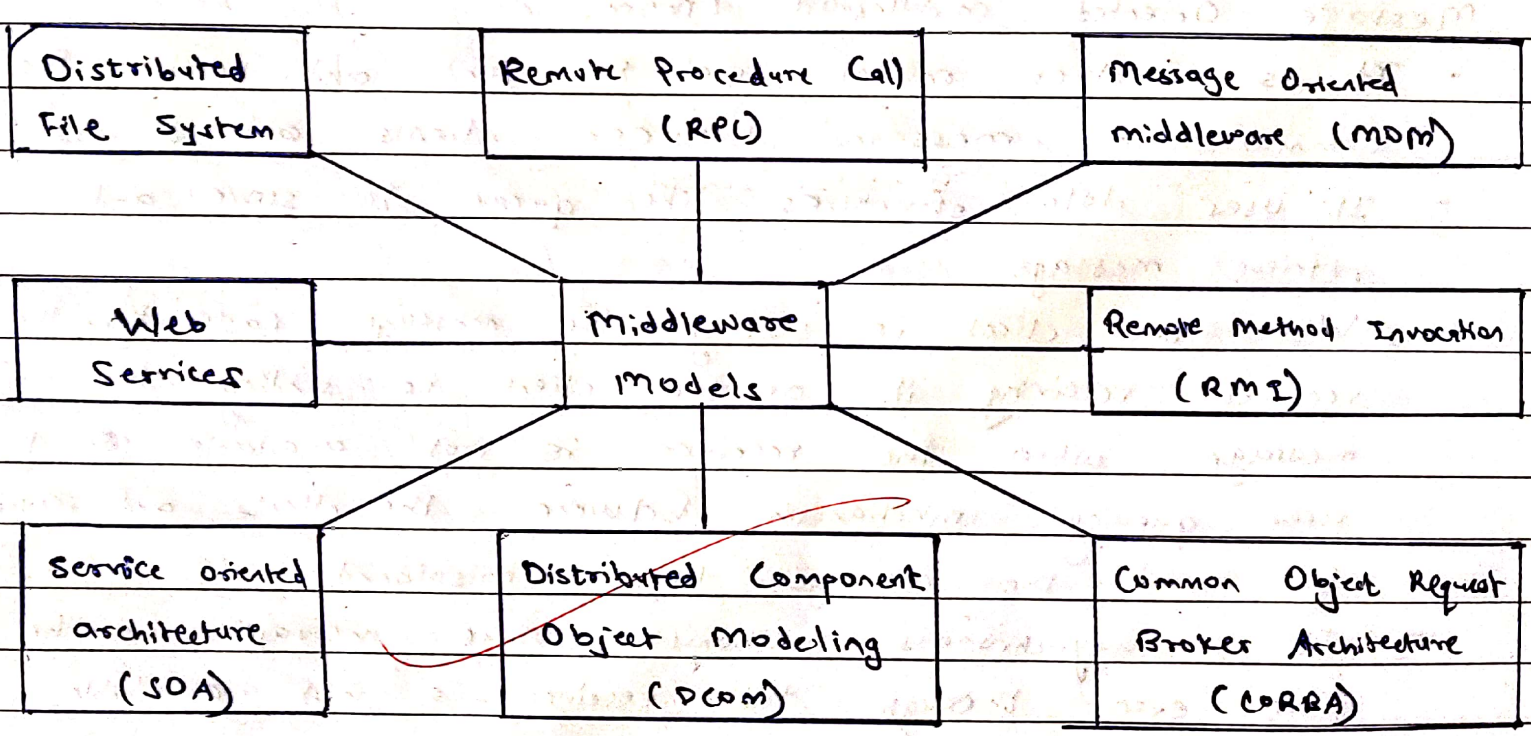
⑧ Security
- Data should be secured and and should be transmitted securely over the network.

AMEY THAKUR B-50 Amey

Q2. Discuss different models of middleware

Ans:

- The middleware era was started with the distributed file systems which was the first middleware model where the files were stored and distributed over the network from the remote file server.
- The files were accessible over the network using distributed file system services which then evolved with the Remote Procedure Call (RPC)
- Remote method Invocation (RMI), CORBA, DCOM, etc.



AMEY THAKUR B-50 Amey

Remote Procedure Call (RPC)

- It is one of the successful middleware models used in modern distributed applications for communication.
- It uses local call to call a procedure resides on the remote machine to get the result.
- Hidden communication is done between client and server.
- For example, if a user wants to get the sum of two numbers which is defined at the remote server, he calls method with parameters using a local method call which is then translated to the remote call.

Message Oriented Middleware (MOM)

- It is another model used to send and receive the communication messages between clients and servers.
- It uses data structures like queue to store and retrieve message
- When the client is sending the message faster than the receiver receiving it or the client is sending the message when the receiver is not available, so it use queuing mechanism between the client and server to avoid the message been misplaced.
- It is asynchronous mechanism where message can be sent even through the receiver is not available
- For Example, Email system, where sender can send ~~message~~ mails to the receipient who is not available at that moment.

AMEY THAKUR TB-50 Amey

Remote Method Invocation (RMI)

- In this objects are distributed and located by using the RMI registry.
- The client can access remote objects by using the interfaces
- Disadvantage of RMI is that it does not have support of the heterogenity and is compatible with Java platform only.

Common Object Request Booker Architecture (CORBA)

- It is one of the most popular distributed object technologies where objects can be accessible from remote location ORB.
- Server and client communicate with each other through object request broker bus.
- To map the semantics of objects and fetch the appropriate object an interface definition language is used.
- It is evolved with service based middleware where service models are used.
- In service model, the services are published by the service providers and consumed by the service consumer.

Distributed Object Technology

- The distributed object technology has changed the scope of middleware technologies to one step up where objects are distributed to the remote server to facilitate the client.
- Eg. RMI & CORBA
- The distributed object mechanism hides the communication interfaces and their details to provide access to the remote objects efficiently.

AMEY THAKUR B-50 Amey

Q3 Discuss the call semantics of RPC

Ans:

- In RPC, the caller and callee processes can be situated on different nodes. The normal functioning of an RPC may get disrupted due to one or more reasons mentioned below:
  1. Call message is lost or response message is lost.
  2. The callee node crashes and is restarted.
  3. The caller node crashes and is restarted.
- In RPC system the call semantics determines how often the remote procedure may be executed under fault conditions.

Types of RPC call semantics

1. May-Be call semantics
   - It is the weakest semantics in which a timeout mechanism is used that prevents the caller from waiting indefinitely for a response from the callee.
   - This means that the caller waits until a pre-determined timeout period and then continues to execute.
   - Hence the semantics does not guarantee the receipt of call message nor the execution. This semantic is applicable where the response message is less important and applications that operate within a local network with successful transmission of messages.

AMEY THAKUR B-50 Amey

(2) Last - Once call semantics

- This call semantics uses the idea of retransmitting the call message based on timeouts until the caller receives a response.
- The call, execution and result of will keep repeating until the result of procedure execution is received by the caller
- The results of the last executed call are used by the caller, hence it is known as last-one semantics.
- Last one semantics can be easily achieved only when two nodes are involved in the RPC, but it is tricky to implement it for nested RPCs and cases by orphan calls.

(3) Last - of - many call Semantics

- This semantics neglects orphan calls unlike least-once call semantics. Orphan call is one whose caller has expired due to node crash.
- To identify each call, unique call identifiers are used which to neglect orphan calls.
- When a call is repeated, it is assigned to a new call identifier and each response message has a corresponding call identifier.
- A response is accepted only if the call identifier associated with it matches the identifier of the most recent call else it is ignored.

AMEY THAKUR B-50 Amey

④ At-least-once call Semantics

- This semantics guarantees that the call is executed one or more times but does not specify which results are returned to the caller.
- It can be implemented using timeout based retransmission without considering the orphan calls.

⑤ Exactly-Once call semantics

- This is the strongest and most desirable call semantics. It eliminates possibility of a procedure being executed more than once irrespective of the number of retransmitted call.
- The implementation of exactly-once call semantics is based on the use of timeouts, retransmission, call identifiers with same identifier for repeated calls and a reply cache associated with the callee.

AMEY THAKUR B-50 Amey

Q4. Discuss the stream oriented communication in detail.

Ans:

Stream-oriented Communication

- RPC and message-oriented communication are based on the exchange of discrete messages.
- Timing might affect performance, but not correctness.
- In stream-oriented communication the message content must be delivered at a certain rate, as well as correctly
- Example: Music or Video
- Stream oriented communication supports continuous media communication.

Transmission modes in stream-oriented communication

① Asynchronous Mode
- No restrictions with respect to when data is to be delivered.

② Synchronous Mode
- Define a maximum end-to-end delay for individual data packets.

③ Isochronous Mode
- Define a maximum end-to-end delay and maximum delay variance

AMEY THAKUR B-50 Amey

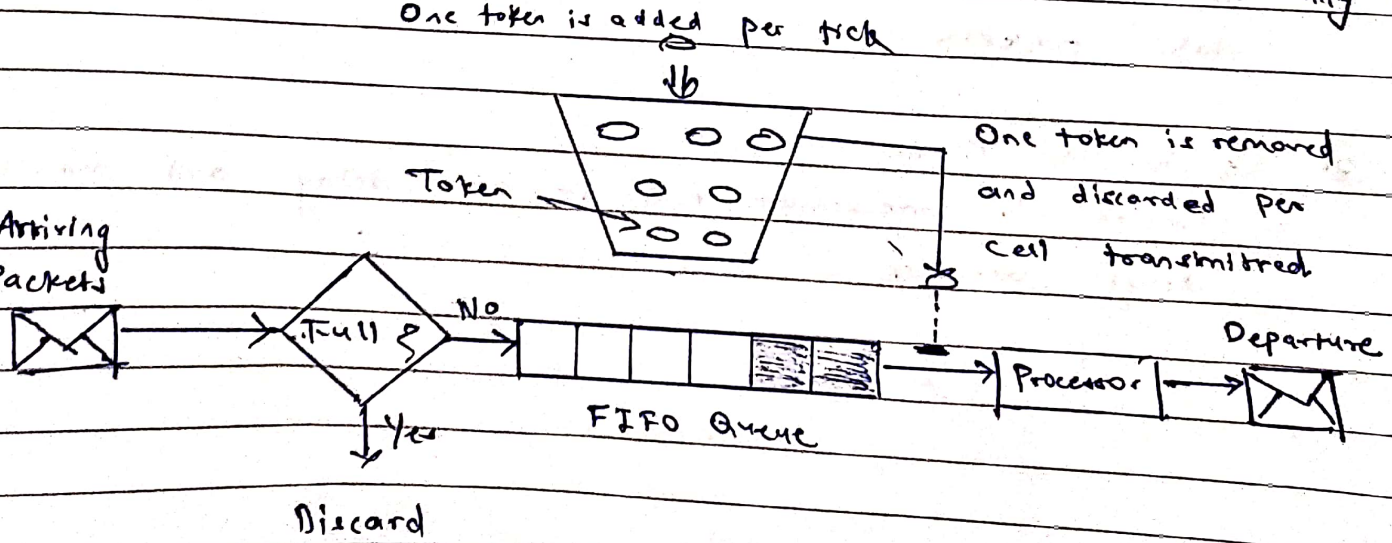
Stream:

- A data stream is a connection oriented communication facility
- It supports isochronous data transmission
- Some common stream characteristics:
  - (a) stream are unidirectional
  - (b) There is generally a single source.
- Stream Types:
  - (a) Simple: It consists of a single flow of data. (eg audio or video)
  - (b) Complex: It consists of multiple data flows (eg stereo audio or combination audio/video)

Example:

Token Bucket Algorithm

- The token bucket can be easily implemented with a counter
- The token is initialized to zero.
- Each time a token is added, counter is incremented by 1.
- Each time a unit of data is dispatched, counter is decremented by 1
- If the counter contains zero, the host cannot send any data

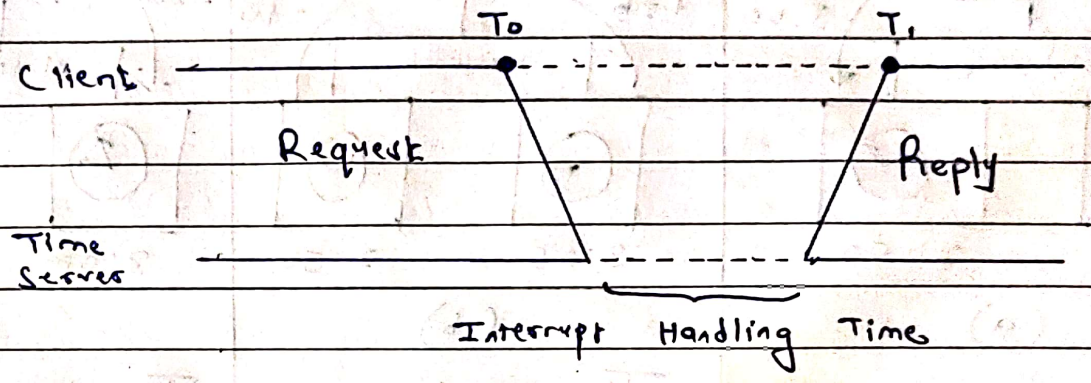


AMEY THAKUR B-50 Amey

Q5 Explain Berkeley Algorithm and Cristian Algorithm along with advantages and disadvantages.

Ans:

Cristian's Algorithm



- Cristian's algorithm is a clock synchronization algorithm used to synchronize time with a time server by client processes.
- This algorithm works well with low-latency networks where round trip time is short as compared to accuracy.
- Here round trip time refers to the time duration between start of a request and end of corresponding response.
- In this method, each node periodically sends a message to server.
- When the time server receives the message it responds with a message T,

  T = Current time of server node.
- Assume the clock time of client be To when it sends the message T, when it receives the message from server
- T, and To are measured using same clock.

  $T_1 - T_0 / 2$
- When reply is received at clients node

  clock is readjusted to $T + (T_1 - T_0)/2$

Advantage:

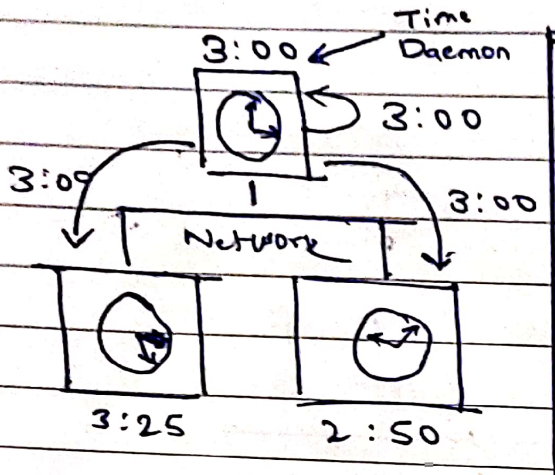
- It assumes that no additional information is available

Disadvantage:

- It restricts the number of measurements for estimating the value

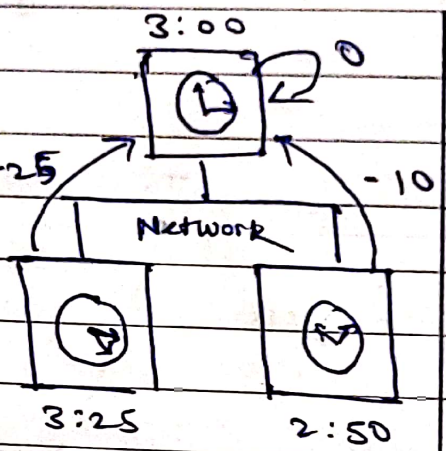
AMEY THAKUR B-50 Amey

Berkeley Algorithm:



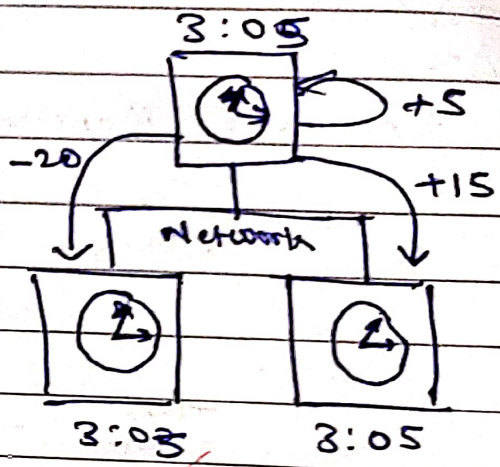
(a)

The time daemon asks the other machines for their clock values



(b)

The machine answer



(c)

The time daemon tells everyone how to adjust their clock

- The Berkeley algorithm is a method of clock synchronization in distributed computing which assumes no machine has an accurate time source.
- It is an active time server approach.
- In this approach, the time server periodically sends a message to all the computers in the group of computers.
- When this message is received each computer send back its own clock value to the time server.
- The time server has a prior knowledge of the approximate time required for propogation of a message which is used to readjust the clock values.
- It then takes average of clock values of all the computers.
- The calculated average is the current time to which all clocks should be readjusted.

AMEY THAKUR B-50 Amey

- The time server readjusts its own clock to value and instead of sending the current time to other computers it sends amount of time each computer needs for readjustment.
- This can be positive or negative value.

AMEY THAKUR B-60 Amey Thakur

Q6. Discuss Maekawa algorithm with example along with deadlock handling message

Ans:

- To get access to a CS, not all processes have to agree.
- Suffices to split set of processes up into subsets ("voting sets") has overlap.
- Suffices that there is consensus within every subset.
- When a process wishes to enter the CS, it sends a vote request to every member of its voting district.
- When the process receives a vote request, it responds with a "YES" vote if it has not already cast its vote.
- When a process exists the CS, it informs the voting districts which can then vote for other candidates may have deadlock

$N = k(k-1) + 1$ where, $N \rightarrow$ No. of processes
$k \rightarrow$ Site of quorum
$R \rightarrow$ No. of quorum

Algorithm

- A site Si executes the following steps to execute the CS requesting the critical section

(a) A site Si requests access to the CS by sending REQUEST (i) message to all sites in its request set Ri

(b) When a site Sj receives the REQUEST (i) message, it sends a REPLY (j) message to Si provided it hasn't sent a REPLY message to a site since its receipt of the last RELEASE message. Otherwise, it queues up the REQUEST (i) for later consideration

AMEY THAKUR B-50 Amey.

- Executing the critical section

(c) Site S_i executes the CS only after it has received a REPLY message from every site in R_i

- Releasing the critical section

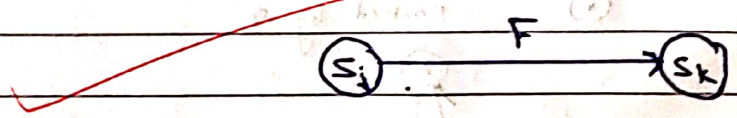
(d) After the execution of the CS is over site S_i sends a RELEASE (i) message to every site in R_i

(E) When a site S_j receives a RELEASE (i) message from site S_i, it sends a REPLY message to the next site waiting in the queue and deletes that entry from the queue. If the queue is empty then the site updates its state to reflect that it has not sent out any REPLY message since the receipt of the last RELEASE message

Deadlock handling message

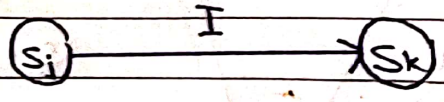
FAILED :

- F, S_j cannot grant permission to S_k because S_j has granted permission to a site with higher request priority



INQUIRE

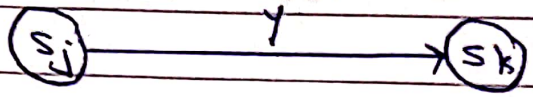
- I, S_j wants to finds out if S_k has successfully locked all sites



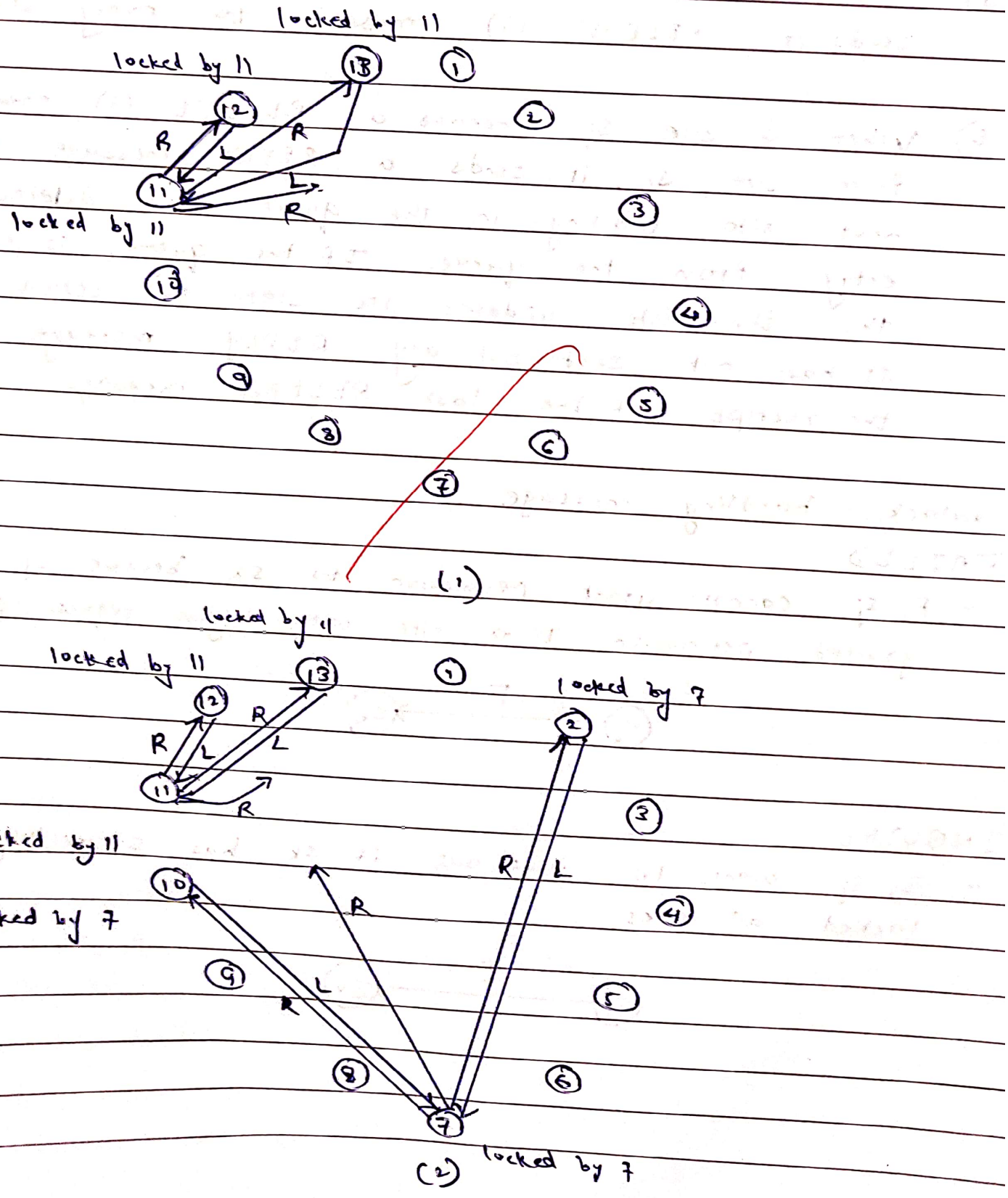
AMEY THAKUR B-50 Amey

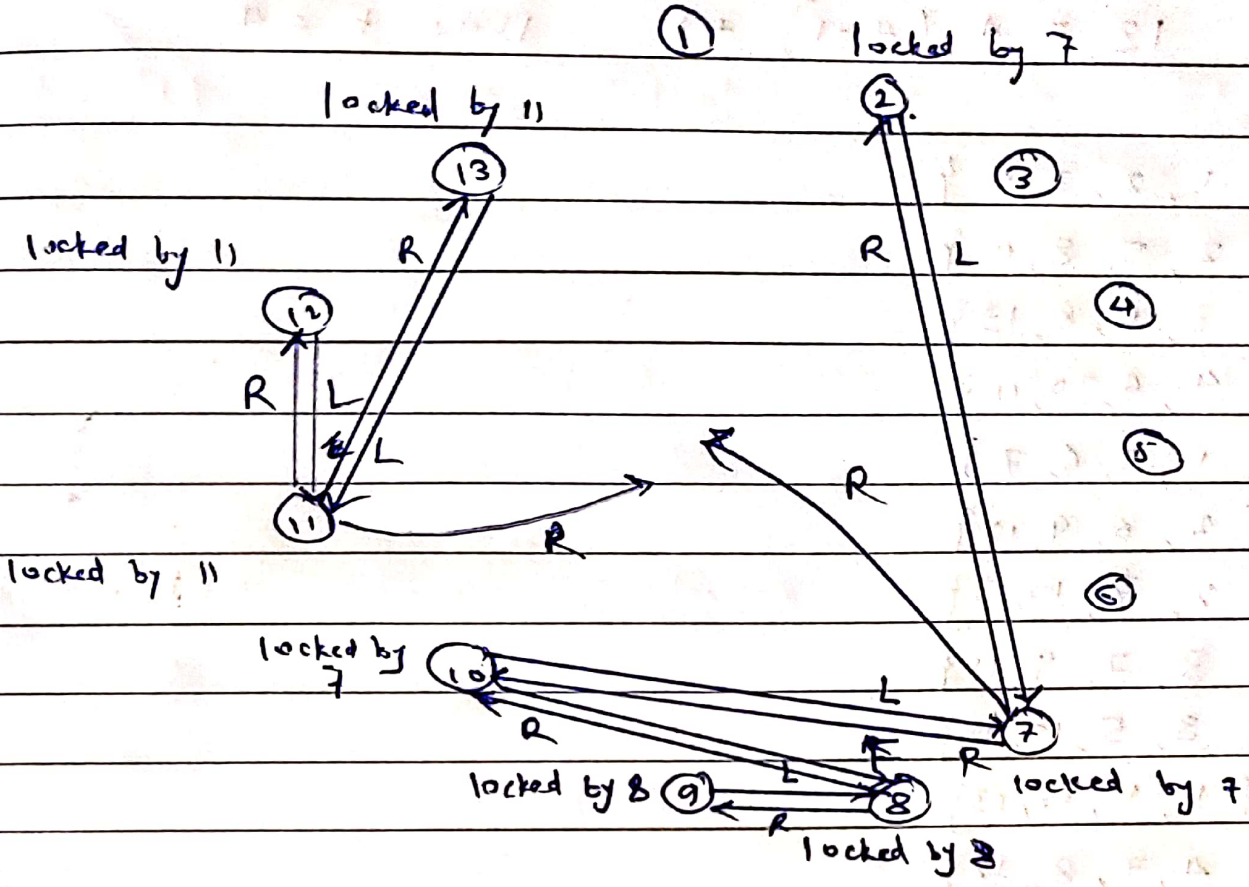
YIELD:

- Y, S_j yields to S_k.

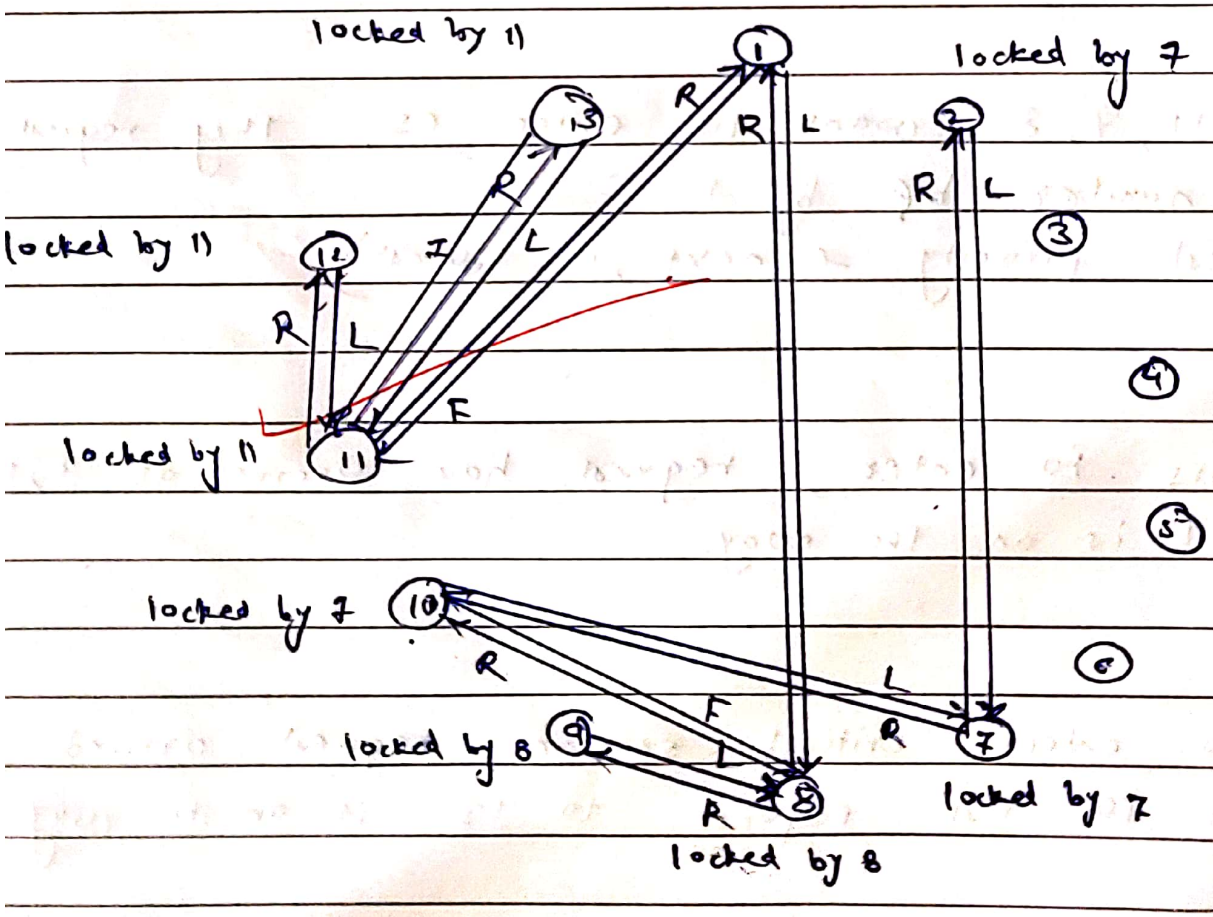


Example:




1
locked by 7
locked by 11
2
13
3
locked by 11
R
R
L
12
4
R
L
L
5
R
11
R
locked by 11
6
locked by 7
10
L
R
7
R
locked by 8
9
L
8
locked by 7
R
locked by 8

(3)


locked by 11
1
locked by 7
13
R
R
L
2
R
L
locked by 11
12
I
L
3
R
L
4
F
locked by 11
11
5
locked by 7
10
R
L
6
F
locked by 8
9
L
R
7
8
locked by 7
R
locked by 8

(4)

AMEY THAKUR B-50 Amey

13 (nodes), $13 = 4(4-1)+1$, thus $k = 4$

$R_1 = \{1, 2, 3, 4\}$
$R_2 = \{2, 5, 8, 11\}$
$R_3 = \{3, 6, 8, 13\}$
$R_4 = \{4, 6, 10, 11\}$
$R_5 = \{1, 5, 6, 7\}$
$R_6 = \{2, 6, 9, 12\}$
$R_8 = \{2, 7, 10, 13\}$
$R_9 = \{3, 7, 9, 11\}$
$R_{10} = \{3, 5, 10, 12\}$
$R_{11} = \{1, 11, 12, 13\}$
$R_{12} = \{4, 7, 8, 12\}$
$R_{13} = \{4, 5, 9, 13\}$

Suppose site 11, 7, 8 want to enter CS. they requests with sequence number of by 1.
(7 has highest priority & next, 11 lowest)

Step 1
- Site 11 wants to enter, request has arrived at 12, 13 request to 1 is on the way.

Step 2
- 7 wants to enter critical section, request arrived at 2 and 10 but request to 13 is on its way.

AMEY THAKUR B-50 Amey

Step 3:
- 8 also wants to enter critical section, request arrived at 1 and 9. but request to 10 fails because 10 has been locked by 7.

Step 4:
- Request from 11 finally arrived at 1 and request from 7 arrived at 13.

Step 5:
- 8 receives F (Failure Message) and cannot enter CS.

Step 6:
- 11 receives F and cannot enter CS.

Step 7:
- 7 cannot enter the critical section because it has not receives all reply messages therefore 11, 7 and 8 are circular locked But the highest priority is given process id 7.

Step 8:
- As 13 is locked by 11 receives request from 7. so it sends an INQUIRE, to 11 to ask it to YIELD.

Step 9:
- When 11 receives an INQUIRE, it knowns that it cannot enter the critical section therefore, it sends a YIELD message to 13

AMEY THAKUR B-50 Amey.

Step 10:
- Then 13 can send locked message (L) to 7 which enters CS.

step 11:
- When 7 finished, sends RELEASE message

Step 12:
- Then 8 locks all members, sends RELEASE

Step 13:
- Then 11 enters CS.

BG
2/3/22