

1) The byzantine agreement Problem:

\* The byzantine agreement problem requires a designated process, called the source process, with an initial value.

\* To reach agreement with the other processes about its initial value.

Agreement:

All non-faulty processes must agree on the same value

Validity:

If the source process is non-faulty, then the agreed upon value by all the non-faulty processes must be same as the initial value of the source.



## Termination:

\* Each non-faulty process must eventually decide on a value.

## 2) Consensus Problem:

The Consensus problem differs from the byzantine agreement problem in that each process has initial value and all the correct processes must agree on a single value.

## 3) Checkpoint and Rollback Recovery:

\* checkpointing and rollback recovery are well-known techniques for recovery of failure processes.

\* when failure occurs, the process roll back to recent checkpoint.

\* The failure under consideration are ~~trans~~ hardware errors and transaction aborts



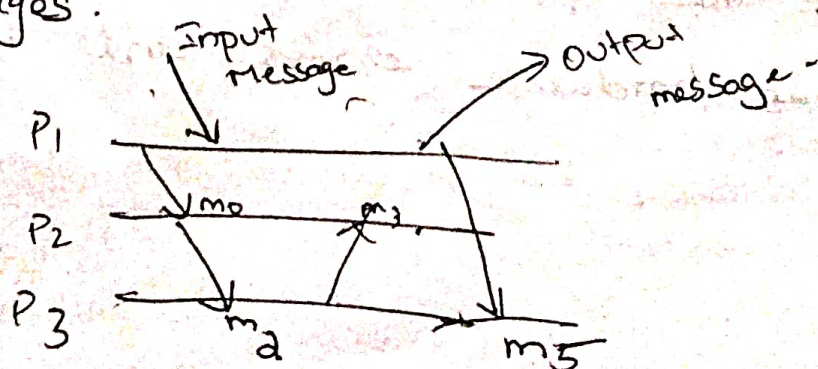
\*The saved state is called a checkpoint and the procedure of restarting is called rollback recovery.

### System model:

\*We consider the distributed system with fixed number of processes ( $P_1, P_2, \dots, P_n$ ). They are communicating with each other by using messages. The processes do not share a common memory or a common clock.

\*The messages generated by processes to advance checkpoint is referred as system messages.

\*The messages generated by the underlying distributed application are referred to as computation messages.





## Local Checkpoint:

\* Local checkpoint is a snapshot of the state of the process at a given instance and the event of recording its state of process is called local checkpointing.

## Checkpointing Algorithm:

\* processes communicate by exchanging messages through channels.

\* channels are FIFO, end to end protocol is used

\* communication failure do not interfere the network

## Types of CP:

Two types

\* Tentative - temporary checkpoint

\* permanent - local checkpoint

## Message Types:

### \* In-transit Message:

message have been sent but not yet received.

### \* Last Messages:

messages whose send is done but receive is undone due to rollback

### \* Delayed messages:

message receive is not recorded because server down or message will arrive after rollback

### \* Orphan Message:

message receive is recorded but send is not recorded.



## \* Duplicate Messages:

Arise due to message  
logging and replaying during process  
recovery

## Issues in Failure Recovery:

- \* Excess amount of resource allocation
- \* fundamental to fault tolerance
- is the recovery from an error
- once a failure occurs, it is essential to recover it to a correct state.

## Solution:

- \* Reclaim resource allocated to process
- \* undo modification made to DB
- \* Restart the process

4) Difference between Uncoordinated, Coordinated Communication Induced Check Pointing

Parameters	Uncoordinated	Coordinated	Communication Induced
No. of Check Point	many	one	many
Domino effect	possible	No	No
Orphan process	possible	No	No
Rollback extent	possible	No	No
Output Commit	Not possible	global coordinate require	global coordinate require