19/01/22

Tutorial - 03 - DS

U19CS009

Page No.:
Date : / /

Q.1 Explain terms availability, reliability, safety and maintainability.

Ans a) **Availability** => It means system is ready to serve immediately or highly available system that will start processing request or time of requesting.

b) **Reliability** => It means a system runs continuously without failure or with very low failure, but have a backup and work through it, so the service is processed without interruption.

c) **Safety** => It means when system fails, nothing catastrophic happens.

d) **Maintainibility** => It refers to how easily a failed system can be replaced.

Q.2 Explain concept of failure, error and fault with eg.

Ans => A system is said to (fail), when it cannot meet it's responses. If a distributed system made to serve n numbers of services but fails to provide one or more of them, than it is called failure.

=> An (error) is part of a system's state that may lead to failure. For eg. while transmitting packages across a network, it is expected that some packages have been damaged when they arrive at recieves's side. It means receiver may incorrectly sense a bit value or unable to detect.

=> The cause of error is called (fault). Clearly finding out what caused an error is important.

Page No.:
Date: / /

Q-3 Explain different types of fault.

Ans Following are types of errors.

1. Transient → these are types of fault which occur once and than disappear. If operation is repeated, they do not interrupt.
2. Intermittent → They cause a great deal of aggravation because they are hard to diagnose because when they occur they vanishes of their own accord and than re-appear and so on.
3. Permanent → These are the ones which continue to exists until fault component is replaced. eg. crash.

Q-3 b) → It is transient type fault.

c) → It is intermitten type fault

c) → Permanent fault.

Q-4 Explain failure masking by redundancy.

Ans → The best a fault tolerant system can do is hide the occurance of failure from other processes. The key technique for masking fault is redundancy. Following are it's major 3 types :-

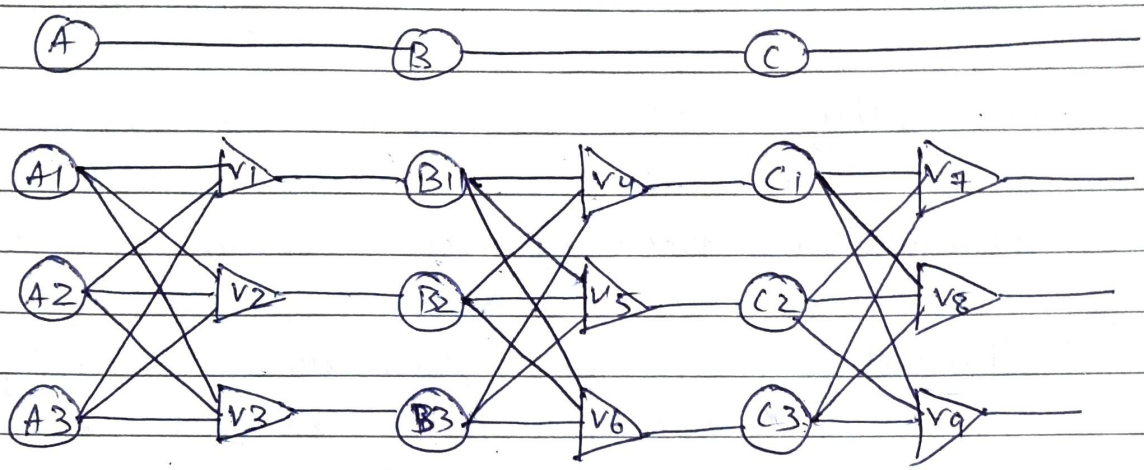
1) Information redundancy :- it adds extra bits to allow for error detection/recovery.

2. Time redundancy :- performs operation and if needs be performs it again.

3) Physical redundancy :- add (extra) duplicate hardware and/or software to system.

Page No.:
Date: / /

Triple Modular Redundancy :-



Q-5 Identify the problem in the picture and your own understanding for the problem

Ans-> Blue army has 4 troops (2 on either side) while white army has 3 troops in center. If at a time only blue army from one side attack, it will succumb.

-> Therefore blue armies (both) have to synchronise their attack, but for the same, they need to send messages through valley. It can be done in following ways, but lost messages may be caught by white army.

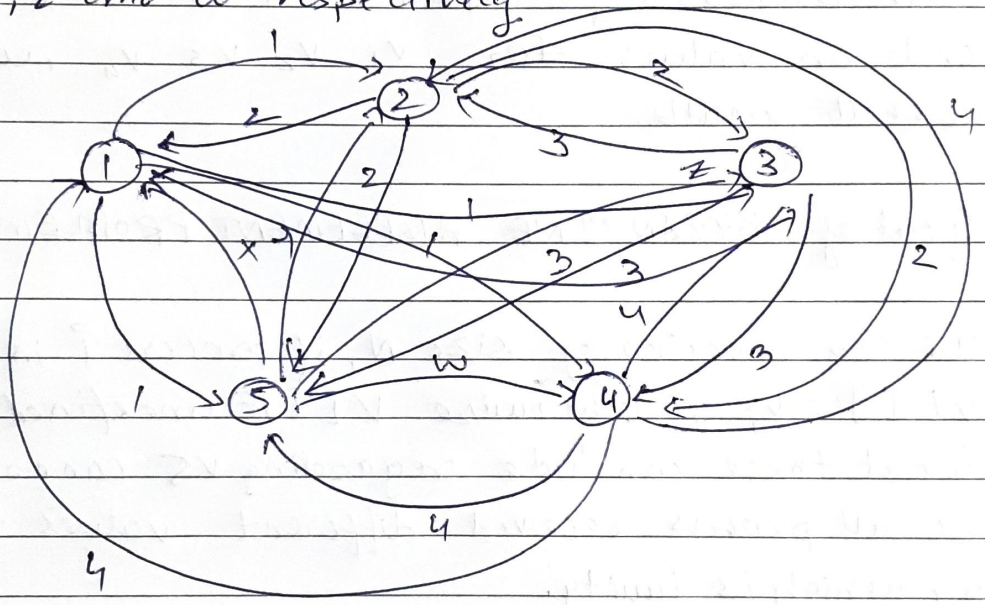
1. Approach 1 -> Blue army on either side tells other side to attack at say 14:00
   Problem -> Blue army does not know if the message is conveyed or if it was caught by white army.
2. Approach 2 -> Blue army (receiving side) sends back a message to acknowledge the blue army.
   Problem -> Blue army which sends now needs acknowledgement as well. Thus if not, they won't attack.

Q-6 Understand the steps in BYZANTINE GENERAL PROBLEM. Illustrate the working of the algorithm for $n=5, m=1$.
n -> no. of processes, m -> faulty processes.

Page No.:
Date : / /

Ans For above parameters, algorithm operates in 4 steps:

Step 1) every non-faulty process i sends v_i to every other process. We can see in below figure that process 1 reports 1, process 2 reports 2, process 3 reports 3, process 4 reports to 4 but process 5 lies to everyone, giving x, y, z and w respectively



Step 2 Results of announcements of step 1 are collected together in for of vectors

1. got (1, 2, 3, 4, x)
2. got (1, 2, 3, 4, y)
3. got (1, 2, 3, 4, z)
4. got (1, 2, 3, 4, w)
5 got (1, 2, 3, 4, 5)

Step 3 each process passes it's vector to every other process

1 got	2 got	3 got	4 got
(1, 2, 3, 4, x)	(1, 2, 3, 4, x)	(1, 2, 3, 4, x)	(1, 2, 3, 4, x)
(1, 2, 3, 4, z)	(1, 2, 3, 4, z)	(1, 2, 3, 4, y)	(1, 2, 3, 4, y)
(1, 2, 3, 4, w)	(1, 2, 3, 4, w)	(1, 2, 3, 4, w)	(1, 2, 3, 4, z)
(a, b, c, d, e)	(f, g, h, i, j)	(k, l, m, n, o)	(p, q, r, s, t)

Page No.:
Date: / /

Step-4 Each process examines i^{th} element of each of newly received vectors. If any value has a majority of that value is put into result vector. If no. of element has majority, corr element of result vector unknown.

From previous steps 1, 2, 3 & 4 all come to agreement on values for v_1 v_2 v_3 v_4, which is correct result.

Goal of BYZANTINE AGREEMENT PROBLEM

If for a vector of size N, if process i is non-faulty. $V[i] = v_i$, otherwise $V[i]$ = undefined.

what these conclude regarding v5 cannot be decided as all process received different values from process 5, which is faulty.