CS 550 Written Assignment 3

Chapter 7

Q1. A file is replicated on 6 servers. list all the combinations of read quorum and write quorum that are permitted by the voting algorithm.

Ans The following possibilities are of read qourum and write quorum :

(1, 6) , (2, 5) , (3, 4) , (4, 3) , (5, 2) , (6, 1)

Q2. What kind of consistency would you use to implement on electronic stock market? Explain your answer?

Ans Casual consistency is probabely enough to implemint an electronic stock market. In this case, the changes in stock values should be uniform. Changes in stocks that are independent can be seen in different orders.

Q3. linearizability assumes ~~that~~ the existence of a global clock. However, with strict consistency we showed that such an assumption is not realistic for most distributed systems. Can linearizability be implemented for physically distributed data stores?

Ans Yes, linearizability assumes loosely synchronized clocks. It assumes that events may happen within the same time slot. Those events need to be ranked conforming to sequential consistency.

Chapter 8

Q4. Suppose we have a system with 99.9996% availability, how much downtime a year can it have?

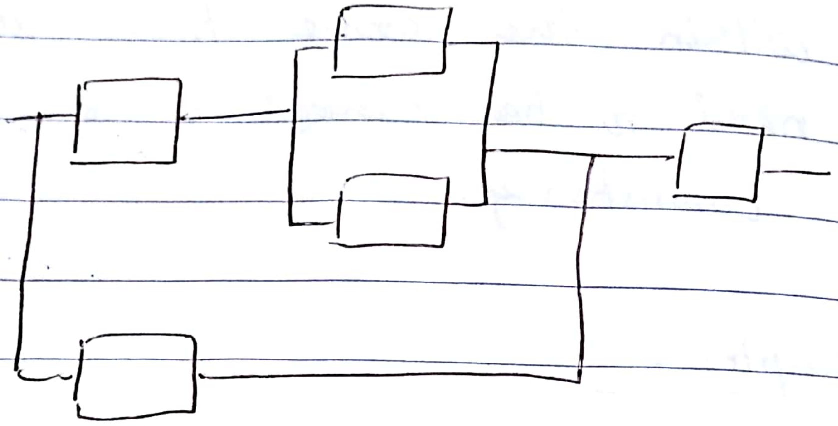
$$\text{Availability} = 99.9996\%$$

$$\text{Downtime} = \frac{100 - 99.9996}{100} \times 365 \times 24 \times 60 \times 60 \text{ sec}$$

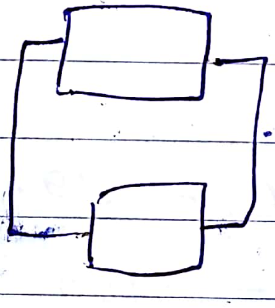
$$= 0.000004 \times 365 \times 24 \times 60 \times 60$$

$$= 126.144 \text{ sec.}$$

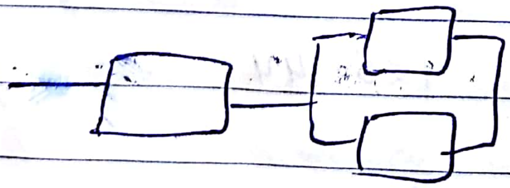
Q5. Write the reliability expression $R_{system}(t)$ of the following Series/parallel system, assuming that each of five modules has a reliability of R(t).



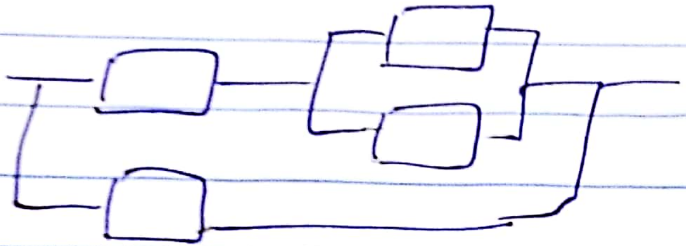
Soln: Starting from small modules :-



∴ for the above module (parallel system) the reliability = 1 – ((1 – R(t)) * (1 – R(t)))

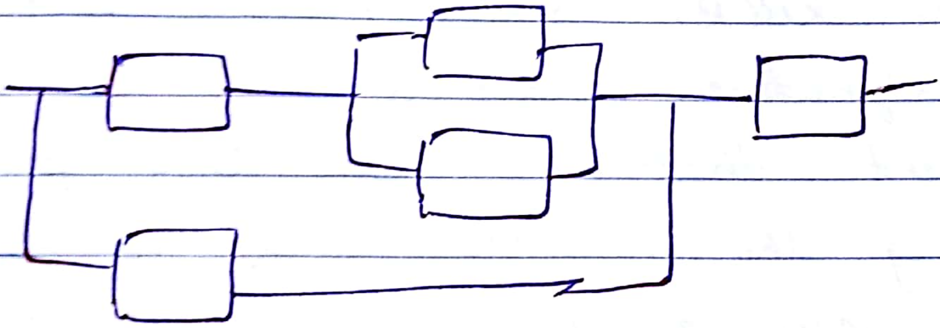


Now reliability = R(t) * (1 – ((1 – R(t)) * (1 – R(t)))



For the above parallel system, the reliability $= 1 - ((1 - R(t)) * (1 - (R(t) *$

$(1 - ((1 - R(t)) * (1 - R(t))))))$



For the above series-parallel system, the reliability $= R(t) * (1 - (1 - R(t)) *$

$(1 - (R(t) * (1 - ((1 - R(t)) * (1 - R(t))))))$

$R\text{system}(t) = R(t) * (1 - (1 - R(t)) * (1 - (R(t) *$

$(1 - ((1 - R(t)) * (1 - R(t))))))$

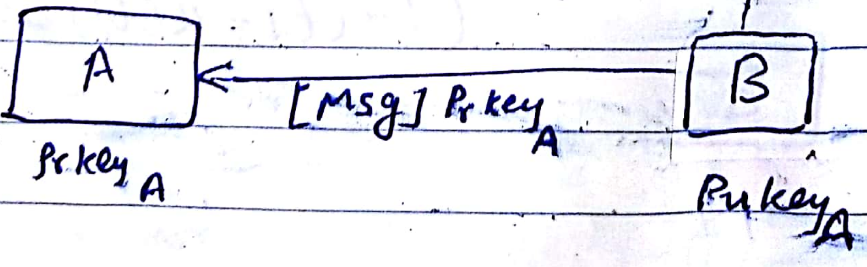
Chapter 9

Q6. Devise a simple authentication protocol using signatures in a public-key cryto-system.

Soln: If A wants to authenticate B, then A sends a challenge R to B. In such case, B will be requested to return KB(R) which means B will send his signature under R. If A is confident that he has B's public key, decrypting the response back to R should be enough for him to know that A is talking to B.

B receives the msg sent by A, To verify that the msg is effectively coming from A, B decrypts it by using A's Public key.

A writes a msg for B. To make sure the msg effectively been written by him, A signs it by encrypting the msg using his private key.



Q7 How are ACL's implemented in a UNIX file system ?

Ans: ACL (Access Control List) implemented in a UNIX file system where each file has three associated entries: One for the owner, one for the group ~~and~~ which is associated with the file and one for everyone else. The access rights can essentially be specified as read, write, execute for each entry.

Chapter 11

Q8. explain whether or not NFS is to be considered a distributed file system.

Ans NFS (Network File System) is not to be considered as distributed file system, but it is actually a protocol that allows local file systems to become accessible to remote clients. Most of the actual file System functionality is not implemented by NFS. Instead, it relies on the Virtual file system interface available in most Operating systems.

Q9. In UNIX-based operating systems, opening a file using a file handle can be done only in the kernel; Give a possible implementation of an NFS file handle for a user-level NFS server for a UNIX system.

Ans: The issue with NFS is to return a file handle that will allow the server to open ~~a file~~ a file using the existing name-based file system interface. One approach is to encode the file name into the file handle. Here the drawback is that once the file name changes, its file handles become invalid.

Q10. Does NFS implement entry consistency?

Ans: Yes, NFS implement entry consistency. The reason behind this is that share reservations & file locking are associated with specific files and a client is forced to revalidate a file when opening it and flush it back to the server when closing it which proves that NFS implements entry consistency.