Chapter 7

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

Each operation then has to obtain a *read quorum* (V_r) or a *write quorum* (V_w) to read or write a data item, respectively. If a given data item has a total of Votes, the quorums have to obey the following rules:

1. $V_r + V_w > V$
2. $V_w > V/2$

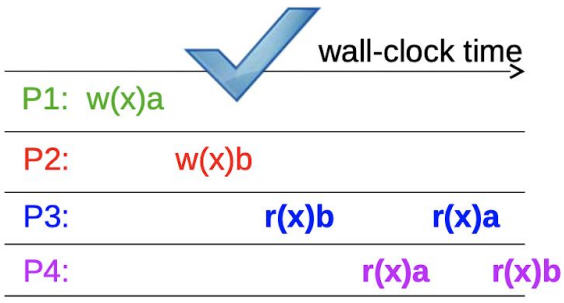
The legal combination for read and write quorum which are permitted are- (1,6), (2,5), (3,4), (2,6), (3,5), (3,6), (4,4), (4,5), (4,6), (5,5), (5,6), (6,6)

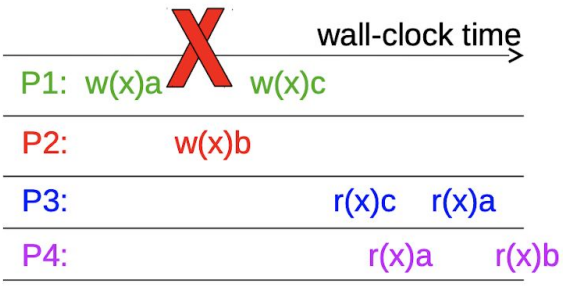
2. What kind of consistency would you use to implement an electronic stock market? Explain your answer ?

Causal Consistency[1]:

Causal consistency: Any execution is the same as if all causally-related read/write ops were executed in an order that reflects their causality . All concurrent ops may be seen in different orders. So everybody should see update of a particular stock value in the same order.

– Reactions to changes in stock values should be consistent.
– Changes in stocks that are independent can be seen in different orders.





3. Linearizability assumes 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?

Yes, Linearizability can be implemented for physically distributed data stores

Linearizability: All processes see all shared accesses in the same order. Accesses are furthermore ordered according to a (non unique) global timestamp .The result of any execution is the same as if the (read and write) operations by all processes on the data store were executed in some sequential order and the operations of each individual process appear in this sequence in the order specified by its program.

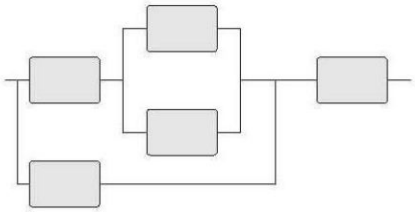
Linearizability assumes loosely synchronized clocks, that is, it assumes that several events may happen within the same time slot. Those events need to be ranked adhering to sequential consistency.

Chapter 8

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

Downtime = ((100- 99.9996)/100) * 365 * 24 * 60 * 60 = 126.144 Seconds/ Year

5. Write the reliability expression Rsystem(t) of the following series/parallel system, assuming that each of the five modules has a reliability of R(t).



The system can be segregated in 2 modules in series . ModuleA leftmost 4 blocks connected in series with ModuleB i.e. the rightmost block . Let's say, the reliability of the leftmost 4 blocks is $R_A(t)$, therefore total systems reliability will be $R_A(t)R(t)$.

We need to calculate the $R_A(t)$ where the blocks are in parallel.

so $R_A(t) = 1 - (1 - R_{Au}(t))\ (1 - R(t))$ where the $R_{Au}(t)$ it the arrangement of top 3 blocks

. Now we need to calculate the $R_{Au}(t)$, where the pattern is in series of one block and two in parallel,

so $R_{Au}(t) = R(t)\ (1 - (1 - R(t))^2) = R(t)(1 – (1 – 2R(t) + R^2\ (t))) = \ 2R^2(t) - R^3(t)$

$R_A(t) = 1 - (1 - R_{Au}(t))\ (1 - R(t))$
$= 1 - (1 - R_{Au}(t))\ (1 - R(t))$
$= 1 - (1 - 2R^2(t) + R^3(t))(1 - R(t))$

$= R(t) + 2R^2(t) - 3R^3(t) + R^4(t)$

$R_{System} = R_A(t)\ R(t) = (R(t) + 2R^2(t) - 3R^3(t) + R^4(t)\ ) \ .\ R(t)$
$= R^2(t) + 2R^3(t) - 3R^4(t) + R^5(t)$

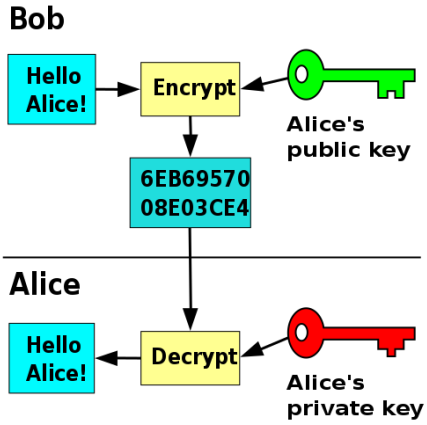
Chapter 9

6. Devise a simple authentication protocol using signatures in a public-key crypto-system.

Public-key cryptography[3], or asymmetric cryptography, is any cryptographic system that uses pairs of keys: public keys which may be shared widely, and private keys which are known only to the owner.

Protocol (For Sending secure message):

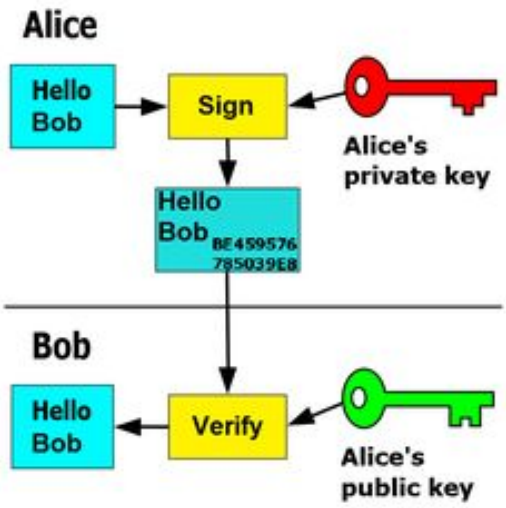
A) Bob wants to send a secure message to Alice ,
B) He will encrypt the message using Alice's Public key .
C) Alice will decrypt the message using her Private key. Now as only Alice has the private key, no one else can retrieve the message .



A second requirement can be just **sending a digitally signed message and not an encrypted message** . i.e. Everyone can read the message , but should be able to verify that it's actually sent by Alice.

Protocol (For Sending unencrypted digitally signed message):

D) Alice wants to send a un encrypted digitally signed message to Bob ,
E) She will encrypt the message using her Private key .
F) Bob will decrypt the message using Alice's Public key. This way , Bob can verify that Alice sent the message and that the message has not been modified



7. How are ACLs implemented in a UNIX file system?

An access control list (ACL) is a table that tells a computer operating system which access rights each user has to a particular system object, such as a file directory or individual file. Each object has a security attribute that identifies its access control list.

Each file has three entries in ACL , namely, owner, user group and the third for every other user. And for these entries we have three access controls which are read only, write , and execute.

Chapter 11

8. Explain whether or not NFS is to be considered a distributed file system.

Network File System (NFS) is a distributed file system **protocol** originally developed by Sun Microsystems in 1984 . It is **not an actual file system**,

NFS is merely a protocol that allows users to access files across a network and treat them as if they resided in a local file directory. The NFS protocol is designed to be independent of the computer, operating system, network architecture, and transport protocol. NFS provides access to shared files through an interface called the Virtual File System (VFS) that runs on top of TCP/IP. Hence **it shouldn't be** considered a distributed file system

9. 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.

NFS provides access to shared files through an interface called the Virtual File System (VFS) that runs on top of TCP/IP. But, in UNIX-based OS, opening a file using a file handle can be done only in the kernel

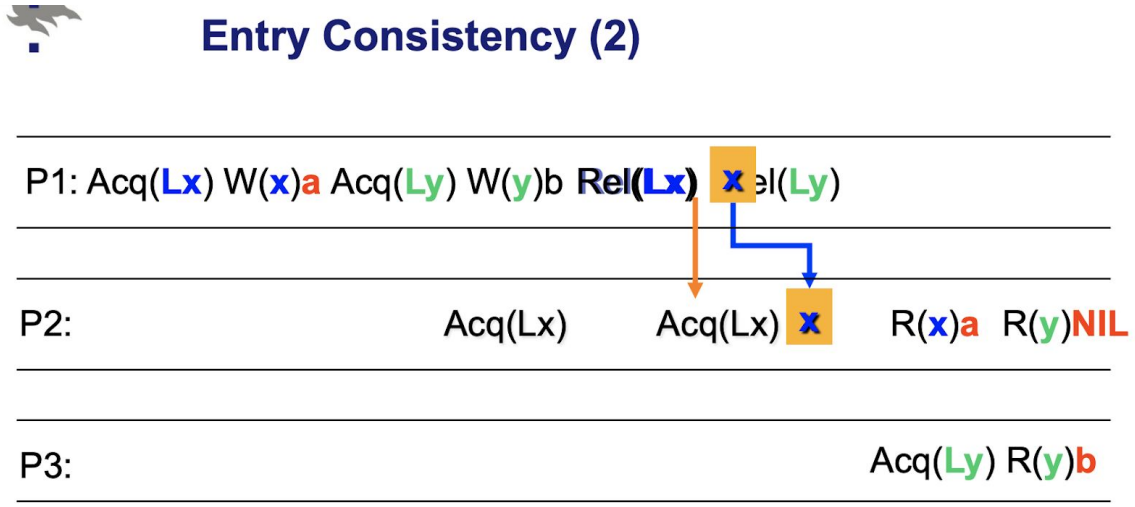
The problem to be solved is to return a file handle that will allow the server to open a file using the existing name-based file system interface. One approach is to encode or somehow piggyback the file name into the file handle.

One possible shortcoming of this approach is that once the file name changes, the file handles become invalid.

10. Does NFS implement entry consistency?

Yes, NFS implements entry consistency, because NFS share file locking and reservations associated with specific files . A client is asked to flush it back to the server when closing a file and re-validate a file while opening it.

Entry Consistency model: Consistency combined with “mutual exclusion” Each shared data item is associated with a synchronization variable S . S has a current owner (who has exclusive access to the associated data, which is guaranteed up-to-date) . Process P enters a critical section: Acquire(S) and retrieve the ownership of S . The associated variables are made consistent Propagation of updates at the next Acquire(S) by some other process.[2]



Bibliography -

1) https://www.cs.princeton.edu/courses/archive/fall16/cos418/docs/P5-consistency.pdf
2) https://www.cs.helsinki.fi/webfm_send/1256
3) https://en.wikipedia.org/wiki/Public-key_cryptography