B. TECH.

TCS – 801/ TIT – 801

EIGHTH SEMESTER EXAMINATION, 2007-08

DISTRIBUTED SYSTEMS

Time : 3 *Hours* *Total Marks* 100

Note : *(i) Attempt all questions.*

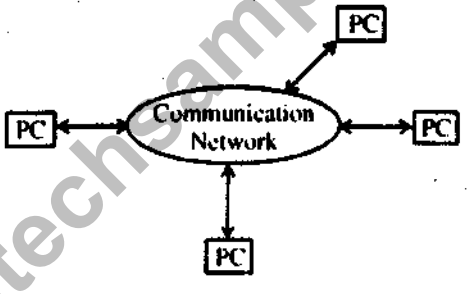
(ii) All questions carry equal marks.

Q.1 Attempt any *four* parts of the following

1. (a) What are Distributed System? Explain its challenges in brief.

Ans. Distributed system is used to describe a system with the following characteristics. It consists of several computers that do not share a memory or a clock.

For example



Challenges of D.S : The main challengies of D.S. are

- Resource sharing
- Enhanced Performance
- Improved Reliability
- Improved Availability
- Modular Expendability

1. (b) What are logical clocks? Why does a logical clock need to be implemented in Distributed Systems? Explain with an example, what are the impacts of absence of global clock and shared memory.

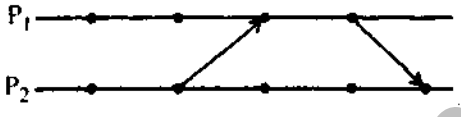
Ans. Logical clock is a mechanism for capturing chronological and causal relationships in a distributed system.

Logical clock algorithms of note are :

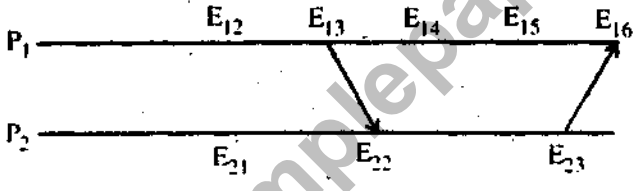
- Lamport timestamps, which are monotonically increasing software counters.
- Vector clocks, that allow for total ordering of events in a distributed system.
- Matrix clocks, an extension of vector clocks that also contains information about other processes' views of the system.

It is implemented to share resources for resources for synchronizing the clocks to avoid the deadlocks.

Example :

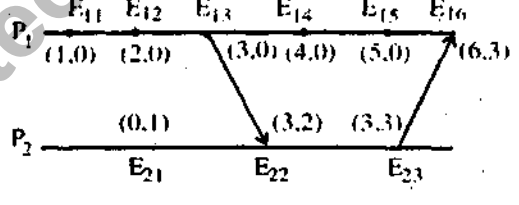


1. (c) Consider the following space time diagram for two processes P_1 and P_2.



Obtain the Lamport time stamp for each event. List the events which casually affect the event E_{22}.

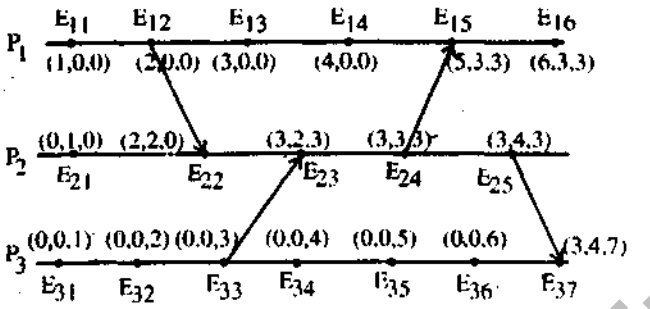
Ans. The lamport-time stamp for each events



1. (d) What do you mean by Casual Ordering of messages? Discuss the salient features of Broadcast based protocol that make the uses of Vector clock which ensures Causual Ordering of messages.

Ans. Casual Ordering of Messages. It is useful for developing the distributed algorithm It automatically arranges the events in a sequence. Each charges in one events may affect the another events automatically only if the casual ordering was applied.

Two protocols are used for casual ordering of message. Consider a broadcast based protocol message ordering.



1. (e) What do you mean by problem of mutual exclusion in Distributed System ? What are the requirements of a good mutual exclusion algorithm? How does the performance of a Distributed algorithm?

Ans. Mutual exclusion, concurrent access to a shared resource by several uncoordinated over-request is serialized to secure the integrity of shared resource. It requires that the action performed by a user on a shared resource must be atomic.

The requirement of mutual exclusion are

1. **Freedom from deadlock** : Two or more sites should not endlessly wait for message.
2. ***Freedom from starvation*** : A site should not-be forced to wait indefinitely to execute CS while other sites are repeatedly executing CS.
3. **Fairness** : It dictates that requests must be executed in the order.
4. **Fault Tolerance** : A mutual exclusion algorithm is fault-tolerant if in the wake of a failure.

1. (f) What are the Token and Non-token based algorithm? Explain Lamport's algorithm with example.

Ans. A token-based algorithm, a unique token is shared among all sites. A site is allowed to enter its CS if it possesses the token. A non-taken based algorithm does not have unique token and site communicates with a set of other sites to arbitrate who should execute the CS ment.

Algorithm

Please see the solution of 1(f), 2004 - 2005

Q. 2. Attempt any *two* of the following :

2. (a) (i) Explain the deadlock handling strategies in distributed system.

(ii) Explain the control organization for Distributed deadlock detection.

Ans. *(i)* See the solution of 2(a) (ii), 2004 - 05

(*ii*) The following are control organization for distributed deadlock detection

1. **Centralized control** : It is responsible for global WFC constantly.
2. **Distributed Control** : Responsible for detecting a global deadlock is shared equally among all sites.
3. **Hierarchiral control** : Responsible for arrange in hierarchial fashion.

2. (b) A centralized global deadlock detector holds the union of local wait-for graphs. Give an example to explain how a phantom deadlock could be detected if a waiting transaction in a deadlock cycle abort during the deadlock detection procedure.

Ans. A centralized global deadlock detector holds the union of local WFG through control site. All sites request and release resource by sending request resource and release resource message to the control site respectively.

For example, suppose two resources R_1 and R_2 are stored at sites S_1 and S_2 respectively Suppose the following two transactions T_1 and T_2 started almost simultaneously at site S_1 and S_4 respectively.

T_1	T_2
lock R_1	lock R_1
unlock R_1	unlock R_1
lock R_2	lock R_1
unlock R_2	unlock R_2

2. (c) (i) What are the shortcomings of Ramamoorthy's two phase algorithm for deadlock detection ?

(ii) Show that Byzantine agreement cannot always be reached among four processors if two processors are faulty.

Ans. (i) The shortcomings of Ramamoorthy's two phase algorithm by selecting only the common transaction found in two consecutive reports, the algorithm gets a consistent view of the system. If a deadlock exists it was argued the same wait for condition must exist in both report. However this chain proved to the incorrect and two phase algorithm may indeed report false deadlock. By getting two consecutive report. The designated site reduces the probability of getting an inconsistent view butdoes not eliminate such a possibility.

(ii) Please see the solution of 2 (c), 2004-05

3. Attempt any *two* of the following :

3. (a) What are the communication models proposed for the distributed objects? Explain the concept of remote method invocation with a suitable example.

Ans. Please see solution of 3(a), 2004-05

3. (b) Discuss how a public key scheme can be used to solve the key distribution problem in a private key cryptographic scheme.

Ans. Please see solution of 3(*b*), (ii), 2004-05

3. (c) Which features of the AFS design make it more scalable than NFS? What are the limits on its scalability, assuming that servers can be added as required?

Ans. AFS has several benefits over traditional networked file systems, particularly in the areas of security and scalability. It is not uncommon for enterprise AFS cells to exceed fifty thousand clients. AFS uses Kerberos for authentication and implements access control lists on directories for users and groups. Each client caches files on the local filesystem for increased speed on subsequent requests for the same file. This also allows limited filesystem access in the event of a server crash or a network outage.

Read and write operations on an open file are directed only to the locally cached copy. When a modified file is closed, the changed portions are copied back to the file server. Cache consistency is maintained by a mechanism called *callback*. When a file is cached, the server makes a note of this and promises to inform the client if the file is updated by someone else. Callbacks are discarded and must be re-established after any client, server or network failure, including a time-out. Re-establishing a callback involves a status check and does not require re-reading the file itself.

A consequence of the file locking strategy is that AFS does not support large shared databases or record updating within files shared between client systems. This was a deliberate design decision based on the perceived needs of the university computing environment. It leads, for example, to the use of a single file per message in the original email system for the Andrew Project, the Andrew Message System, rather than a single file per mailbox.

A significant feature of AFS is the volume, a tree of files, sub-directories and AFS mountpoints (links to other AFS volumes). Volumes are created by administrators and linked at a specific named path in an AFS cell. Once created, users of the file system may create directories and files as usual without concern for the physical location of the volume. A volume may have a quota assigned to it in order to limit the amount of space consumed. As needed, AFS administrators can move that volume to another server and disk location without the need to notify users: indeed the operation can occur while files in that volume are being used.

AFS volumes can be replicated to read-only cloned copies. When accessing files in a read-only volume, a client system will retrieve data from a particular read-only copy. If at some point that copy becomes unavailable, clients will look for any of the remaining copies. Again, users of that data are unaware of the location of the read-only copy; administrators can create and relocate such copies as needed. The AFS command suite guarantees that all read-only volumes contain exact copies of the original read-write volume at the time the read-only copy was created.

The file name space on an Andrew workstation is partitioned into a *shared* and *local* name space. The shared name space (usually mounted as/afs on the Unix file system) is identical on all workstations. The local name space is unique to each workstation. It only contains temporary files needed for workstation initialization and symbolic links to files in the shared name space.

The Andrew File System heavily influenced Version 4 of Sun Microsystems' popular Network File System (NFS). Additionally, a variant of AFS, the Distributed File System (DFS) was adopted by the Open Software Foundation in 1989 as part of their Distributed computing environment.

Available permissions : The following Access Control List permissions can be granted:

Lookup (l) : allows a user to list the contents of the AFS directory, examine the ACL associated with the directory and access subdirectories.

Insert (i) : allows a user to add new files or subdirectories to the directory.

Delete (d) : allows a user to remove files and subdirectories from the directory.

Administer (a) : allows a user to change the ACL for the directory. Users always have this right on their home directory, even if they accidentally remove themselves from the ACL. Permissions that affect files and subdirectories include:

Read (r) : allows a user to look at the contents of files in a directory and list files in subdirectories. Files that are to be granted read access to any user, including the owner, need to have the standard UNIX "owner read" permission set.

Write (w) : allows a user to modify files in a directory. Files that are to be granted write access to any user, including the owner, need to have the standard UNIX "owner write" permission set.

Lock (k) : allows the processor to run programs that need to "flock" files in the directory. Additionally, AFS includes Application ACLs (A)-(H) which have no effect on access to files.

Q 4. Attempt any *two* of the following.

4. (a) The two-phase commit protocol is a centralized protocol where the decision to abort or commit is taken by the co-ordinator. Design a decentralized two-phase commit protocol where no site is designated to be a co-ordinator.

Ans. Two phase commit protocol : The two phase commit protocol is a distributed algorithm which lets all sites in a distributed system agree to commit a transaction. The protocol results in either all nodes committing the transaction or aborting, even in the case of site failures and message losses. However, due to the work by Skeen and Stonebraker, the protocol will not handle more than one random site failure at a time. The two phases of the algorithm are broken into the COMMIT-REQUEST phase, where the COORDINATOR attempts to prepare all the COHORTS, and the COMMIT phase, where the COORDINATOR completes the transactions at all COHORTS.

Assumptions : The protocol works in the following manner: One node is designated the coordinator, which is the master site, and the rest of the nodes in the network are called cohorts. Other assumptions of the protocol include stable storage at each site and use of a write ahead log by each node. Also, the protocol assumes that no node crashes forever, and eventually any two nodes can communicate with each other. The latter is not a big deal since network communication can typically be rerouted. The former is a much stronger assumption; suppose the machine blows up!

Basic Algorithm : During phase 1, initially the coordinator sends a query to commit message to all cohorts. Then it waits for all cohorts to report back with the agreement message. The cohorts, if the transaction was successful, write an entry to the undo log and an entry to the redo log. Then the cohorts reply with an agree message, or an abort if the transaction failed at a cohort node. During phase 2, if the coordinator receives an agree message from all cohorts, then it writes a commit record into its log and sends a commit message to all the cohorts. If all agreement messages do not come back the coordinator sends an abort message. Next the coordinator waits for the acknowledgment from the cohorts. When acks are received from all cohorts the coordinator writes a complete record to its log. Note the coordinator will wait forever for all the acknowledgments to come back. If the cohort receives a commit message, it releases all the locks and resources held during the transaction and sends an acknowledgment to the coordinator. If the message is abort, then the cohort undoes the transaction with the undo log and releases the resources and locks held during the transaction. Then it sends an acknowledgment.

4. (b) Describe how a non-recoverable situation could arise if write locks are released after the last operation of a transaction but before it. commitment.

Ans. Please see solution of 4(a), 2006-07

4. (c) Explain how the two-phase commit protocol for nested transaction ensures that if the top-level transaction commits all the right descendents are committed or aborted.

Ans. A commit operation is, by definition, an all-or-nothing affair. If a series of operations bound as a transaction cannot be completed, the rollback must restore the system (or cooperating systems) to the pre-transaction state.

In order to ensure that a transaction can be rolled back, a software system typically logs each operation, including the commit operation itself. A transaction/recovery manager uses the log records to undo (and possibly redo) a partially completed transaction.

When a transaction involves multiple distributed resources, for example, a database server on each of two different network hosts, the commit process is somewhat complex because the transaction includes operations that span two distinct software systems, each with its own resource manager, log records, and so on. In this case, the distributed resources are the nested transaction. But when the top level transaction commits all the rights of descendents are committed or aborted.

Q. 5. Attempt any *two* of the following :

5. (a) What are Wave and Traversal algorithms? Discuss the usage and application of wave algorithms. Give any three requirements satisfied by wave algorithm.

Ans. Please see solution of 5(a), 2004-05

5 (b) What do you mean by Routing? Discuss the Correctness, Complexity, Efficiency and Robustness criteria of a good routing algorithm.

Ans. Please see solution of 5(b) 2004-05

5 (c) Write short notes on :

(i) CORBA Services

(ii) Election algorithm

(iii) Balanced Sliding Window protocol.

Ans. (i) Please see solution of 5 (c) (i) 2004-05

(ii) Please see solution of 5 (b) (ii) 2006-07

(iii) Please see solution of 5 (c) (iii) 2004-05