Midterm Solutions
ECE 151 / CMPSC 171 - Spring 2013

(10) 1. (a) Explain what we mean by transparency in a distributed system.
(b) Give three different forms of transparency in a distributed system and explain what each of them are.

Transparency is the ability of distributed system to present itself to the user or applications as a single computer system.
Different forms of transparency are:

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may be moved to another location while it is being used
Relocation	Hide that a resource may be moved to another location
Replication	Hide that a resource may be available on several distinct computers
Concurrency	Hide that a resource may be shared by several competing users at the same time
Failure	Hide the failure and recovery of a resource

(14) 2. In distributed systems, multithreading is often used. Discuss the benefits of using multithreading in a
(a) Web client
(b) Web server.

(a) Multithreading enables a web client to spawn multiple threads in parallel to establish connections with more than one web server simultaneously. This is more beneficial when the web client tries to access the web pages stored in different web servers in parallel.
Web client can establish connections with more than one web server to download different portions of the same webpage, with one portion downloaded from each web server. This ensures speedy download of the required data. This technique is being followed in the download of a file by BitTorrent protocol where the client downloads pieces/portions of the same file from different peers in the system.

(**b)** Multithreading enables a web server to service the requests of many web clients in parallel. This ensures that the requests for web pages coming from the web clients are responded to quickly.

(10) 3. Consider a Web server that maintains a table in which client IP addresses are mapped to the most recently accessed Web pages accessed by the clients. When a client connects to the server, the server looks up the client in its table and, if found, returns the Web page listed in the table. Is this server (a) stateless or (b) stateful? Explain why.

The server is stateful. The reason is that it keeps information or state related to the client, in particular, the Web pages that the client most recently accessed. The server can use this information when the client connects again or when the client issues another request, in order to improve performance.

(10) 4. A process P requires access to a file F, which is locally available on the machine on which P is currently running. When P is migrated to another machine, it still requires access to F. If the file-to-machine binding is fixed, how would you implement a system-wide reference to F.

The simplest solution is to create a separate process Q that handles remote requests for the file F. Process P is offered the same interface to F as before, for example in the form of a proxy. Effectively the process Q acts as the file server. When the process P running on a remote machine wants to access the file F, it makes a remote procedure call to the process Q. Q in turn services the file access request of process P. This kind of file access is more or less equivalent to an client requesting for access to a file stored in server's storage disk.

(12) 5. (a) Explain the difference between a name, an address, and an identifier.
(b) Why do we decouple names and addresses in distributed systems?
(c) For each of the strings below, state whether the string can be used as a name, an address, or an identifier of a person and explain under what assumptions it can be so used.
(i) John Brown
(ii) +1-650-322-1234 (cell phone number)
(iii) 123-45-6789 (social security number)

(a) A name refers to an entity in a distributed system; a name is not necessarily unique and it doesn't provide information about the entity's location. An address allows us to locate an entity and provides access to the entity. An identifier is a unique and persistent name of an entity; an identifier refers to at most one entity and an entity has at most one identifier.

(b) We decouple names and addresses in a distributed system, because an entity may have multiple names and multiple addresses. Moreover, the names of an entity or the addresses of an entity may change over time, e.g., because the entity moves around. Decoupling names and addresses provides greater flexibility and ease-of-use. Moreover, it allows us to have separate naming and location services, which can be more efficient.

(c) (i) John Brown is the name of a person. There might be more than one person named John Brown, so John Brown is not an identifier of a person. John Brown is not the address of a person, because it provides no location information about the person.

(ii) +1-650-322-1234 is the address of a person; it can be used to access a person. Typically, we would not refer to a person by his/her cell phone number, so it is not a name. Moreover, it is not an identifier. A person might have more than one cell phone and more than one cell phone number, so it is not unique. Moreover, the person might give up his cell phone number and another person might then acquire it; in such a case, the cell phone number is not persistent.

(iii) 123-45-6789 is an identifier of a person, because it is unique and persistent. We assume that the person's social security number has not been stolen and that the person has not stolen another person's social security number and, thus, that he/she is using two different social security numbers. Typically, we would not refer to a person by his/her social security number, so it is not a name. Moreover, a social security number provides no location information about a person, and we do not access the person by his/her social security number (although we might access information about the person using his/her social security number).

(14) 6. Consider three machines in a distributed system in which the time daemon server is on machine 0. The local clocks on these three machines tick at different rates, and each tick is counted as a second on the machines. The clock tick rates are 60/minute, 40/minute, and 65/minute on machine 0, machine 1, and machine 2, respectively. The time daemon server initiates the Berkeley Algorithm for clock synchronization periodically once its local clock ticks 3600 times. Assume that, at the start, the local clocks of all three machines are synchronized at 8:00am.
(a) What are the local clock times on each of the three machines immediately before invocation of the Berkeley Algorithm the first time (after the start), and what is the synchronized clock time on all three machines after invocation of the Berkeley Algorithm the first time (after the start)?
(b) What are the local clock times on each of the three machines immediately before invocation of the Berkeley Algorithm the second time (after the start), and what is the synchronized clock time on all three machines after invocation of the Berkeley Algorithm the second time (after the start)?

M0= 60 ticks/min (ideal clock)
M1= 40 ticks/min (slow clock)
M2= 65 ticks/min (fast clock)

Initially, clocks synchronized at 8:00am. They will gradually drift apart due to different clock tick rates. Synchronization is initiated **every time** M0 completes 3600 ticks.
M0: 3600 ticks = 3600/60 = 60 min (real time)
Now, we have to find out time in logical clocks at M1 and M2 when real time 60min have passed.
M1: 40 ticks/min * 60 min = 2400 ticks = 40 min (real time)
M2: 65 ticks/min * 60 min = 3900 ticks = 65 min (real time)

(a) Before Berkeley Algorithm is applied for 1st time,

M0= 8:00am + 60 min = 9:00am
M1= 8:00am + 40 min = 8:40am
M2= 8:00am + 65 min = 9:05am

Berkeley's Algorithm calculates the average time on clocks of participating nodes. After applying Berkeley algorithm,

M0=M1=M2= 8:55am

(b) Now after synchronization, all clocks show 8:55am. The clocks again start drifting apart at individual rates.

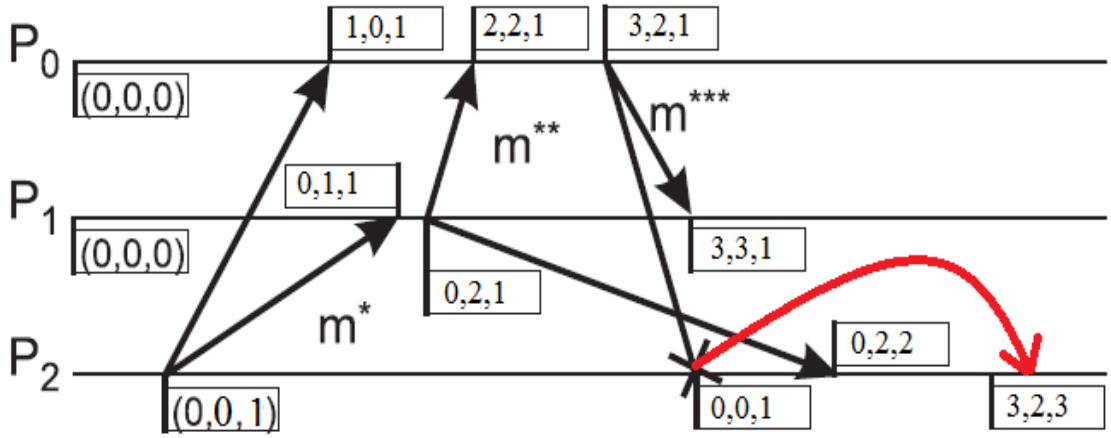
Before Berkeley Algorithm is applied for 2nd time,

M0= 8:55am + 60 min = 9:55am
M1= 8:55am + 40 min = 9:35am
M2= 8:55am + 65 min = 10:00am

Berkeley's Algorithm calculates the average time on clocks of participating nodes. After applying Berkeley algorithm,

M0=M1=M2= 9:50am

(14) 7. A distributed system uses vector clocks for causal ordering of events in the distributed system. In the following diagram, fill in the values of the vector clocks and show the point at which process P2 delivers the message m.**



(16) 8. (a) Explain what mutual exclusion means in a distributed system, and give an example of a shared resource in a distributed system.
(b) Describe the centralized mutual exclusion algorithm that uses a single coordinator.
(c) Consider a scenario in which there are two processes (other than the coordinator), each of which wants to access the shared resource at about the same time. Explain what happens to the second process after the coordinator has granted access to the first process.

(d) Again, consider a scenario in which there are two processes that want to access the shared resource at about the same time. Suppose now that one of the processes wants to access two resources, so that it can make a coordinated update to the two resources. The process initiates two separate requests to the coordinator to access the two resources. What is the potential problem with this method of acquiring the resources? How would you overcome this problem?

(a) Mutual exclusion refers to the phenomenon by which only one of the members in the distributed system has access to a shared resource at a time. By way of mutual exclusion, the access of a shared resource is restricted to one member at a time. This ensures that the members of the distributed system have a unified and a consistent view of the shared resource. An example of a shared resource can be a web page stored in a web server.
(b) As per this algorithm there exists one member in the system called the coordinator which controls the access of the other members to the shared resources in the system. The other members in the system need to obtain the permission of the coordinator in order to access any shared resource in the system. Only after it has been granted access by the coordinator a member can have access to the shared resource. After finished accessing the resource the member has to have a way of informing this to the coordinator. The basic functionality of the coordinator is to listen to access requests of the members and to grant them access sequentially based on any fixed priority or on first-come-first-served basis.
(c) The second process has to wait and its access request remains unresponded in the request queue of the coordinator while the first process goes on to access the resource. Internally the second process may either block or continue to do some other task till it obtains the permit to access. After the first process has finished accessing the resource it informs the coordinator that it no longer needs the exclusive access to the shared resource. At this point the coordinator responds to the access request of the second process and grants it access after which the second process goes on to access the shared resource.
(d) One process may have access to one resource and the other process may have access to other resource at the same time. This leads to a deadlock since a process requires both the resources to make a coordinated update. This leads to a situation where each of the two processes lie idle waiting for access to the other resource which is in the other process's kitty. Many possible solutions can be thought of to overcome this problem and one of them is,

1. One possible solution is to have a time-out mechanism by which a process gives up the right of access to a resource in case if it lies idle for a particular period of time without doing anything with the resource. This may lead to a situation where the processes re-initiate their requests for accessing the shared resources. This sequence of steps may continue as loops until one of the processes gets accesses to both the resources.