

程序代写代做 CS编程辅导



Synchronization

WeChat: cstutorcs

Assignment Project Exam Help

Lecturer: Dr. Joseph Doyle
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Introduction

程序代写代做 CS编程辅导

- Events occur at different nodes in the network
- These events are recorded using different clocks
- How do other nodes in the network determine when events took place



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Why is it important

程序代写代做 CS 编程辅导

- May seem like a trivial issue but actually very important
- How can we determine if data is up-to-date unless we can determine the order of writes to the data
- Is it possible to sell stock that does not exist?



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Synchronize Clocks

程序代写代做 CS 编程辅导

- Simplest solution would be to synchronise clocks on every node in the network
- In practice this is very difficult to achieve

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Hardware Clocks

程序代写代做 CS 编程辅导

- Hardware clock computers are crystal oscillators which are connected to oscillation counter circuitry
- This counter is then scaled to provide an approximate representation of physical time

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 740389476

<https://tutorcs.com>

Problems with Hardware Clocks

程序代写代做CS编程辅导

- Clocks are not all exactly the same
- The clock rate is affected by
 - Physical differences in the crystal
 - Temperature differences
 - Voltage
 - Humidity



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Problems with Hardware Clocks

程序代写代做CS编程辅导



- This results in the following problems:

- **Skew:** This is a disagreement in the reading of two clocks
- **Drift:** This is the difference in the rate at which two clocks count the time
- **Clock Drift Rate:** This is the difference in precision between a hardware clock and a perfect reference clock.
Approximately 10^{-6} sec/sec in normal clocks and $10^{-7} - 10^{-8}$ sec/sec in high precision clocks

WeChat: cstutorcs

Assignment Project Exam Help

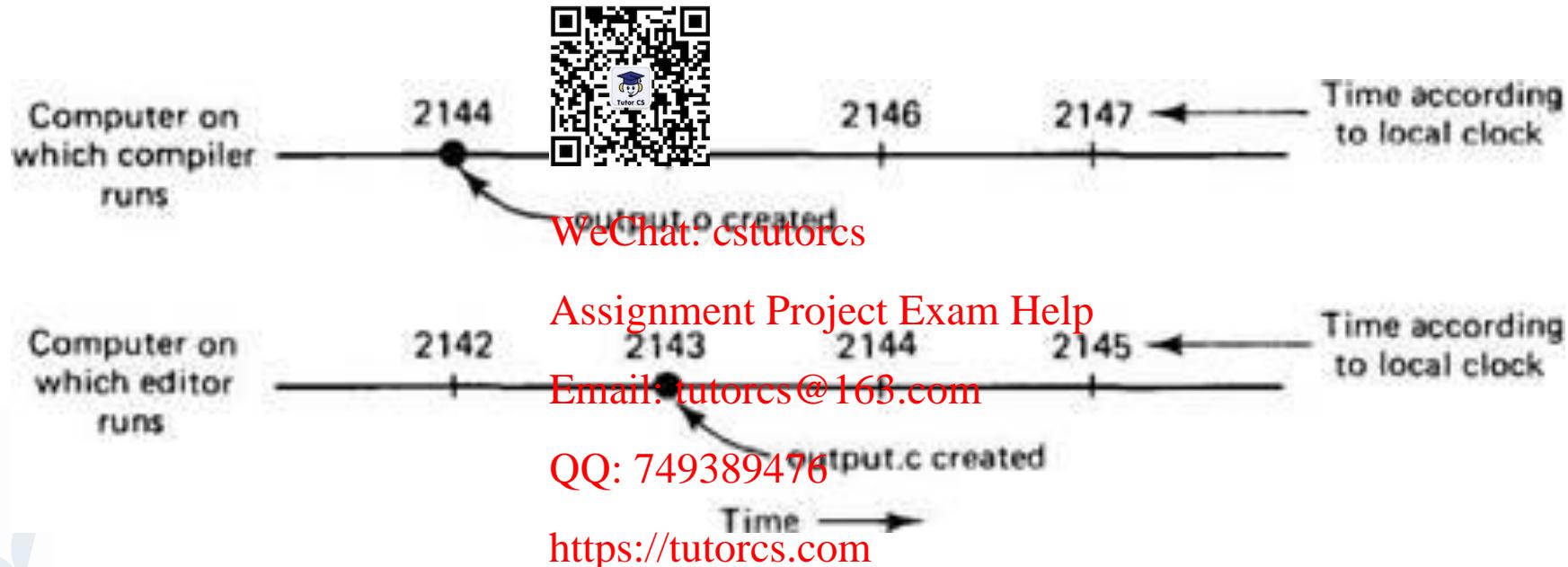
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Clock Synchronization

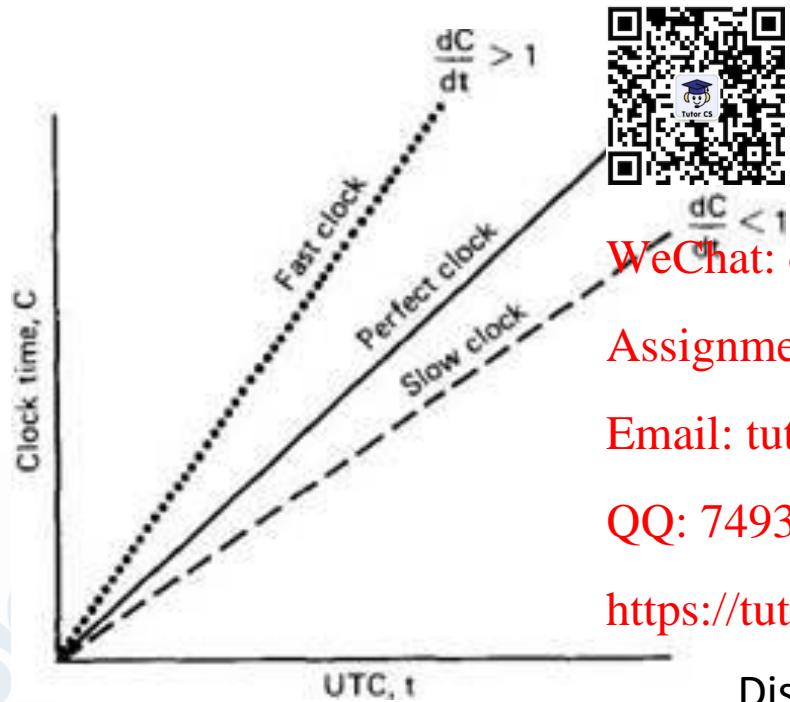
程序代写代做 CS 编程辅导



Distributed Operating Systems. Andrew Tanenbaum

Clock Synchronization

程序代写代做 CS 编程辅导



WeChat: cstutors

- Assuming a normal clock drift rate of 10^{-6} sec/sec a drift of 1ms occurs in approximately 17 minutes and a drift of 1s occurs in approximately 12 days

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Distributed Operating Systems. Andrew Tanenbaum

Synchronizing Clocks

程序代写代做 CS 编程辅导

- External Synchronization

- An external authoritative clock updates the clock of nodes in the network
- Skew is limited to the delay between the authoritative clock server and the node



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

- Internal Synchronization

QQ: 749389476

- Nodes in the network collaborate to limit the skew to a delay bound
- Delay is larger here as a round trip is required

Software Based Clock Synchronization

程序代写代做 CS 编程辅导

- Christian's Algorithm
- Berkley Algorithm
- Network Time Protocol (Internet)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Christian's Algorithm

程序代写代做 CS编程辅导

- Based around the observation that round trip time in LAN networks is sufficiently small to ensure reasonable clock accuracy
- It requires a clock server which is synchronized to UTC time

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Christian's Algorithm

程序代写代做 CS编程辅导

- A node in the network sends a request to the clock server and records the round trip time RTT
- The clock server measures its current time T and sends this to the node
- The node then updates its time t to be $t=T+RTT/2$

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

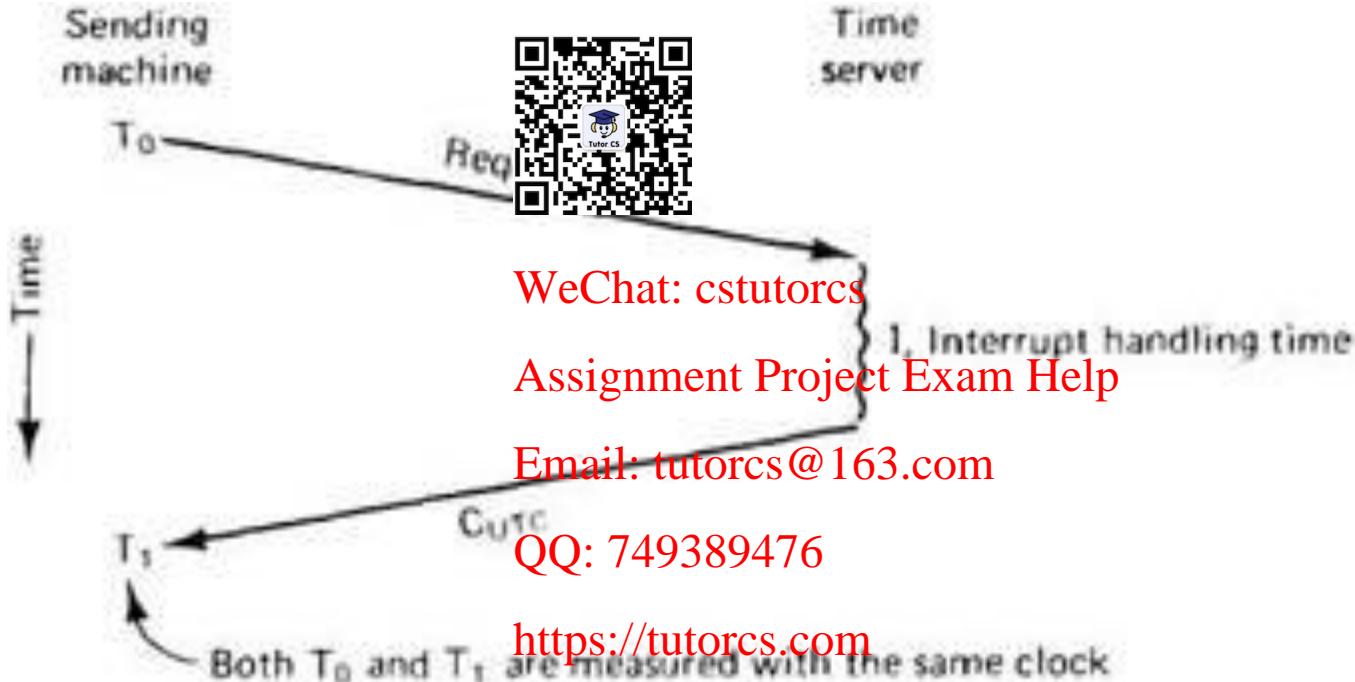


@QMUL



Christian's Algorithm

程序代写代做 CS 编程辅导



Distributed Operating Systems. Andrew Tanenbaum

Christian's Algorithm Problems

程序代写代做 CS 编程辅导



- Assumes that the propagation of the two parts of the round trip are equal
- Not really suitable outside a LAN as the RTT can increase dramatically
- The clock server is a central point of failure

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Berkeley Algorithm

程序代写代做 CS 编程辅导

- Variation of Chandy-Lamport's Algorithm which does not require a clock server that is synchronized to UTC time
- Uses a manager server instead of a clock server to alter the clocks of nodes in the network



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Berkeley Algorithm

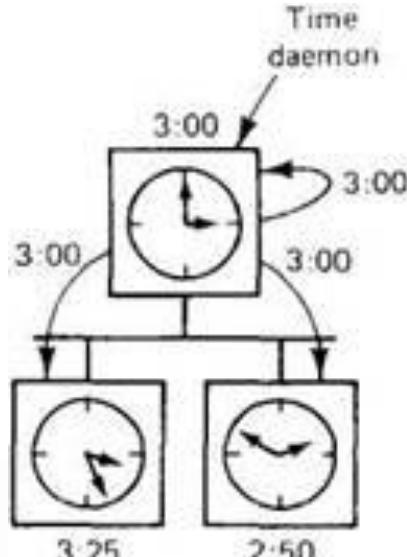
程序代写代做 CS 编程辅导

- Manager server periodically polls all nodes in the network
- It records the RTT and uses this to estimate the clock of the node in a similar fashion to Christian's algorithm
- It averages the values obtained from all nodes
- It instructs the nodes to alter their clocks based upon this average to synchronize the times of the nodes
- If the manager fails a new manager is elected using a manager election algorithm (This is discussed later)

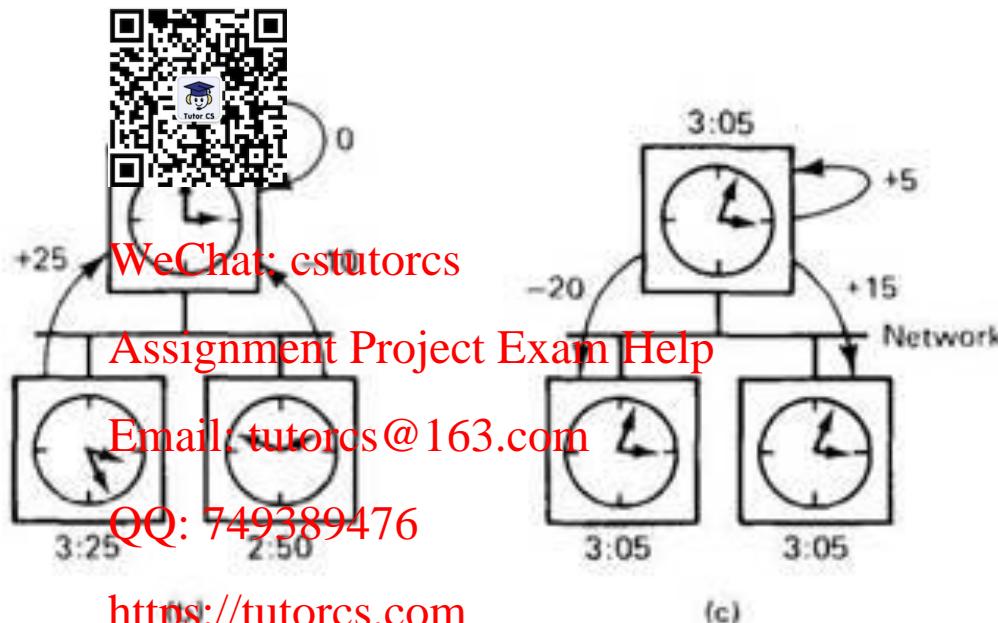


Berkeley Algorithm

程序代写代做 CS 编程辅导

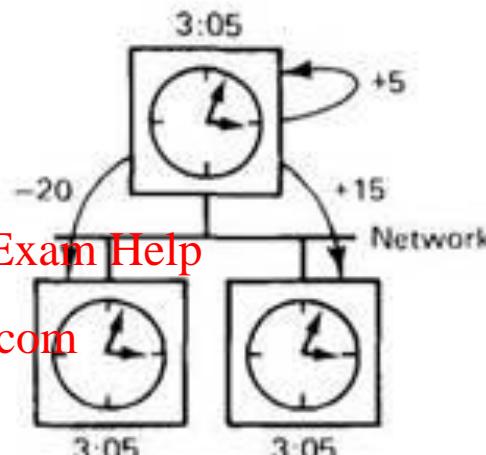


(a)



<https://tutorcs.com>

(b)



(c)

Distributed Operating Systems. Andrew Tanenbaum

Berkeley Algorithm Problems

程序代写代做 CS编程辅导

- Again this is really suitable for LAN networks
- Also while the nodes are synchronized with each other they are not synchronized with external systems if this is required as is the case with Christian's algorithm



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Network Time Protocol (NTP)

程序代写代做 CS编程辅导

- The goal of this protocol is to improve on the previously discussed algorithms to allow:
 - The ability to synchronize clients via the Internet
 - Reliable service in the event of lengthy losses of connectivity
 - Provide protection against interference



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutores@163.com

QQ: 749389476

<https://tutorcs.com>

Network Time Protocol (NTP)

程序代写代做 CS编程辅导

- NTP uses a hierarchical model with different stratum in the hierarchy assigned different levels of accuracy
 - **Stratum 0** are high precision timekeeping devices such as atomic clocks, GPS or other radio clocks. They are known as reference clocks
 - **Stratum 1** are computers which are synchronised to with a few microseconds of a Stratum 0 devices. These are the primary time servers.
 - **Stratum 2** are nodes connected to primary time servers. It is possible to connect to multiple primary time servers



WeChat: cstutors

Assignment Project Exam Help

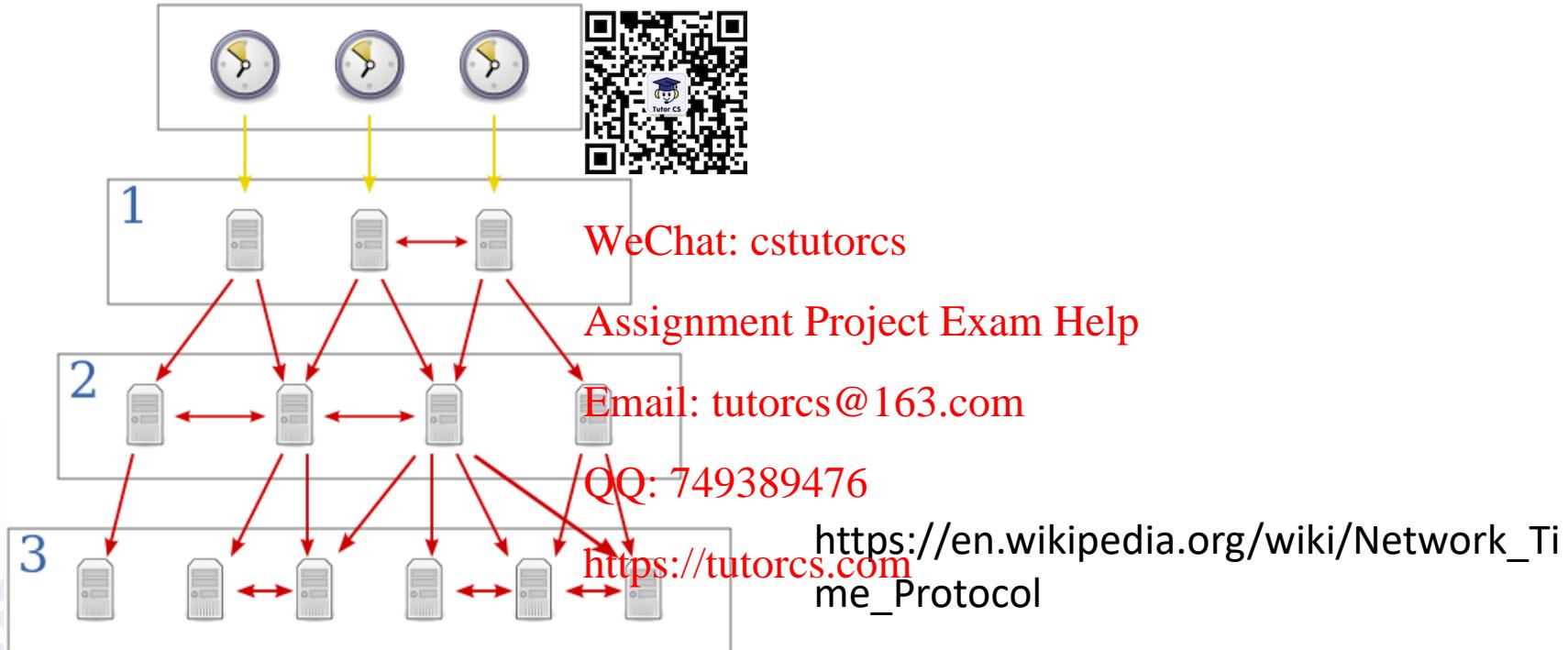
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Network Time Protocol (NTP)

程序代写代做 CS编程辅导



Network Time Protocol (NTP)

程序代写代做 CS编程辅导

- Up to 15 stratum possible in NTP
- The higher the stratum number the less accurate the clock of the node due to the increased RTT



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

NTP Timestamps

程序代写代做 CS编程辅导

- Uses a 64 bit time  with 32 bits for seconds and 32 bits for fractions of seconds
- It has a theoretical resolution of 233 picoseconds
- It has a timescale that rolls over every 136 years. First rollover will occur on February 7th 2036 (Something to look forward to)

WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

NTP Algorithm

程序代写代做CS编程辅导

- The NTP client will regularly poll one or more servers



WeChat: cstutorcs

- Using this information they will calculate the time offset θ and round trip delay δ as:

$$-\theta = \frac{(t_1 - t_0) + (t_2 - t_3)}{2}$$

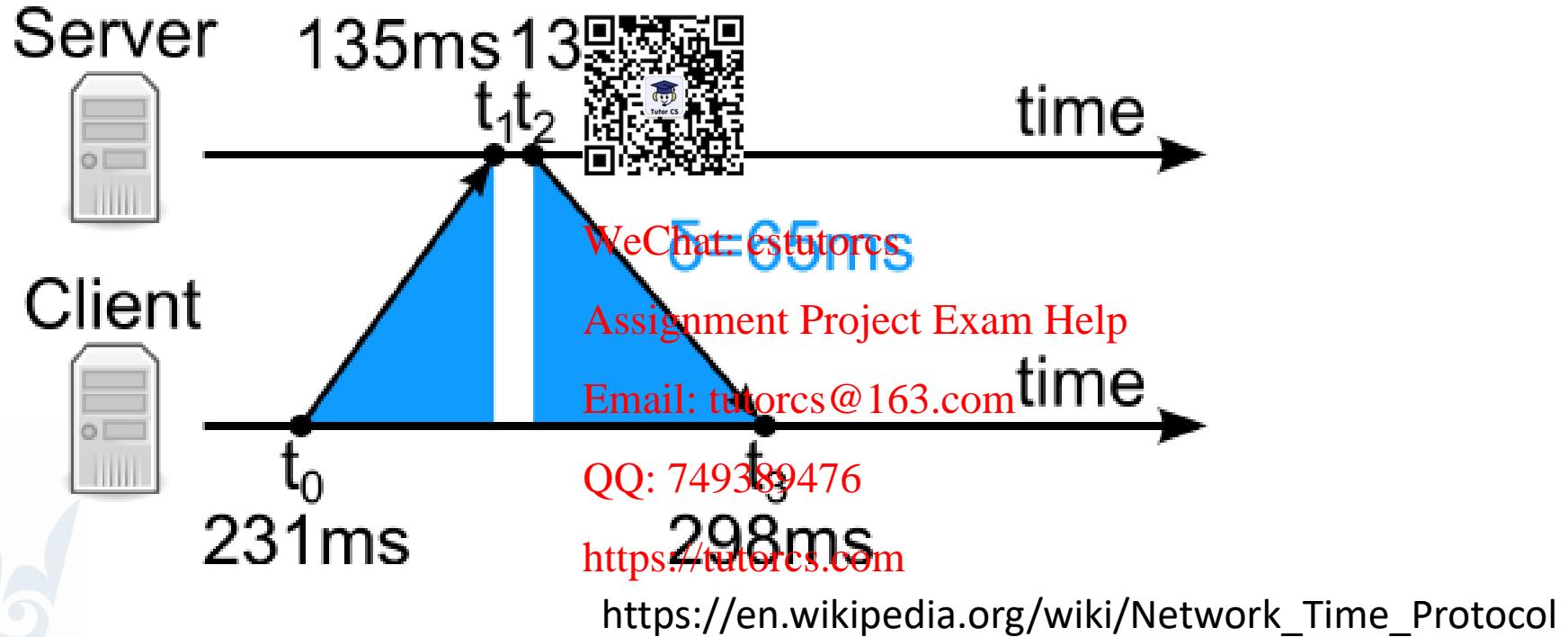
Assignment Project Exam Help

Email: tutorcs@163.com

$$-\delta = (t_3 - t_0) - (t_2 - t_1)$$

NTP Algorithm

程序代写代做 CS 编程辅导



NTP Algorithm

程序代写代做CS编程辅导

- Once these values are returned to the client they are passed through a statistical filter to eliminate outliers
- An estimate of the offset is calculated from the best three remaining candidates (it favours messages from higher stratum)
- The client's clock is then adjusted gradually to reduce the offset



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Synchronization Requirements

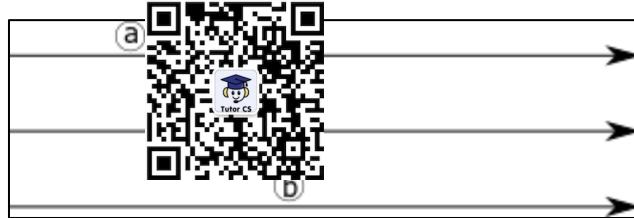
程序代写代做 CS编程辅导

- **Causality:** real-time order ~ timestamp order
- **Groups/replicated:** All members of the group see events in the same order
WeChat: cstutorcs
- **Multiple-copy-updates:** order of the updates,
Assignment Project Exam Help
Email: tutorcs@163.com
consistency conflicts
QQ: 749389476
- **Serializability of transactions:** Common order of transaction order
<https://tutorcs.com>



Happened-Before Relations (a->b)

程序代写代做 CS编程辅导



WeChat: cstutorcs

- a and b are defined as events in the same process
[Assignment Project Exam Help](#)
- If a occurs before b then a->b
- For example if a is a message being sent and b is a message being received
then a->b
[Email: tutorcs@163.com](#)
[QQ: 749389476](#)
- a | b if neither a->b and b->a (a and b are concurrent)
<https://tutorcs.com>
- If a->b and b->c then a->c

Lamport Timestamps

程序代写代做 CS 编程辅导

- The rules for this algorithm are as follows:

- A process increments its counter before each event that it processes
- The process includes this counter when it sends a message
- On receiving a message the counter of the recipient is updated if necessary to the greater of the its current counter and timestamp received in its message. The counter is incremented by one to indicate that the message has been received.

<https://tutorcs.com>

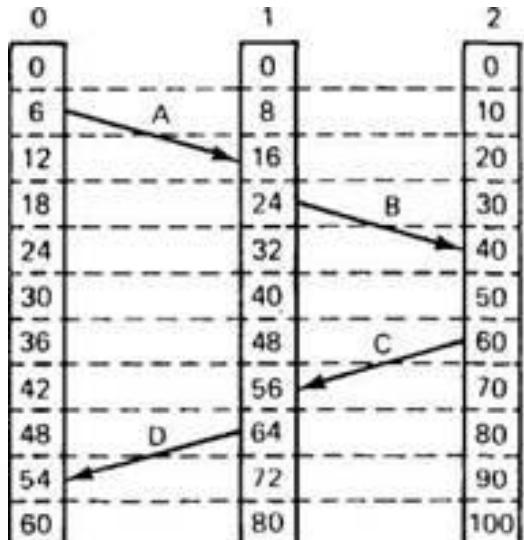
Lamport Timestamps

程序代写代做 CS 编程辅导

- If we define a process  an event as e , a counter as L_i and a timestamp as $L_i(e)$ can define this algorithm as
- At p_i : before each event $L_i = L_i + 1$
- When p_i sends a message m to p_j
 - $p_i: L_i = L_i + 1; t = L_i$; [Email: tutorcs@163.com](mailto:tutorcs@163.com)
 - $p_j: L_j = \max(L_j, t); L_j = L_j + 1$ [QQ: 749389476](#)
 - $L_j(\text{receive event}) = L_j$ <https://tutorcs.com>

Lamport Timestamps

程序代写代做CS编程辅导



(a)

(Uncorrected)



(b)

<https://tutorcs.com>
(Lamport Timestamps)

Distributed Operating Systems. Andrew Tanenbaum

Lamport Timestamps Problems

程序代写代做 CS 编程辅导

- Suppose there are two events a and b
- If there is any way that event a has influenced event b then we can state that $a \rightarrow b$ and the Lamport timestamp of event a is less than the Lamport timestamp of event b
 - WeChat: cstutorcs
- However, if we have two Lamport timestamps $L(a) < L(b)$ we cannot explicitly state that $a \rightarrow b$ as Lamport timestamps do not fulfil the **strong clock consistency condition**
 - Email: tutorcs@163.com
- Lamport timestamps only create a **partial ordering** of events
 - QQ: 749389476
- They can be used to create a **total ordering** of events by including an arbitrary mechanism to break ties (albeit with the caveat that this cannot be used to imply a causal relationship)
 - <https://tutorcs.com>



Total Ordered Lamport Timestamps

程序代写代做 CS 编程辅导

- Expand Timestamp $L_i(e), i$
- $(L_i(e), i) < (L_j(e), j)$
- If $L_i(e) < L_j(e)$ or $(L_i(e) = L_j(e) \text{ and } i < j)$



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Total Order Multicasting Examples

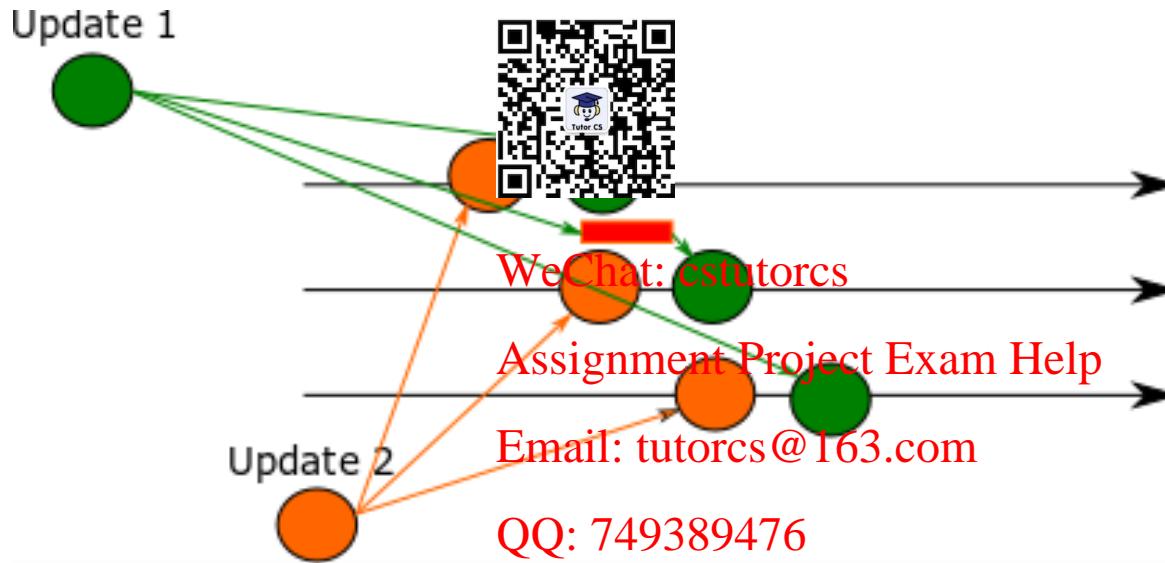
程序代写代做 CS编程辅导



Database is updated leaving an inconsistent state
<https://tutorcs.com>

Total Order Multicasting Examples

程序代写代做 CS编程辅导



All receivers see all messages in the same order
(which is not necessarily the original sending order)
E.g. Group Updates

Vector Clocks

程序代写代做 CS 编程辅导

- Vector clocks expand Lamport Timestamps by including additional data for detecting causality violations
- Each Process P_i maintains a vector V_i
- $V_i[i]$ is the number of events that have occurred at P_i
- If $V_i[j] = k$ is then P_i knows about the k events that have occurred at P_j



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Vector Clocks

程序代写代做 CS编程辅导

- Order of timestamps
- $V = V'$ iff $V[j] = V'[j]$
- $V \leq V'$ iff $V[j] \leq V'[j]$ for all j
- $V < V'$ iff $V \leq V'$ and $V \neq V'$
- Order of events
- $e \rightarrow e' \Rightarrow V(e) < V(e')$
- $V(e) < V(e') \Rightarrow e \rightarrow e'$
- $e || e' \text{ if } \neg(V(e) \leq V(e'))$
and $\neg(V(e') \leq V(e))$



Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Vector Clocks

程序代写代做 CS编程辅导

Time

A
A:0

Cause

A:1 A:2
B:2 B:2
C:1 C:1

B
B:0

B:1 B:2
C:1 C:1

C
C:0 C:1

Independent



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476
<https://tutorcs.com>

Effect



https://en.wikipedia.org/wiki/Vector_clock

Vector Clocks

程序代写代做 CS编程辅导

- The algorithm for vector V_i is as follows:
- P_i multicast $V_i[i] = V_i[i]$
- Each message includes V_i
- For each P_j which is receiving a message:
The message can be delivered
when
 - $V_i[i] = V_j[i] + 1$ (All previous messages from i have arrived)
 - $V_j[k] \geq V_i[k]$ for all k , $k \neq i$ (j has seen all the messages i has seen when the message was sent)
- Upon delivery of message $V_j[i] = V_j[i] + 1$



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Global State

程序代写代做 CS编程辅导

- Timestamps can be used for a variety of purposes such saving a snapshot of a distributed system
- To create a snapshot the system needs information on:
 - The state of processes
 - Messages in transfer
- There are potential problems which can affect this snapshot namely:
 - Garbage Collection
 - Deadlock
 - Termination



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Garbage Collection

程序代写代做 CS编程辅导

- Garbage collection is an automatic form of memory management
- Very important on distributed systems as manual memory management is more difficult when using multiple systems
- For example consider a simple distributed messaging system
- Messages must be kept until they can be processed after which they can be discarded by a garbage collector
- Need to consider timestamps of messages to determine if they should be included in a snapshot or if they should be discarded

<https://tutorcs.com>



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

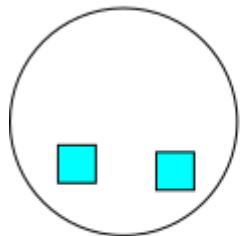
QQ: 749389476

Garbage Collection

程序代写代做 CS编程辅导



P1



P2

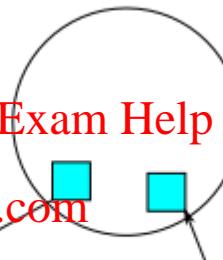
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

Message Object
<https://tutorcs.com>



Garbage Object

Deadlock

程序代写代做 CS 编程辅导

- Deadlock occurs when two processes are waiting for each other to take an action such as releasing a lock which is associated with a resource
- This can occur for a variety of reasons. Consider the following example
- P_1 has a lock on resource R_1 and P_2 has a lock on resource R_2 . P_1 needs to obtain a lock on R_2 before it will release R_1 and P_2 needs to obtain a lock on R_1 before it will release R_2 .
- Unless action is taken to resolve the deadlock P_1 and P_2 will continue to wait for a lock they cannot obtain.
- It would be useful to resolve deadlocks before taking a snapshot as they prevent progress in distributed systems



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Deadlock

程序代写代做 CS编程辅导



<https://tutorcs.com>

<https://en.wikipedia.org/wiki/Deadlock>

Global Snapshot

程序代写代做 CS编程辅导

- To create a global snapshot the following algorithm can be used:
- At each node i
 - A local clock records the time T_i
 - A state S_i is recorded as a list of tuples containing {event, timestamp}
- The system state S is defined as a vector of the state of each node S_i
- A snapshot will contain information on each node up to time T
- A snapshot can be considered consistent or inconsistent



WeChat: cstutorcs

Assignment Project Exam Help

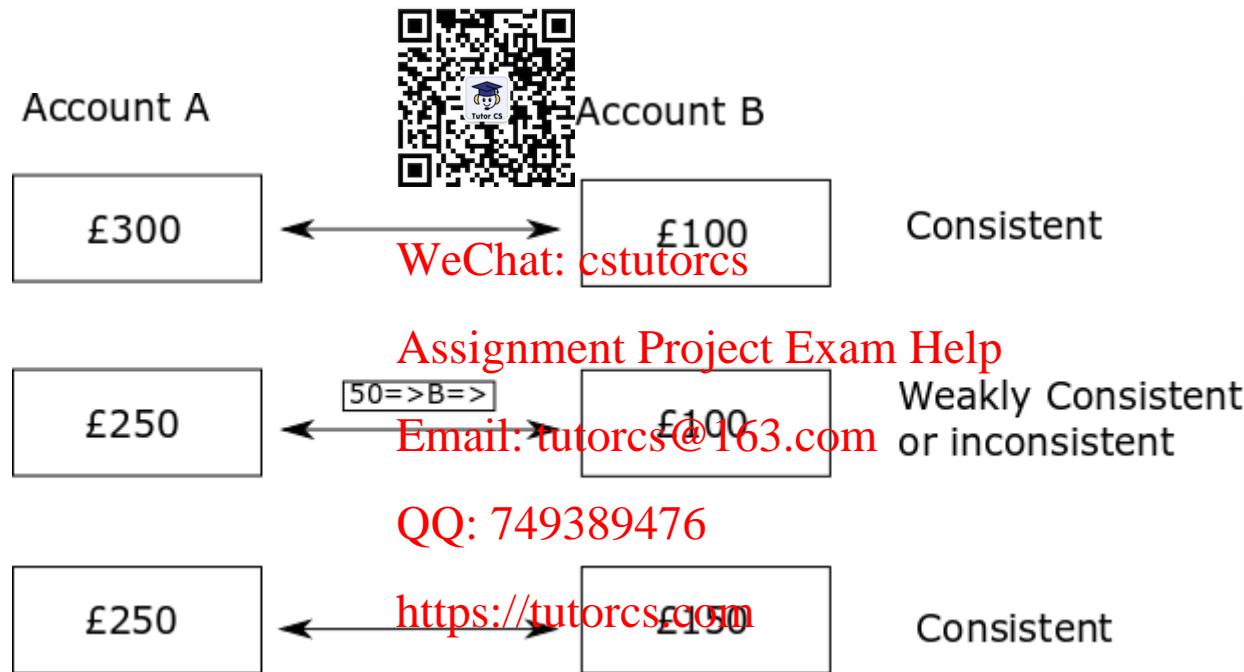
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Inconsistent Snapshot Example

程序代写代做 CS 编程辅导



Consistent Snapshot

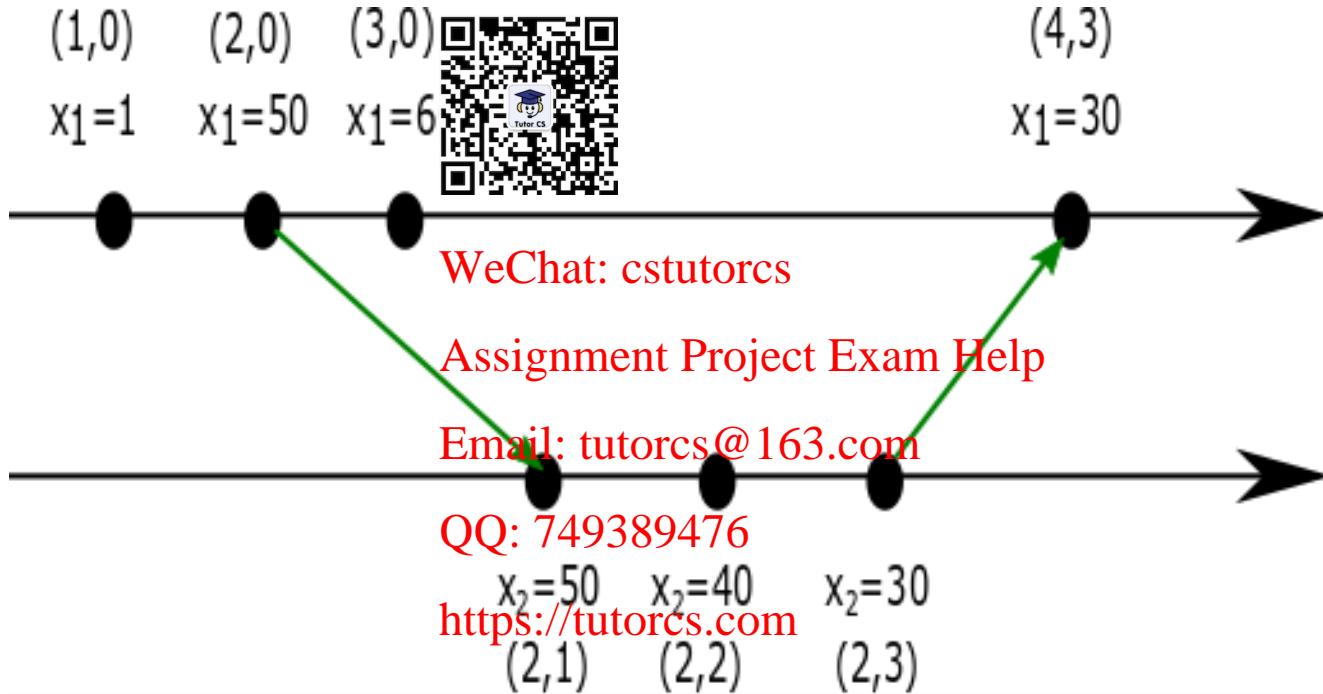
程序代写代做 CS 编程辅导

- A snapshot is consistent if it contains all the events which happened before the snapshot time T
- Consider the example where changes to a local value x_i at processor i only need to be sent to another processor when they exceed a certain threshold (>19)
 - WeChat: estutors
 - Assignment Project Exam Help
 - Email: tutorcs@163.com
 - QQ: 749389476
 - <https://tutorcs.com>
- A vector clock is incremented when x_i changes
- S records all changes to the x_i value



Consistent Snapshot

程序代写代做 CS 编程辅导



Chandy Lamport Algorithm

程序代写代做 CS编程辅导

- To create a consistent initial state of a distributed system the Chandy-Lamport Algorithm can be used
- The algorithm assumes that:
 - There are no network failures and all messages arrive intact
 - The snapshot algorithm does not interfere with the normal operation of the processes
- It would be possible to modify this algorithm with TCP/IP to relax the no network failures assumption



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Chandy Lamport Algorithm

程序代写代做 CS编程辅导

- The process which is initiating the snapshot process:
 - Saves its own state
 - Sends a snapshot request to all processes bearing a snapshot token
- When a process receives a snapshot request it
 - Sends the snapshot process its saved state
 - Attaches the snapshot token to all subsequent messages
- When a process receives a message that does not have the snapshot token when it has received the snapshot token it forwards the message to the snapshotting process so that it can be included in the snapshot



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutordes@163.com

QQ: 749389476

<https://tutorcs.com>

Resource Reservation

程序代写代做 CS 编程辅导

- Timestamps are also used for reserving resources
- The process which controls resource allocation can be centralized or distributed
- Centralized in general is easy to implement but are not fault tolerant
- Distributed processes more complicated but are in general more fault tolerant
WeChat: cstutorcs
Assignment Project Exam Help
Email: tutorcs@163.com
- The other disadvantage of distributed solution is that it tends to be more difficult to debug
QQ: 749389476
<https://tutorcs.com>
- This is sometimes referred to as Mutual Exclusion



Centralized Resource Allocation

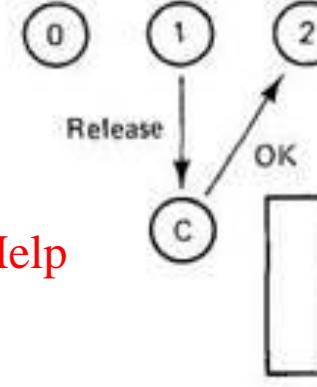
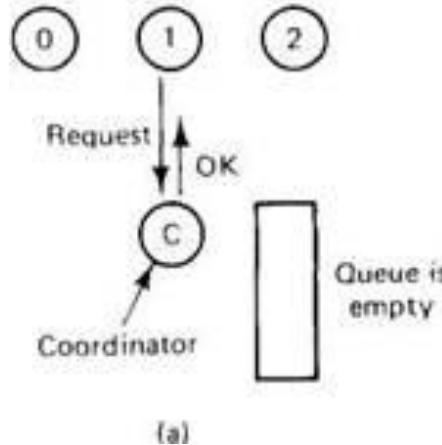
程序代写代做 CS编程辅导

- In the central case a process asks the coordinator for access to the resource
- The coordinator maintains a queue and if the queue is empty it will grant access to the process WeChat: cstutorcs
- If another process contacts the coordinator it is placed in the queue and the coordinator does not respond to the request Email: tutorcs@163.com
- When the first process finishes it informs the coordinator and the coordinator then allows the next process in the queue access to the resource QQ: 749389476
<https://tutorcs.com>



Centralized Resource Allocation

程序代写代做 CS编程辅导



<https://tutorcs.com>

Distributed Operating Systems. Andrew Tanenbaum

Centralized Resource Allocation

程序代写代做 CS编程辅导

- The general requirements of mutual exclusion are:
- **Safety:** Only one process has access to the resource at any given time (The algorithm fulfils this requirement)
- **Liveness:** All requests eventually succeed. Deadlock is prevented. (This requirement is not specifically fulfilled. Several methods can be used to break deadlock with the simplest being timeout functionality)
- **Fairness:** If request A happens before request B then A is honoured before B (Timestamps play a role in this requirement as request A may have been delayed in the network but it should still be honoured before request B)



WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Distributed Resource Allocation

程序代写代做 CS编程辅导

- Also possible to use a distributed scheme
- When a process wants a resource it multicasts a request to all process and waits for the response
- When a process receives a request if it does not want the response it responds immediately
- If it is using the resource or wants to use the resource and the timestamp of its request is lower than the received timestamp it puts the request in a queue
- When it finishes using the request it replies to all requests in the queue



WeChat: cstutorcs

Assignment Project Exam Help

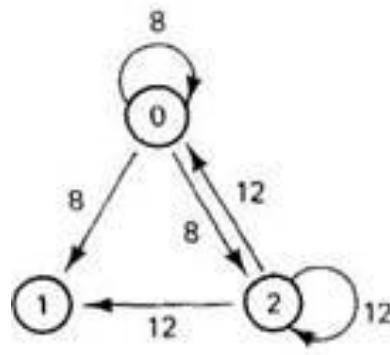
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Distributed Resource Allocation

程序代写代做 CS 编程辅导



(a)



Tutor CS
Enters critical region

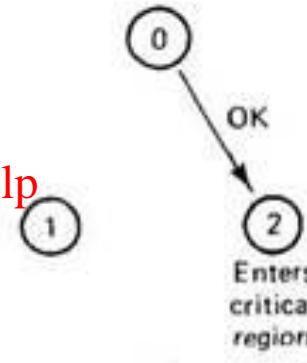
WeChat: cstutorcs

Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

(b)



(c)

<https://tutorcs.com>

Distributed Operating Systems. Andrew Tanenbaum

Token Ring Resource Allocation

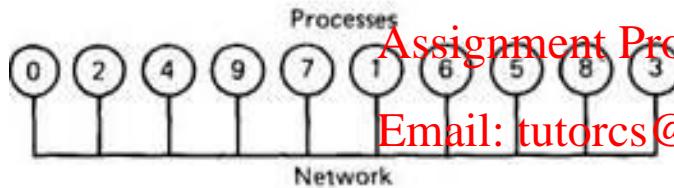
程序代写代做 CS 编程辅导

- Also possible to use a logical token ring structure
- When the algorithm is implemented process 0 has a token
- The process passes the ring to its neighbour in the token ring structure
- When the process receives the token it checks to see if it wants the resource
 - Assignment Project Exam Help
- If it does it utilises the resource and then releases the token
- A process will have to wait until every other process in the token utilises the resource in the worst case scenario

<https://tutorcs.com>



Token Ring Resource Allocation



WeChat: cstutorcs

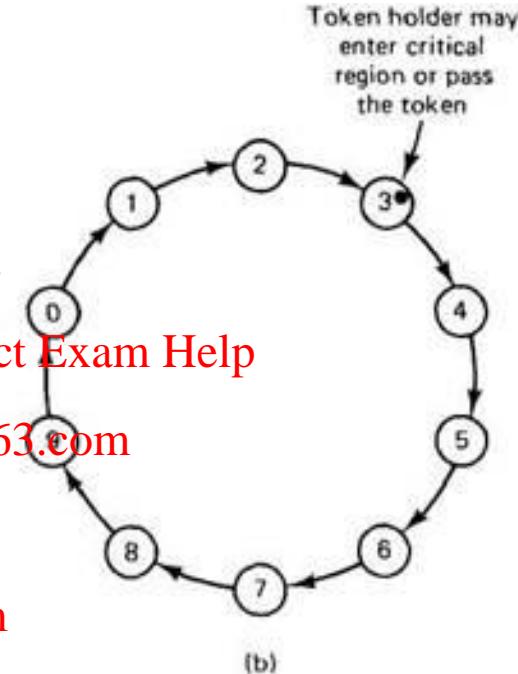
Assignment Project Exam Help

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

(a)



(b)

Distributed Operating Systems. Andrew Tanenbaum

Comparison of Resource Allocation Algorithm

程序代写代做 CS 编程辅导

Algorithm	Message Resource	Delay before Resource Allocation (in message times)	Potential Problems
Centralised	3 WeChat: cstutorcs Assignment Project Exam Help	2	Coordinator Crash (Central Point of Failure)
Distributed	$2(n-1)$ Email: tutorcs@163.com	$2(n-1)$	Crash of any process
Token Ring	$1 - \infty$ QQ: 749389476 https://tutorcs.com	$0 - (n-1)$	Lost token, Process Crash

Election Algorithms

程序代写代做CS编程辅导

- In the event of failure of a leader process in a centralized scheme, for other algorithms such as the Berkley Algorithm discussed earlier an election algorithm is necessary to select a new leader
- Election algorithms include:
 - Bully Algorithm
 - Ring Algorithm

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>



Bully Algorithm

程序代写代做 CS编程辅导

- In the Bully algorithm each process is assigned a unique weight value
- If a process notices that the coordinator has stopped it sends election messages to processes which have a higher weight
- When a process receives an election message it replies to the election message. In addition if it has not sent messages to processes with a higher weight value it does so
- If the process receives an election message and it has sent messages it stops
- If after a certain time period a process has received no replies to its election message it determines that it has won the election and sends out a message to all processes indicating that it is the coordinator
- When a process recovers it will launch a new election



We Chat messages

Assignment Project Exam Help

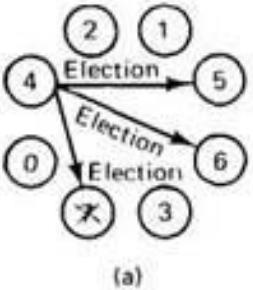
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Bully Algorithm

程序代写代做 CS编程辅导

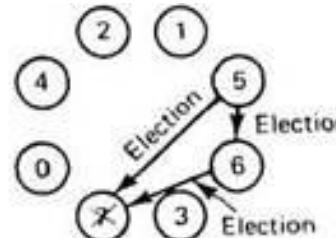


(a)

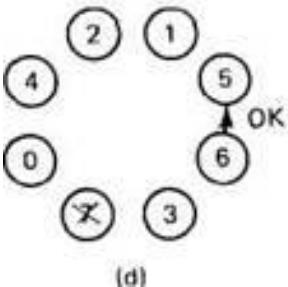


WeChat: cstutorcs
Previous coordinator
has crashed

Assignment Project Exam Help



(c)



(d)

Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

(e)



Distributed Operating Systems. Andrew Tanenbaum

Ring Algorithm

程序代写代做 CS 编程辅导

- In a ring algorithm when any process receives that the coordinator is down it sends an election message to the next process in the logical ring
- The election message includes its process number
- If the next process in the logical ring does not acknowledge the message then it sends it to the process after this in the logical ring until it receives an acknowledgement
- When a process receives an election message it adds its process number to the election message and forwards it to the next process in the ring
- When a process receives an election message it then sends a coordinator message around the ring which indicates who is the new coordinator (the highest process number) and the members of the logical ring
- When a process receives the coordinator message it has sent it removes the message as it has gone around the ring. This is done to reduce overhead



WeChat: cstutorcs

Assignment Project Exam Help

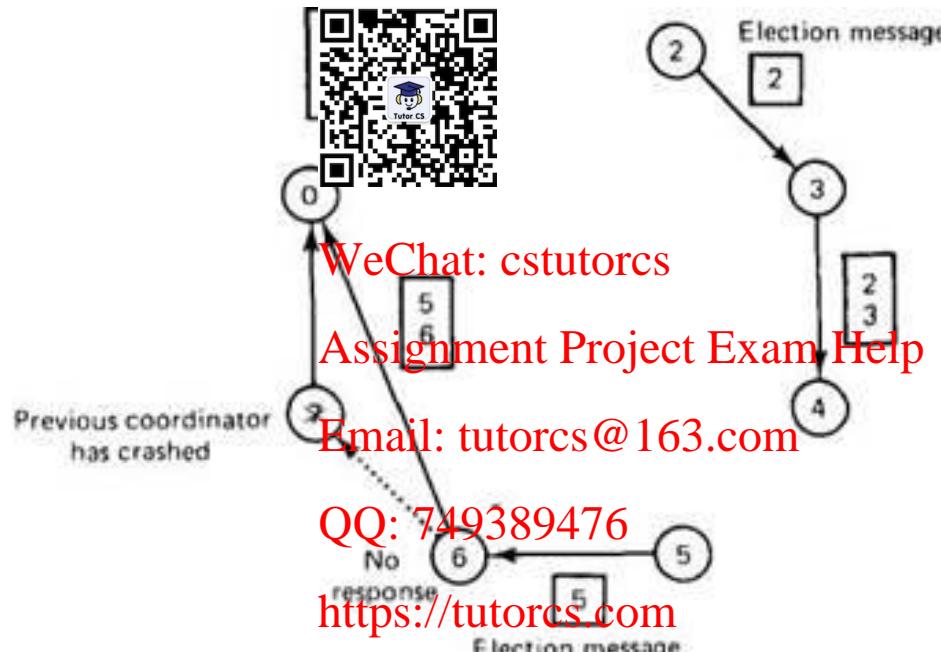
Email: tutorcs@163.com

QQ: 749389476

<https://tutorcs.com>

Ring Algorithm

程序代写代做 CS编程辅导



Distributed Operating Systems. Andrew Tanenbaum