Define bandwidth.

1

Define throughput.

2

On a graph diagram of a distributed system, what is represented by the nodes on the graph?

3

Name some properties of edges that connect the nodes in a graph of a distributed system.

4

Why is it that even though connections between nodes can be implemented in different ways (such as wifi or ethenet), they can be treated as the same?

5

Name the eight axioms of distributed systems.

6

Why is transparency desirable in a distributed system?

7

What is transparency of location? How can we achieve it?

8

Bandwidth measures the maximum amount of data that can be communicated within a certain amount of time.

1

Throughput measures the actual rate at which messages are communicated.

2

The physical nodes of the network (individual systems). Each can host multiple processes and resources.

3

The type of connection (wired/wifi/mobile data etc).
The bandwidth.
The latency.

4

The implementation details of each connection is abstracted away by many layers of protocols.

5

- *Latency is greater than zero.*
- *Bandwidth is less than infinate.*
- *Transport cost is greater than zero.*
- *There is more than administrator.*
- *The network topology can and will change.*
- *The network is not homogenous (the nodes and edges differ).*
- *The network is not secure.*
- *The network is not reliable.*

6

It allows us to design systems as though the distributed axioms were false.

7

An attempt to hide the need to know of where a specific resource is physically located.
Use DNS servers to map host names to IP addresses.

8

What is transparency of migration? How can we achieve it?

9

What is transparency of relocation? How is it achieved?

10

What is transparency of replication. How is it achieved?

11

What is transparency of access? How is it achieved?

12

What is transparency of concurrency?

13

What is deadlock?

14

What is livelock?

15

What is transparency of failure? How is it achieved?

16

When a host moves location in the network, we shouldn't need to the details of the move.
The DNS architecture implements this, though if a resource keeps moving, then the route through the network and therefore the latency of the connection to the host is hard to predict.

9

Transparency of relocation is when parts of the system move while they are being accessed. This is hard to mitigate, and is often a problem with mobile phone communications.

10

When there is more than one physical resource that does the same job, which one do we use? It is hard to achieve.

11

Transparency of access is the ability to not care about how a node is implemented. This is often achieved using protocols and API's and middleware.

12

Different users shouldn't need to know that others are using the same resource and may be competing for its time. Atomic operations and enforcing consistency are ways to achieve this, but this can force users to wait on each other (deadlock, livelock etc).

13

When two different processes are unable to progress since each is waiting for information from the other.

14

When two processes change with respect to one another so that neither can make progress.

15

Users should not know that a specific node has failed or has recently had downtime. Hard to ensure, since sometimes slow connections are indistinguishable from failed nodes.

16

Define systems software.

17

Describe the start of the mainframe era.

18

What advances were made during the mainframe era?

19

What is the danger of multiprogramming?

20

What three things can help multiprogramming be effective?

21

How does the scheduler make decisions about programs?

22

In the beginning of the PC era, what was a typical PC like?

23

What happens to the price of PC's during the PC era?

24

- *Hardware was vastly more expensive than people.*
- *Only single user programs.*
- *Programs had to be written every time they were to be run.*

18

The underlying level of software which allows applications to perform their tasks efficiently. Examples include device drivers and the operating system.

17

One program could potentially have access to data in other programs, which could present security/compatability concerns. Memory protection is used so that programs can only read and write stuff they own.

20

- *The ability to store programs was developed.*
- *Batch processing, where jobs would be written to a queue to be executed. Lost the ability to debug.*
- *Use interrupt handlers and buffers to allow the CPU and IO to work in parallel.*
- *Multiprogramming so that one proram has the CPU and another has the IO to enable full resource utalization.*

19

Give them priority levels.

22

- *Fairness policies that impose limits on how much resources jobs should use*
- *Shedulers that aim to minimise the time it takes to complete jobs, by reducing their response and turnaround time.*
- *Preemtive sheduling will temporarily stop jobs if they are hogging resources.*

21

Price falls due to comoditization and competition.

24

Everything was cut back with just a single user in mind; no timesharing, multiprogramming or protection in the OS, few resources such as memory and CPU power etc.

23

How does the usage of PC's change during the PC era?

25

What does WIMP stand for and when was it introduced?

26

What drove an increase in security focus during the PC era?

27

What are the four standard layers of the network protocol stack?

28

What is the function of the `HEAD` *HTTP verb?*

29

What is the function of the `GET` *HTTP verb?*

30

What is the function of the `POST` *HTTP verb?*

31

What is the function of the `OPTIONS` *HTTP verb?*

32

Windows, Icons, Menus, Pointer

The WIMP system became common during the PC era.

26

They become used for more simple tasks such as word processing, rather than number crunching and enterprise processing.

25

Name	**Example**
Application Layer	`HTTP`, `POP`, `SSL`
Transport Layer	`TCP`, `UDP`
Internet Layer	`IP`, `ICMP`, `IGMP`
Link Layer	`ARP`, `Ethernet`, `DSL`, `IDSN`

28

The fact that sharing data over networks was becoming increasingly common, which exposed security risks. Authentication and authorization methods were therefore developed.

27

Gets a resource from the server.

30

Provides metadata about a resource. Just like a `GET` *request, except the body of the request is not returned. Used to let clients know if cached data has changed.*

29

Returns the list of commands supported by the server.

32

Submits data to the server. The state of the server may change as a result of a `POST` *request.*

31

What does the 'store and forward' model mean?

33

What is the SMTP protocol?

34

Describe 'reduction to absurdity'.

35

With a Lamport clock, how is the clock of A affected if B sends A a message?

36

Describe the purpose of a mutex.

37

What is a critical section?

38

Describe the two-phase commit.

39

Describe the bully algorithm.

40

The Simple Mail Transfer Protocol is a method for sending email. It is a stateful, connection based protocol, so a connection to the server must be opened before the message can be transferred, and then closed again after.

34

When a message needs to be transmitted along a network, but not all the nodes along the path may be active at the same time. Concequently, the message is stored by each node until it has been confirmed to have been recieved by the next node.

33

```
if LCy < (LCx + 1):
    LCy = LCx + 1
```

36

This is a type of argument where an idea is shown to be false by showing that an absurd result can be obtained by the acceptance of the idea.

35

A critical section is a sequence of actions that should execute in an atomic matter. That is to say that if the critical section does not succeed, then the state of the system should be exactly the same as when the critical section started.

38

A mutex ensures that two processes do not enter into a critical section of their execution at the same time and therefore prevents race conditions and enables the safe sharing of resources.

37

1 *A node wants to be a coordinator, so it sends a message to all the other nodes with it's number.*
2 *All the other nodes will only reply if they have a higher number. If they do, they will start another election.*
3 *The process continues until a node recieves no replies, and is therefore the coordinator, at which point, it will inform all the other nodes of this.*

40

1 *A coordinator node wishes all of the nodes to perform an action.*
2 *It sends a request to all of the other nodes.*
3 *The other nodes attempt to perform the action, and reply with* `commit` *or* `abort` *depending on whether they to performed the action.*
4a *If any one of the nodes replied* `abort`*, then the coordinator node will reply* `global_abort` *and all the nodes will roll back their state to that of the start of the transaction.*
4b *If all the nodes replied* `commit`*, then the coordinator node will send* `global_commit` *and the nodes will retain their current state.*

39

Describe Cristian's algorithm?	*Describe the Berkly algorithm.*
41	42

1 *A master node is chosen.*
2 *The master polls the other nodes for the time.*
3 *The master observes the round trip time for each state, and estimates the time of each client.*
4 *The master then averages out the clock times and sends each node a message containing the difference that it is from the average.*

In order to syncronize the clocks of two systems, one system S will send it's time to another system C. After recieving the request from S, C will then reply with it's own clock time. When S recieves this, then it will set its clock to be the time that C sent added to the round trip time (discernable from C's reply) divided by two.