ENGINEER PRO

[1] EngineerPro- Fanpage:https://www.facebook.com/EngineerPro.Official

Tổng hợp các câu hỏi về CS fundamental

1. **What is the index in Database (in MySQL, Microsoft SQL server…)?**
   Answer:
   Indexes are used to find rows with specific column values quickly. Without an index, Database must begin with the first row and then read through the entire table to find the relevant rows. The larger the table, the more this costs. If the table has an index for the columns in question, Database can quickly determine the position to seek to in the middle of the data file without having to look at all the data. This is much faster than reading every row sequentially.

2. **What are the Hash Index and B Tree Index ?**
   Answer: (To know more, let access this link)
   - Hash index:
     + It is a kind of hashing technique based on a hash table which supports storing and retrieving data.
     + They are used only for equality comparisons that use the = or != operators (but are *very* fast). They are not used for comparison operators such as < that find a range of values. Systems that rely on this type of single-value lookup are known as “key-value stores”; to use MySQL for such applications, use hash indexes wherever possible.
   - B-Tree Index:

    - B-tree stands for “balanced tree”, it’s a self-balancing tree data structure that maintains sorted data and allows searches.
    - A B-tree index can be used for column comparisons in expressions that use the =, >, >=, <, <=, or BETWEEN operators. The index also can be used for LIKE comparisons if the argument to LIKE is a constant string that does not start with a wildcard character

3. **What is sharding in the database?**
   Answer: https://en.wikipedia.org/wiki/Shard_(database_architecture)

4. **What is partition in the database?**
   Answer: https://en.wikipedia.org/wiki/Partition_(database)

5. **Compare NOSQL vs Relational database?**
   Answer: https://www.mongodb.com/scale/nosql-vs-relational-databases

6. **Tell me about the HTTP methods?**

   HTTP methods tell the server what to do. There are a lot of HTTP methods but we’ll study the most common ones: GET, POST, HEAD, PUT, or DELETE.

- **GET** is the most common and **requests data**.
- **POST puts an object on the server**.
    - This method is generally used when the client is not sure where the new data would reside. The server responds with the location of the object.
    - The data posted can be a message for a bulletin board, newsgroup, mailing list, a command, a web form, or an item to add to a database.
    - The POST method technically requests a page but that depends on what was entered.

- **HEAD** is similar to the GET method except that **the resource requested does not get sent in response. Only the HTTP headers are sent** instead.
    - This is useful for quickly retrieving meta-information written in response headers, without having to transport the entire content. In other words, it's useful to check with minimal traffic if a certain object still exists. This includes its meta-data, like the last modified date. The latter can be useful for caching.
    - This is also useful for testing and debugging.
- **PUT uploads an enclosed entity under a supplied URI**. In other words, it **puts** data at a specific location. If the URI refers to an already existing resource, it's replaced with the new one. If the URI does not point to an existing resource, then the server creates the resource with that URI.
- **DELETE deletes an object** at a given URL.

7. **What are UDP and TCP?**

   - UDP, or **User Datagram Protocol**, is a transport layer protocol that works over the network layer's famous **Internet protocol**.

   - Does not ensure in-order delivery of messages that we call 'datagrams.'

   - Detects any modifications that may have been introduced in the packets during delivery but does not correct them by default.

   - Does not ensure reliable delivery.

   - Generally faster than TCP because of the reduced overhead of ensuring uncorrupted delivery of packets in order.

   - Applications that use UDP include: Domain Name System (DNS), live video streaming, and Voice over IP (VoIP).

- TCP (Transmission Control Protocol) is a kind of transport layer protocol.
    - Delivers messages that we call ‘segments’ reliably and in order.
    - Detects any modifications that may have been introduced in the packets during delivery and corrects them.
    - Handles the volumes of traffic at one time within the network core by sending only an appropriate amount of data at one time.
    - Examples of applications/application protocols that use TCP are: HTTP, E-mail, File Transfers.

8. **How does UDP work?**

   UDP does not involve any initial handshaking like TCP does, and is hence called a connectionless protocol. This means that there are no established ‘connections’ between hosts.

   UDP prepends the source and destination ports to messages from the application layer and hands them off to the network layer. The Internet Protocol of the network layer is a best-effort attempt to deliver the message. This means that the message-

   1. May or may not get delivered.
   2. May get delivered with changes in it.
   3. May get delivered out of order.

   UDP only adds the absolute bare minimum functionality over the network layer. So it…

   - Does not ensure that messages get sent.
   - It does check, however, if a message got ‘corrupted’ yet does not take any measures to correct the errors by default.

9. **How does TCP work?**

You can answer:

* Describe TCP handshake process
* Describe the work process of a TCP proxy
* Open & close connection
* How TCP could make reliable data transfer
* Congestion control

You can refer to this link for more detail!

10. What is the transaction?

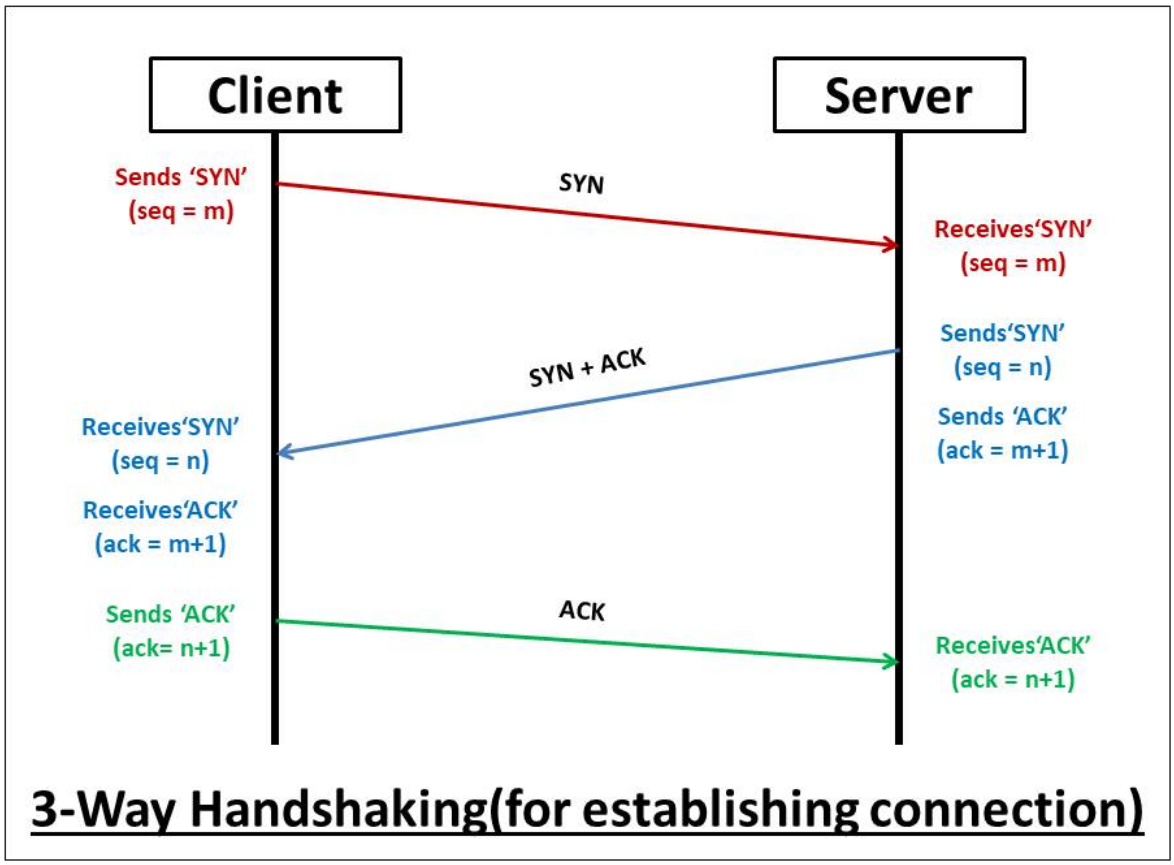
A transaction is a logical unit of work that contains one or more SQL statements. A transaction is an atomic unit. The effects of all the SQL statements in a transaction can be either all committed (applied to the database) or all rolled back (undone from the database).

11. What is eventual consistency? (Em có thể đọc thêm link này)

Eventual consistency is a consistency model used in distributed computing to achieve high availability that informally guarantees that, if no new updates are made to a given data item, eventually all accesses to that item will return the last updated value.[1] Eventual consistency, also called optimistic replication,[2] is widely deployed in distributed systems, and has origins in early mobile computing projects.[3] A system that has achieved eventual consistency is often said to have converged, or achieved replica convergence.[4] Eventual consistency is a weak guarantee – most stronger models, like linearizability, are trivially eventually consistent, but a system that is merely eventually consistent does not usually fulfill these stronger constraints.

12. Let you describe TCP handshake

TCP uses a three-way handshake to establish a reliable connection. The connection is full duplex, and both sides synchronize (SYN) and acknowledge (ACK) each other. The exchange of these four flags is performed in three steps—SYN, SYN-ACK, and ACK—as shown in Figure.



13. What is the TCP proxy?

A TCP proxy is a server that acts as an intermediary between a client and the destination server. Clients establish connections to the TCP proxy server, which then establishes a connection to the destination server. TCP proxy supports Window Scale (WS) options that are carried by SYN and SYN ACK packets. If both the client and server support the WS option, the client and server use Window Scale Factor (WSF) to scale the receive and send window. TCP proxy supports a maximum receive window size of 1 MB per session. If the scaled TCP receive window size from the receiver is smaller or equal to 1 MB, the TCP proxy will use it directly. But if the scaled TCP receive window size is larger than 1 MB, the TCP proxy will use 1 MB as the proxy receive window size.

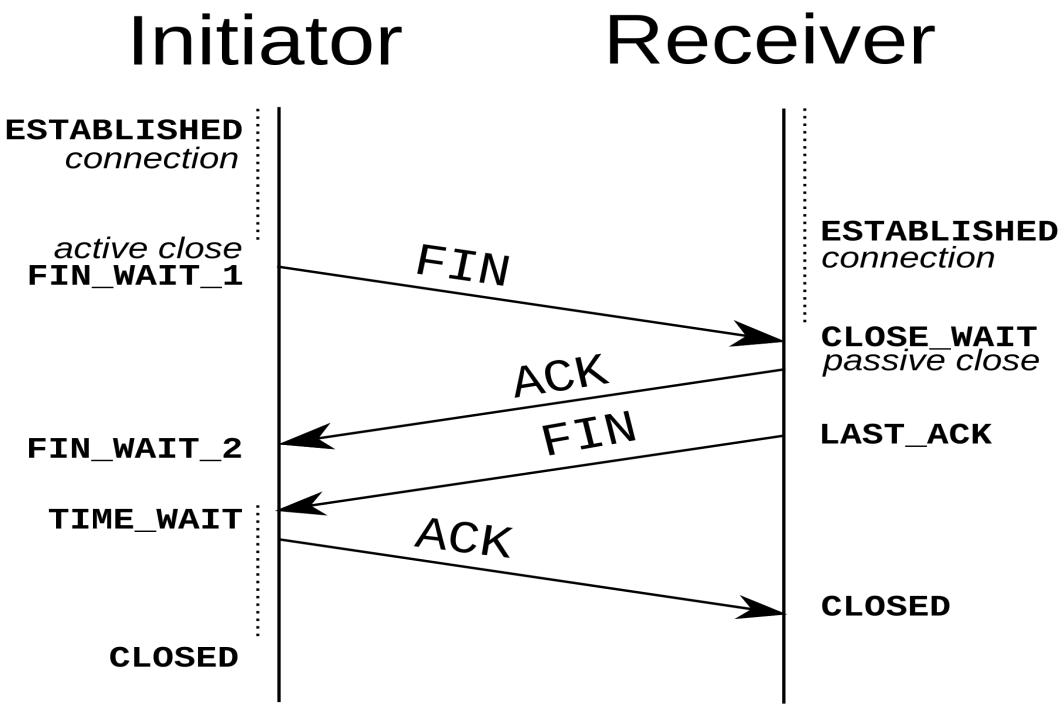
14. How does TCP open and close a connection?

Reference to this link

1. Open a connection: Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a passive open. Once the passive open is established, a client may establish a connection by initiating an active open using the three-way (or 3-step) handshake.

2. Close a connection:

   The connection termination phase uses a four-way handshake, with each side of the connection terminating independently. When an endpoint wishes to stop its half of the connection, it transmits a FIN packet, which the other end acknowledges with an ACK. Therefore, a typical tear-down requires a pair of FIN and ACK segments from each TCP endpoint. After the side that sent the first FIN has responded with the final ACK, it waits for a timeout before finally closing the connection, during which time the local port is unavailable for new connections; this prevents possible confusion that can occur if delayed packets associated with a previous connection are delivered during a subsequent connection



15. What is the congestion control in TCP?

Transmission Control Protocol (TCP) uses a network congestion-avoidance algorithm that includes various aspects of an additive increase/multiplicative decrease scheme, along with other schemes including slow start and congestion window, to achieve congestion avoidance. **(Slow start and congestion window are part of congestion control strategy)**

16. What is a CA (Certificate authority)?

In cryptography, a certificate authority or certification authority (CA) is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows others (relying parties) to rely upon signatures or on assertions made about the private key that corresponds to the certified public key. A CA acts as a trusted third party—trusted both by the subject (owner) of the certificate and by the party relying upon the certificate.

17. What is a DNS server?

The Domain Name System (DNS) is the phonebook of the Internet. When users type domain names such as ‘google.com’ or ‘nytimes.com’ into web browsers, DNS is responsible for finding the correct IP address for those sites. Browsers then use those addresses to communicate with origin servers or CDN edge servers to access website information. This all happens thanks to DNS servers: machines dedicated to answering DNS queries.

18. For Pythonista. How does python dictionary handle collisions?

https://stackoverflow.com/questions/21595048/how-python-dict-stores-key-value-when-collision-occurs

19. Compare thread vs process

https://www.geeksforgeeks.org/difference-between-process-and-thread/

20. Tell me about states of process?

Answer: https://en.wikipedia.org/wiki/Process_state

21. What is virtual memory?

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

22. What is Swapping Space?

Answer: https://www.geeksforgeeks.org/swap-space-in-operating-system/

23. What is the paging technique in virtual memory?

(more info: https://en.wikipedia.org/wiki/Memory_paging)

It is sometimes said that the operating system takes one of two approaches when solving most any space-management problem. The first approach is to chop things up into *variable-sized* pieces, as we saw with segmentation in virtual memory. Unfortunately, this solution has inherent difficulties. In particular, when dividing a space into different-size chunks, the space itself can become fragmented, and thus allocation becomes more challenging over time.

Thus, it may be worth considering the second approach: to chop up space into *fixed-sized* pieces. In virtual memory, we call this idea paging, and it goes back to an early and important system, the Atlas. Instead of splitting up a process's address space into some number of variable-sized logical segments (e.g., code,

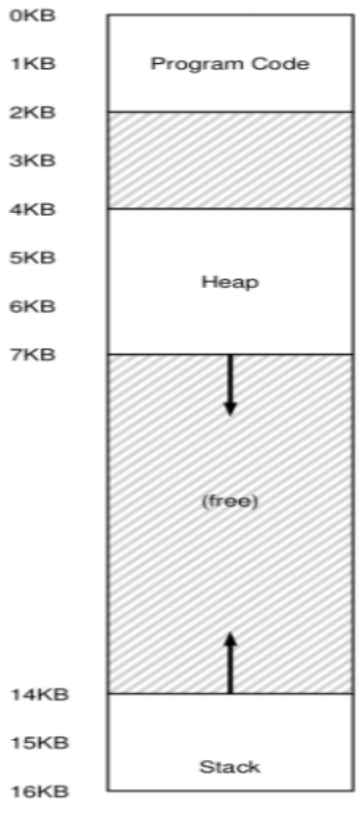
heap, stack), we divide it into fixed-sized units, each of which we call a page. Correspondingly, we view physical memory as an array of fixed-sized slots called page frames; each of these frames can contain a single virtual-memory page

24. What is the segmentation technique in virtual memory?

(more info: https://en.wikipedia.org/wiki/Memory_segmentation)

So far we have been putting the entire address space of each process in memory. With the base and bounds registers, the OS can easily relocate processes to different parts of physical memory. However, you might have noticed something interesting about these address spaces of ours: there is a big chunk of "free" space right in the middle, between the stack and the heap.

As you can imagine from the figure below, although the space between the stack and heap is not being used by the process, it is still taking up physical memory when we relocate the entire address space somewhere in physical memory; thus, the simple approach of using a base and bounds register pair to virtualize memory is wasteful.



An Address Space (Again)

It also makes it quite hard to run a program when the entire address space doesn't fit into memory; thus, base and bounds are not as flexible as we would like.

25. What is Inter-Process Communication (IPC)?

Answer: https://en.wikipedia.org/wiki/Inter-process_communication

26. What is the loopback network interface(localhost)?

Answer: https://en.wikipedia.org/wiki/Localhost