1. 多刷题!!! 经典题目 https://github.com/yuanguangxin/LeetCode

2. Java 垃圾回收机制: https://www.jianshu.com/p/5261a62e4d29

3. Payment process: https://www.paypal.com/us/brc/article/how-online-payments-processing-works#:~:text=The%20payment%20processor%20sends%20a,to%20pay%20for%20your%20stuff.&text=The%20payment%20processor%20sends%20the,bank%20to%20credit%20your%20account.
4. DDOS protection: https://zhuanlan.zhihu.com/p/22953451
   - SYN Cookie 的作用是缓解服务器资源压力。启用之前，服务器在接到 SYN 数据包后，立即分配存储空间，并随机化一个数字作为 SYN 号发送 SYN+ACK 数据包。然后保存连接的状态信息等待客户端确认。启用 SYN Cookie 之后，服务器不再分配存储空间，而且通过基于时间种子的随机数算法设置一个 SYN 号，替代完全随机的 SYN 号。发送完 SYN+ACK 确认报文之后，清空资源不保存任何状态信息。直到服务器接到客户端的最终 ACK 包，通过 Cookie 检验算法鉴定是否与发出去的 SYN+ACK 报文序列号匹配，匹配则通过完成握手，失败则丢弃。
   - 除了定制 TCP/IP 协议栈之外，还有一种常见做法是 TCP 首包丢弃方案，利用 TCP 协议的重传机制识别正常用户和攻击报文。当防御设备接到一个 IP 地址的 SYN 报文后，简单比对该 IP 是否存在于白名单中，存在则转发到后端。如不存在于白名单中，检查是否是该 IP 在一定时间段内的首次 SYN 报文，不是则检查是否重传报文，是重传则转发并加入白名单，不是则丢弃并加入黑名单。是首次 SYN 报文则丢弃并等待一段时间以试图接受该 IP 的 SYN 重传报文，等待超时则判定为攻击报文加入黑名单。
   - 第一个是统计每个 TCP 连接的时长并计算单位时间内通过的报文数量即可做精确识别。一个 TCP 连接中，HTTP 报文太少和报文太多都是不正常的，过少可能是慢速连接攻击，过多可能是使用 HTTP 1.1 协议进行的 HTTP Flood 攻击，在一个 TCP 连接中发送多个 HTTP 请求。
   - 第二个是限制 HTTP 头部传输的最大许可时间。超过指定时间 HTTP Header 还没有传输完成，直接判定源 IP 地址为慢速连接攻击，中断连接并加入黑名单。
5. CAP (consistency, availability, partition tolerance) theorem in distributed system: https://www.cnblogs.com/duanxz/p/5229352.html

6. HTTP: https://juejin.cn/post/6844904045572800525
7. http 请求特征:
   - 支持客户-服务器模式
   - 简单快速：客户向服务器请求服务时，只需传送请求方法和路径。请求方法常用的有 GET、HEAD、POST。每种方法规定了客户与服务器联系的类型不同。由于 HTTP 协议简单，使得 HTTP 服务器的程序规模小，因而通信速度很快。
   - 灵活：HTTP 允许传输任意类型的数据对象。正在传输的类型由 Content-Type 加以标记。
   - 无连接：无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间。
   - 无状态：HTTP 协议是无状态协议。无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。另一方面，在服务器不需要先前信息时它的应答就较快(use cookie to memorize the state)

8. What is the tcp three-way handshake? SYN ACK SYN
9. Implement hashmap, hashtable
   HashMap is **non synchronized**. It is **not-thread s**afe and can't be shared between many threads without proper synchronization code whereas Hashtable is synchronized. It is thread-safe and can be shared with many threads. HashMap allows one null key

HashMap 计算 hash 对 key 的 hashcode 进行了二次 hash，以获得更好的散列值，然后对 table 数组长度取模
添加、删除、获取元素时都是先计算 hash，根据 hash 和 table.length 计算 index 也就是 table 数组的下标，然后进行相应操作 https://zhuanlan.zhihu.com/p/65267707

Implement: dynamic arrays and linkedlist

10. The difference between a reference and a pointer:
    - References are used to refer an existing variable in another name whereas pointers are used to store address of variable.
    - References cannot have a null value assigned but pointer can.
    - A reference variable can be referenced pass by value whereas a pointer can be referenced by pass by reference.
    - A reference must be initialized on declaration while it is not necessary in case of pointer.
    - A reference shares the same memory address with the original variable but also takes up some space on the stack whereas a pointer has its own memory address and size on the stack.

11. How to design a dynamic array in java. **Auto-resize array**
12. Different cashing strategy of a mobile app
    Cache Aside： If the data exists (we call this a 'cache hit'), the app will retrieve the data directly. If not (we call this a 'cache miss'), the app will request data from the database and write it to the cache so that the data can be retrieved from the cache again next time.
    Read Through: Unlike cache aside, the cache sits in between the application and the database. The application only request data from the cache. If a 'cache miss' occurs, the cache is responsible to retrieve data from the database, update itself and return data to the application.
    https://bluzelle.com/blog/things-you-should-know-about-database-caching#:~:text=A%20caching%20strategy%20is%20to,design%20and%20the%20resulted%20performance.

13. How react's virtual DOM help improve performance.
14. Merging k sorted list **merge one by one. Array**
15. Find k-th largest element in unsorted array. **Random + quick select.**
16. Reverse a linked list **head= head.next.next, array**
17. Difference between tcp and udp
18. How DNS works (UDP):
    1. DNS recursor:
    2. Root nameserver
    3. TLD nameserver
    4. Authoritative nameserver

19. Complexity of hashtable， O(1), resize O(n)
20. How https works
    https://segmentfault.com/a/1190000021494676

21. Difference between https and http
22. what are Transactions in RDBS
    https://www.mysqltutorial.org/mysql-transaction.aspx/
    ACID:

Atomicity

All changes to data are performed as if they are a single operation. That is, all the changes are performed, or none of them are.

Consistency

Data is in a consistent state when a transaction starts and when it ends.
MySQL 为了保证 ACID 中的一致性和持久性，使用了 WAL(Write-Ahead Logging,先写日志再写磁盘)。Redo log 就是一种 WAL 的应用。当数据库忽然掉电，再重新启动时，MySQL 可以通过 Redo log 还原数据。也就是说，每次事务提交时，不用同步刷新磁盘数据文件，只需要同步刷新 Redo log 就足够了。

Isolation

The intermediate state of a transaction is invisible to other transactions. As a result, transactions that run concurrently appear to be serialized. 同一时间，只允许一个事务请求同一数据

Durability

After a transaction successfully completes, changes to data persist and are not undone, even in the event of a system failure. 事务完成后，事务对数据库的所有更新将会保存到数据库不能 rollback,

https://www.ibm.com/support/knowledgecenter/SSGMCP_5.4.0/product-overview/acid.html

23. Isolation level in RDBS:
    - **Read Uncommitted:**
      A transaction can see changes to data made by other transactions that are not committed yet, awalys cause **dirty read（rollback）**
    - **Read Committed:** Avoid dirty read, A **non-repeatable read(update)** occurs when a transaction performs the same transaction twice but gets a different result set each time.
    - **Repeatable Read**(default in MySQL): all the select use the same result when it first time to run. cause **Phantom read,** transaction A 读取了 B 已经提交的新增数据.
    - **Serializable**: row selection of one transaction cannot be changed by another transaction until it finishes.
24. Difference between threads and process
    https://zhuanlan.zhihu.com/p/46368084

25. Drop-out layer
26. Compare layer 4 and layer 7

WEB

- **HTTP**: Hypertext Transfer Protocol is an application protocol that defines a language for clients and servers to speak to each other. This is like the language you use to order your goods.

Cookies: https://www.kaspersky.com/resource-center/definitions/cookies

Cookies are text files with small pieces of data — like a username and password — that are used to identify your computer as you use a computer network. Specific cookies known as HTTP cookies are used to identify specific users and improve your web browsing experience.

Data stored in a cookie is created by the server upon your connection. This data is labeled with an ID unique to you and your computer.

When the cookie is exchanged between your computer and the network server, the server reads the ID and knows what information to specifically serve to you.

Cookies let websites remember you, your website logins, shopping carts and more

How to reduce the page loading time:

- Reduce the image size
- Use the latest generation formats for images
- Minify HTML, CSS, and Javascript
- Postpone uploading off-screen images
- Create Accelerated Mobile Pages (AMPs)
- Remove unnecessary widgets
- Avoid multiple redirects
- Place CSS at the top and script referencing at the bottom or external files
- Reduce lookups
- Minimize redirects and caching
- Check the current speed of the website
- Finding a good hosting to host your website
- Clean the web code
- Cache

HTTPS 原理：加密 SSL,TLS(RSA,DSA) （AES 是对称加密）

1. http（完全明文传输，不安全）传输，经过，电脑操作系统，wifi router, proxy(singtel), server
2. 非对称加密，通过第三方 CA 把服务器的公钥传给浏览器（证书）
3. 靠 tcp 建立网络连接
4. Application layer
5. **客户端在拿到服务器的公钥后，会生成一个随机码（KEY），然后客户端使用公钥把 key 加密发送给服务器，服务器用私钥解开 key 后，这样双方就有了同一个密钥，然后双方再通过对称加密交互数据。**

http request 响应程序：

1. 浏览器会从缓存中查找 ip 地址，（浏览器缓存，本地缓存，DNS->路由器缓存）
2. 获取 ip，之后，浏览器会向 ip 地址端建立 tcp 连接，
   - 浏览器->syn 请求连接->server
   - Server->syn ack 确认链接->浏览器
   - 浏览器->确认收到 ack->server
3. 开始通信(GET POST)
4. 发送 package with PIN flag turned on 关闭连接

Transport Layer:

IP does not guarantee:

- Segment delivery
- Orderly delivery of segments
- Integrity of segments

Multiplexing: take data, covert to segments, send from process to network
Demultiplexing: collect segments from network and send to process
For TCP, all 4 addresses information (source IP, destination IP, source port and destination port) are used during demultiplexing the segment to a socket.

ARQ (Automatic Repeat reQuest) protocols:

- Error detection: Something like what UDP does.
- Receive Feedback: Receiver will send positive (ACK) and negative (NAK) acknowledgment as feedback. In principle, these packets need only be one bit long; for example, a 0 value could indicate a NAK and a value of 1 could indicate an ACK.
- Retransmission. A packet that is received in error at the receiver will be retransmitted by the sender.

If the ack/nck is corrupted, simply resend package, it may cause duplicate packet. The solution is to add a sequence number.
Pipeline (to increase the bandwidth utilization):

- Go back N:
- Selective repeat:

TCP

1. Sequence number is based on byte system, not segment
   Flow control: It is the sender's responsibility to make sure not to overflow receiver's buffer unnecessarily.
2. Three ways handshake:

- **Step 1 (SYN) :** In the first step, client wants to establish a connection with server, so it sends a **segment with SYN(Synchronize Sequence Number)** which informs server that client is likely to start communication and with what sequence number it starts segments with

- **Step 2 (SYN + ACK):** Server responds to the client request with SYN-ACK signal bits set. Acknowledgement(ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with
- **Step 3 (ACK) :** In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start the actual data transfer，如果第三次握手 ack 丢失了，但他的下一个数据包没有丢失，那么连接正常，否则，连接重置. 第三次握手可以携带数据

DDOS ： SYN attack: hackers send many connection request with many ips to server, server will allocate many resources to them, and in SYN_RCVD state for a long time.

最大连接数	此虚拟服务器允许的最大并行连接数。
RPS 最大值	每秒钟允许的来自客户机的最大请求数。
RPS 计算间隔	计算每秒请求数 (request per second, RPS) 平均值的时间间隔。默认值为 30 秒。

3. Congestion control: Congestion is detected using timeout and three duplicate acknowledgment. Sending rate: congestion window size
   additive increase/multiplicative decrease (AIMD) scheme, along with other schemes including **slow start** (double every RTT) (after timeout) and **congestion window.** After congestion windows exceed the slow-start threshold, ->AIMD

On timeout:

1. Congestion window is reset to 1 MSS.
2. *ssthresh* is set to half the congestion window size before the timeout.
3. *slow start* is initiated.

Tahoe: Three duplicate ACKs received, retransmit, reset ssthreshold to half of the current congestion window, reduce the congestion window to 1.
Reno: Three duplicated Acks received, retransmit, and skip the slow start phase by instead half the congestion window, setting ssthreshold to the new congestion window.

The standard size of a TCP packet has a minimum size of 20 bytes, and a maximum of 60 bytes. The UDP packet consists of only 8 bytes for each packet.

TCP Segment Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Sequence Number			
64	Acknowledgment Number			
96	Data Offset	Res	Flags	Window Size
128	Header and Data Checksum		Urgent Pointer	
160...	Options			

UDP Datagram Header Format

Bit #	0 – 7	8 – 15	16 – 23	24 – 31
0	Source Port		Destination Port	
32	Length		Header and Data Checksum	

Ethernet with a MTU of 1500 would result in a MSS(maximum segments size) of 1460 after subtracting 20 bytes for IPv4 header and 20 bytes for TCP header.

General Network:

IP address: four parts: 192.198.17.5
Subnet mask: 255.255.255.0(255 means it belongs IP network, 0 means it belongs to host network)
For IP address (0 is for large scale network, 255 is used for broadcasting.)
Default gateway: router's IP address(start with). 192.168.1.1(private IP address)
Domain Name System(**DNS**)(UDP): The home router will then either answer directly if it already knows the answer, or it will forward the query to a DNS server on the Internet.
How it works:

1. The computer OS checks its DNS cache to see if it already knows what IP address
2. The computer will construct a DNS query that it can send off to the DNS server
3. Check the ARP table for a valid MAC address
4. Computer must construct an ARP request to the rest of the network. The request will be sent to destination MAC address FF:FF:FF:FF which is the broadcast address
5. Received MAC address from router, send DNS query
6. Home router checks it DNS cache
7. Router prepares a DNS query that it will send to its DNS server. The router learned about available DNS servers via DHCP from the Internet Service Provider when the home router first booted up and got its own public IP address from the ISP.
8. Each router on the Internet that receives the DNS request will perform the following:

- Receive the packet
- Looks at the *destination IP address* to see where the packet is going
- Looks in its *routing table* to see which path that is best for the packet
- Removes the old *MAC addresses* from the packet and adds new ones. It will use its own MAC address on the outbound interface as the *Source MAC address* for the traffic, and will put the next-hop router's MAC address as the *destination MAC address*
- Sends off the packet to the next hop router

9. receive ip address from DNS server

10. Three-ways handshake:

4 ways handshake to terminate:

1. Client set FIN =1 and send to server, don't send, still receive. FIN_WAIT-1
2. Server send ACK to confirm received. CLOSE_WAIT
3. After server send all data to client, it set FIN =1, send to client, FIN-WAIT-2
4. Client send ACK to server, close connection. TIME-WAIT
   The server will close connection before client.

Dynamic Host Configuration Protocol**(DHCP):** routers automatically hand out the IP address, subnet mask, default gateway and IP address of DNS server to its network device.

Private IP address space	
From	**To**
10.0.0.0	10.255.255.255
172.16.0.0	172.31.255.255
192.168.0.0	192.168.255.255

NAT (Network Address Translation): when LAN(computers) want to communicate to WAN, the router will translate its private IP to public IP.

Service	Port and Protocol
FTP server	21/TCP
DNS server	53/UDP
HTTP Web server	80/TCP
HTTPS Web server	443/TCP

TCP	UDP
Keeps track of lost packets. Makes sure that lost packets are re-sent	Doesn't keep track of lost packets
Adds sequence numbers to packets and reorders any packets that arrive in the wrong order	Doesn't care about packet arrival order
Slower, because of all added additional functionality	Faster, because it lacks any extra features
Requires more computer resources, because the OS needs to keep track of ongoing communication sessions and manage them on a much deeper level	Requires less computer resources
Examples of programs and services that use TCP: - HTTP - HTTPS - FTP - Many computer games	Examples of programs and services that use UDP: - DNS - IP telephony - DHCP - Many computer games

Port forwards:
WAN network wants to communicate with LAN, then we need to tell the router, once there is a traffic to this IP and port, direct to our LAN network. Eg. Create a web server on LAN, with port 80.
Upnp(network discovery): Automatically port forwarding
Broadcast address: FF:FF:FF:FF
ARP (Address resolution protocol): It is used to associate MAC addresses with IP addresses and is a way for a computer to look up an unknown MAC address for a device that it wants to communicate with.
When a computer sends a DNS query, it must include the DNS server IP and MAC address, IP is given by DHCP, for MAC, it will create an ARP query first to broadcast the MAC address of the certain IP. Every time it receives a MAC address, it will save it to an ARP table for minutes.
WIFI and Hubs are both ***half duplex*** (only one device and communicate at a time)

Interview Question:
1. Factors to affect the reliability of a network:
   Frequency of failure and recovery time.
2. Factors to affect the performance:
   Large number of users, hardware, software, transmission medium types.
3. RIP(routing information protocol): used to find the best rout from source to destination by using hop count algorithm.
4. Netstat: a program to get the information of a TCP/IP connection (similar to Ping)

Socket:
https://realpython.com/python-sockets/

Python

```
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    pass  # Use the socket object without calling s.close().
```

The arguments passed to `socket()` specify the address family and socket type. `AF_INET` is the Internet address family for IPv4. `SOCK_STREAM` is the socket type for TCP, the protocol that will be used to transport our messages in the network.

`bind()` is used to associate the socket with a specific network interface and port number:

`bind()` is used to associate the socket with a specific network interface and port number:

Python

```
HOST = '127.0.0.1'  # Standard loopback interface address (localhost)
PORT = 65432        # Port to listen on (non-privileged ports are > 1023)

# ...

s.bind((HOST, PORT))
```

The values passed to `bind()` depend on the address family of the socket. In this example, we're using `socket.AF_INET` (IPv4). So it expects a 2-tuple: `(host, port)`.

`host` can be a hostname, IP address, or empty string. If an IP address is used, `host` should be an IPv4-formatted address string. The IP address `127.0.0.1` is the standard IPv4 address for the loopback interface, so only processes on the host will be able to connect to the server. If you pass an empty string, the server will accept connections on all available IPv4 interfaces.

`port` should be an integer from `1-65535` (`0` is reserved). It's the TCP port number to accept connections on from clients. Some systems may require superuser privileges if the port is < `1024`.

Continuing with the server example, `listen()` enables a server to `accept()` connections. It makes it a "listening" socket:

Python

```
s.listen()
conn, addr = s.accept()
```

`listen()` has a `backlog` parameter. It specifies the number of unaccepted connections that the system will allow before refusing new connections. Starting in Python 3.5, it's optional. If not specified, a default `backlog` value is chosen.

If your server receives a lot of connection requests simultaneously, increasing the `backlog` value may help by setting the maximum length of the queue for pending connections. The maximum value is system dependent. For example, on Linux, see `/proc/sys/net/core/somaxconn`.

One thing that's imperative to understand is that we now have a new socket object from `accept()`. This is important since it's the socket that you'll use to communicate with the client. It's distinct from the listening socket that the server is using to accept new connections:

Python

```
conn, addr = s.accept()
with conn:
    print('Connected by', addr)
    while True:
        data = conn.recv(1024)
        if not data:
            break
        conn.sendall(data)
```

After getting the client socket object `conn` from `accept()`, an infinite `while` loop is used to loop over blocking calls to `conn.recv()`. This reads whatever data the client sends and echoes it back using `conn.sendall()`.

Now let's look at the client, `echo-client.py`:

Python

```
#!/usr/bin/env python3

import socket

HOST = '127.0.0.1'  # The server's hostname or IP address
PORT = 65432        # The port used by the server

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    s.sendall(b'Hello, world')
    data = s.recv(1024)

print('Received', repr(data))
```

In comparison to the server, the client is pretty simple. It creates a socket object, connects to the server and calls `s.sendall()` to send its message. Lastly, it calls `s.recv()` to read the server's reply and then prints it.

SQL

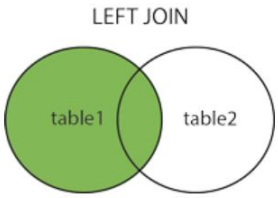
REGEXP "^[aeiou]." ".[^aeiou]$"

Right("absc",3)

IF(X>Y,"YES","NO")

```
SELECT N, IF(P IS NULL,"Root",IF((SELECT COUNT(*) FROM BST WHERE P=B.N)>0,"Inner","Leaf"))
FROM BST AS B ORDER BY N
```

SELECT something FROM tablename INNER JOIN another table ON condition;



INNER JOIN ON

LEFT JOIN ON

RIGHT JOIN ON

FULL JOIN

```
from Employee E inner join Company C on E.company_code = C.company_code group by E.company_code,
C.founder order by E.company_code
```

```
1 set @row := 0;
2 select repeat('* ', @row := @row + 1) from information_schema.tables where @row < 20
```

Add a column or drop a table:

ALTER TABLENAME

ADD/TABLE column name;

主键（Primary Key）：唯一标识一条记录，不能有重复，不允许为空。

外键（Foreign Key）：表的外键是另一表的主键，外键可以有重复，可以是空值。

唯一键（ Unique Key）：唯一标识一条记录，不能有重复，可以为空。
索引（Index）：该字段可以有重复值记录，可以有空值，如果是唯一索引，那么就不可以有重复的记录，可以有空值。

作用：
主键：用来保证数据完整性。
外键：用来和其他表建立联系，以保证数据的一致性和级联操作。
唯一键：用来用防止数据插入的时候重复。
索引：是提高查询排序的速度。

1. The default port for MySQL server is 3306
2. CHAR column length is fixed to the length that is declared while creating table. The length value ranges from 1 and 255
3. Drivers: PHP(script), PYTHON, JDBC,
4. % corresponds to 0 or more characters, _ is exactly one character in the LIKE statement.
5. LIKE and REGEXP operators are used to express with ^ and %.
6. Total 5 types of tables are present: MYISAM(default storage engin), Heap, INNO DB, Merge, ISAM
7. SELECT * FROM (top 50 rows)
   LIMIT 0,50;
8. Maximum of 16 indexed columns can be created for any standard table.
9. NOW():year-second, CURRENT_DATE():year-date
10. CREATE statement can create DATABASE, USER, TABLE, INDEX, TRIGGER,
11. Six triggers are allowed in Mysql, before/after insert/update/delete
12. Heap index should be not NULL, HEAP tables are present in memory and they are used for high speed storage on temporary, BLOB or TEXT fields are not allowed, auto_increment is not supported.
13. Float 4 bytes, double 8 bytes
14. MySQL server is open source

How MySQL Uses Indexes:

https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html

Operation System:

1. Six types of process scheduling algorithms are:

- First Come First Serve (FCFS)
- Shortest-Job-First (SJF) Scheduling
- Shortest Remaining Time
- Priority Scheduling
- Round Robin Scheduling
- Multilevel Queue Scheduling

TYPE:

- Batched operating systems
- Distributed operating systems
- Timesharing operating systems
- Multi-programmed operating systems
- Real-time operating systems

Process: an executing program
Thread: a basic unit of CPU utilization, it contains a thread id, program counter, register, and a stack.

SMP:

SMP stands for **s**ymmetric MultiProcessing. It is the most common type of multiple processor system. In SMP, each processor runs an identical copy of the operating system, and these copies communicate with one another when required.

Spooling:
When many applications send output to a printer at the same time. Spooling keeps these jobs to a disk file and queue them.
Semaphore: Semaphore is a protected variable or abstract data type that is used to lock the resource being used, binary Semaphore(0 or 1, same as mutex) and counting semaphore.

Starvation: A waiting process does not get the resources for a long time, because it is allocated to other process,
Aging: a solution to starvation.
Logical address and physical address: address generated by CPU and memory unit.
Overlay: make a process larger than the allocated memory to make sure only important instructions and data are at any given time are kept in memory

What is Virtual Memory? How is it implemented?

Virtual memory creates an illusion that each user has one or more contiguous address spaces, each beginning at address zero. The sizes of such virtual address spaces is generally very high.
The idea of virtual memory is to use disk space to extend the RAM. Running processes don't need to care whether the memory is from RAM or disk. The illusion of such a large amount of memory is created by subdividing the virtual memory into smaller pieces, which can be loaded into physical memory whenever they are needed by a process.
Ideally, the data needed to run applications is stored in RAM, where they can be accessed quickly by the CPU. But when large applications are being run, or when many applications are running at once, the system's RAM may become full.

To get around this problem, some data stored in RAM that is not actively being used can be temporarily moved to virtual memory (which is physically located on a hard

drive or other storage device). This frees up space in RAM, which can then be used to accommodate data which the system needs to access imminently.

A computer can only run threads and manipulate data that is stored in RAM rather than virtual memory. And it takes a non-negligible amount of time to swap data that is needed into RAM

One potential problem with virtual memory is that if the amount of RAM present is too small compared to the amount of virtual memory then a system can end up spending a large proportion of its CPU resources swapping data back and forth. (Thrashing). Virtual memory<=1.5*RAM

https://www.enterprisestorageforum.com/storage-hardware/virtual-memory.html#:~:text=Virtual%20memory%20is%20an%20area,accessed%20quickly%20by%20the%20CPU.

python is slower than c because it is an interpreter language

GIL(Global Interpreter Lock): allows only one thread to hold the control of the Python interpreter.

https://realpython.com/python-gil/#:~:text=The%20Python%20Global%20Interpreter%20Lock%20or%20GIL%2C%20in%20simple%20words,at%20any%20point%20in%20time.

Python uses reference counting for memory management(count the variable reference number for an object, when it reach 0, release memory), GIL ensures the memory thread-safe

In multiprocessing: Each Python process gets its own Python interpreter and memory space so the GIL won't be a problem

Page replacement Algorithm(Virtual memory or Cache):

LRU(least recently used): In **L**east **R**ecently **U**sed (LRU) algorithm is a Greedy algorithm where the page to be replaced is least recently used. The idea is based on locality of reference, the least recently used page is not likely

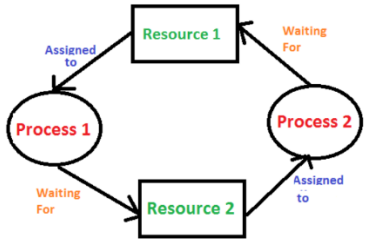
First In First Out (FIFO)

Optimal Page replacement: In this algorithm, pages are replaced which would not be used for the longest duration of time **in the future**(往后查找，看哪个最后用).
Page fault occurs when a requested page is mapped in virtual address space but not present in memory.

Continuous memory management:
Partition management:
Fixedpartition: easy to implement, inefficient.
Dynamic partition:分配内存的方法： nearest-fit(从第一个分区的内存开始查找), next fit(从上一个分配的分区开始查找), best fit, worst fit(find biggest memory space)
Solution: each memory is 2^n

Discontinuous memory management(Virtual memory):
Paging：fixed size pages and frames
优点：没有外碎片（external fragmentations），一个程序不用连续存放，便于改变程序占用空间的大小，
Segmentation: variable size
优点：没有内碎片，便于实现内存共享，less processing
缺点，程序必须完全放入内存。

DeadLock:



Condition:
Mutual Exclusion: One or more than one resource is non-shareable (Only one process can use at a time)
Hold and Wait: A process is holding at least one resource and waiting for resources.
No Preemption: A resource cannot be taken from a process unless the process releases the resource.
Circular Wait: A set of processes are waiting for each other in circular form

Solution:
1. Wait for all sources are available to start a process.
2. Request for resource, if rejected, release own resource. Or high priority have the resource

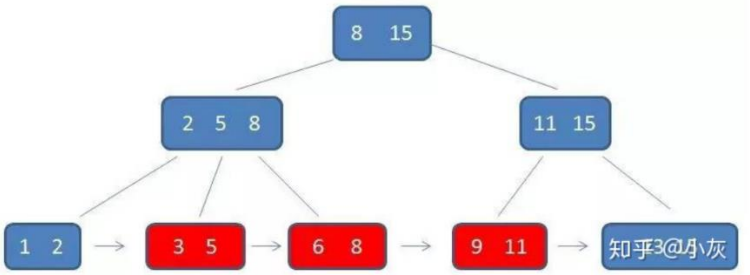
Banker's algorithm: To avoid deadlock

Linux:

Frequently used commands: https://www.hostinger.com/tutorials/linux-commands

字节：
面试题
1. reverse 一个 linkedlist
2. 一个 unsorted array,排列成正负交错，空间复杂度 O(1)
3. 把一个 string 里的分数找出来，变成小数
4. a+b=k, 2Sum
5. 灌水
6. m*n 的格子，有多少个区域
7. https,
8. mutiprocesing 和 multithreading
9. why choose **B+** tree(balance+ tree)：(every element in parent node is included in child node, so all leaves node contains all the value. Leaves are in the same height and formed an ascending linked list, middle node only do index, do not store value)
   - Less IO(.单一节点存储更多的元素，使得查询的 IO 次数更少。)(IO 等于树的高度)
   - More stable(所有查询都要查找到叶子节点，查询性能稳定。)
   - Interval research is efficient(所有叶子节点形成有序链表，便于范围查询)

B- tree is used in Non-relational database(MongDB)



10. difference between GET and POST
https://medium.com/theagilemanager/difference-between-the-http-requests-post-and-get-3b4ed40164c1#:~:text=The%20GET%20and%20POST,your%20password%20should%20use%20POST%20
.
The `GET` and `POST` are two different types of HTTP requests. `GET` is used for viewing something, without changing it, while `POST` is used for changing something. For example, a search page should use `GET` to get data while a form that changes your password should use `POST`. Essentially `GET` is used to retrieve remote data, and `POST` is used to insert/update remote data.
GET is cached but POST is never cached

Java

The four principles of object-oriented programming are
Encapsulation: all methods and data are in an object, they keep to themselves.
Abstraction: you don't have to know how a method is implemented, you know how

to use it. (you don't have to know how a coffee machine works, you just press button)

Inheritance: The child class reuses all the fields from its parent(**extends**), When a class extends a parent class, it inherits all the non-private fields and methods

Polymorphism: mixed type, a class can implements from many interfaces.

1. Composite data type:

```
struct circle {
  float x, y; // (x,y) coordinate of the center.
   float r; // radius
}
```

2. A **class** is a data type with a group of functions associated with it(encapsulation). We call the functions as *methods* and the data in the class as *fields* (or *members*, or *states*, or *attributes*[4]). **Objects** are *instances* of a class.
3. Data hiding: **private** variable cannot be changed outside the class. Public can
4. A method that initializes an object is called a ***constructor***
5. A variable of **primitive** type stores the *value* instead of a reference to the value. Java supports eight *primitive* data types: `byte`, `short`, `int`, `long`, `float`, `double`, `boolean` and `char`
6. If a class promises to implement an **interface**, then we are guaranteed that the methods defined in the interface are already implemented in the class. When declaring the methods that already in the interface @override annotation is optional to add above the method. Exp:
   interface shape{…
   public double getArea();
   }
   class point **implements** shape{…
   @override
   Public double getArea{}
   }
   Class A implements B, C{} Then class A can use the methods in both class C and B.
   Methods in interface are abstract method(without body), with **default** keyword, it can has body.
7. When a class **extends** a parent class, it inherits all the non-private fields and methods(**inheritance**), a class can **only** inherits from one class.

8.

S.No.	Extends	Implements
1.	By using "extends" keyword a class can inherit another class, or an interface can inherit other interfaces	By using "implements" keyword a class can implement an interface
2.	It is not compulsory that subclass that extends a superclass override all the methods in a superclass.	It is compulsory that class implementing an interface has to implement all the methods of that interface.
3.	Only one superclass can be extended by a class.	A class can implement any number of an interface at a time
4.	Any number of interfaces can be extended by interface.	An interface can never implement any other interface

9. Two methods in a class can have the same name and still co-exist peacefully together. This is called ***overloading.*** Keyword **Super** is used to initialize the parent private fileds.
10. Use **final** to prevent a class being inherited. Use final to prevent a method being overrided.
11. Widening reference conversion: child class to parent class.
    Narrowing reference conversion: parent class to child class.
12. Checked exception: **Try**{}catch{}finally{} the finally blocks are always executed even return or throw is called in catch{} block.
13. `throw` is to generate an exception, `throws` is to specify that the exception(s) thrown by a method.
14. Java has a rule that a class can extend only one **abstract** class(abstract class is designed for subclass), but can implement multiple interfaces (fully abstract classes). Abstract class cannot be initialized. A method without body(implementation) is known as abstract method. A class who has an abstract method must be declared as abstract class.
15. **Generic** type: <T>, <?>
16. **Type erasure**: The type argument is erased during compile time. Ex: during compile time, `Queue<Circle>` is translated into `Queue`.
17. **Raw type:** Java allows generic class to be used without the type argument
18. **Hashcode:** a=b -> a.hashcode()=b.hashcode()
19. Variable is accessed from within **inner class**, needs to be **final or effectively final**.

SOLID design principle:

https://www.c-sharpcorner.com/UploadFile/damubetha/solid-principles-in-C-Sharp/

Single Responsible Principle: every class or a single structure of code should only have one job to do.

Open/Close Principle: A software module is open for extension but close to modification

Liskov Substitution Principle: a derived class doesn't affect the behavior of its parent class.

Interface Segregation Principle: a class should not be forced to implement the interface it doesn't use.

Dependency inversion principle: high-level class should not depend on low-level class. Both should depend upon abstractions. Secondly, abstractions should not depend upon details. Details should depend upon abstractions.