**从定义、来源、实现、问题、优化、应用方面系统性地回答**

# C++：

## 基本-（基础）：

int类型占多少字节？  
一个字节的int能表示多少个数？

最大的2^8怎么表示？全为1换算后为多少？

有符号的范围是多少？-128怎么来的？

字节对齐 \*

1、各个硬件平台对存储空间的处理不尽相同  
2、会对CPU的存取效率产生影响

c++ 类型转换介绍下

const\_cast

任何隐式转换：static\_cast

子类到基类：dynamic\_cast

reinterpret\_cast

无符号数与有符号数的区别

C++静态局部变量为什么线程安全

auto 声明需要初始化吗

strlen和sizeof的区别

指针和引用的区别 \*

内联函数介绍一下 内联函数体有没有什么要求 联合Union

float == float，有无风险

怎么在main函数之前和之后执行代码；

C++构造函数调用顺序

虚表指针在初始化列表之前被赋值，可以放在初始化列表之后赋值吗？即顺序是：列表初始化、虚表指针赋值、构造函数体？？

**为了防止在初始化列表出现调用虚函数的情况**！！！

无继承时：

1、分配内存  
2、初始化列表之前赋值虚表指针  
3、列表初始化  
4、执行构造函数体

有继承时：

1、分配内存  
2、基类构造过程（按照无继承来）  
3、初始化子类虚表指针  
4、子类列表初始化  
5、执行子类构造函数体

## 基本-（一些关键字）

static关键字

* 静态成员变量（面向对象）
* 静态成员函数（面向对象）
* 静态全局变量（面向过程）
* 静态局部变量（面向过程）
* 静态函数（面向过程）

static应用场景

 extern “C” 的作用

## 基本-（面向对象特性及STL）

 c++多态怎么实现，虚函数的原理 \*\*

C++哪几种构造函数

给c++添加个GC机制

c++和c近几年的机制变化

C++为什么是面向对象的程序设计语言 C++有哪些特性 继承方式

STL容器，[源码](https://www.nowcoder.com/jump/super-jump/word?word=%E6%BA%90%E7%A0%81)。。。。vector、list的联系区别vector是如何扩容的，什么时候失效，map和set之类

介绍stl的vector,list,unordered\_map,map  
unordered\_map的哈希冲突怎么解决的？

重载，重写，覆盖。

父类指针指向子类对象，调用非虚函数，是子类的还是父类的

为什么可以由子类对象new一个父类对象，反之不行

迭代器失效的情况分析（我说了序列型容器在erase一个迭代器元素后，后面的元素也会erase，导致迭代器失效，面试官说不完全正确，但是想不到别的失效情况了）

  1.由于插入元素，使得容器元素整体“迁移”导致存放原容器元素的空间不再有效，从而使得指向原空间的迭代器失效。  
  2.由于删除元素使得某些元素次序发生变化使得原本指向某元素的迭代器不再指向希望指向的元素。

## 基本-（内存管理与布局）

C++代码到可执行程序的过程，详细描述四个阶段 \*

c++的异常处理机制

写个类，问在内存中如何存放，包括成员变量和各种函数存放的位置

虚函数指针存放在内存中什么位置

为什么不能用宏代替大型函数

智能指针的实现，weak\_ptr为什么能解决循环引用

全局静态变量在什么阶段初始化。除了.bss 还有什么段，线程的栈你觉得放哪，堆栈中间的部分是什么

内存堆和栈有什么区别？

堆和栈哪个存储效率高，为什么？空间哪个大，为什么？

访问NULL指针为什么会报错？编译期报错还是运行期报错？编译器会编译这个指令，所以应该是运行期报错，0号地址也是地址，如何实现这个访问0号地址报错这个机制？内存是如何寻址的？

new和malloc的区别

讲一下内存分区，和每个分区存的什么。

怎么解决内存泄漏？

有哪几种智能指针？发生循环引用怎么办呢? 那你给我写一个循环引用的例子吧。

程序的内存布局

## 基本- （一些底层原理）

函数调用过程， “保存现场”具体指哪些，函数调用涉及到的汇编指令

C++代码到可执行程序的过程，详细描述四个阶段 \*

可执行程序具体来说是什么

机器指令是完全由0/1组成的，那么机器怎么对这些0/1进行划分

静态库和动态库的区别

静态链接与动态链接

字节流和字符流说一下

程序从产生到运行的全过程；

早期c语言函数怎样实现多个返回值

关于函数返回的一件小事——是返回值还是返回引用？

define的宏定义过程

## 基本- (一些设计与实现)

C++设计一个定时器

实现线程安全的智能指针

Sring 类 补充（析构 构造等 delete关键字 各种运算符重载…const情况.

\*\*\* 线程池怎么实现的

# C#

c#中静态类和普通类在存储方面有什么区别？  
静态类中能不能调用非静态类的方法？为什么？  
c#中值类型和引用类型有什么区别？  
判断两个引用类型是否相同，原理是什么？  
值类型呢？  
装箱拆箱是什么过程？

c#基本类型有哪几个？

C#垃圾回收

# 数据结构：

## 数据结构

[红黑树](https://www.nowcoder.com/jump/super-jump/word?word=%E7%BA%A2%E9%BB%91%E6%A0%91)和AVL树，[红黑树](https://www.nowcoder.com/jump/super-jump/word?word=%E7%BA%A2%E9%BB%91%E6%A0%91)的平衡因子

[哈希表](https://www.nowcoder.com/jump/super-jump/word?word=%E5%93%88%E5%B8%8C%E8%A1%A8)原理 散列函数

栈和队列区别，应用场景 \*

一个[链表](https://www.nowcoder.com/jump/super-jump/word?word=%E9%93%BE%E8%A1%A8)频繁增删改查，改用什么结构， 说了B+树或者[红黑树](https://www.nowcoder.com/jump/super-jump/word?word=%E7%BA%A2%E9%BB%91%E6%A0%91)

描述b+树， 为什么用[红黑树](https://www.nowcoder.com/jump/super-jump/word?word=%E7%BA%A2%E9%BB%91%E6%A0%91)不用其他AVL树等

前序遍历/中序遍历/后序遍历的递归和栈的实现方式

栈实现队列 \*

无锁队列实现、CAS底层怎么实现的

hash表冲突过多怎么办

unordered\_map的底层原理说一下，它是怎么解决哈希冲突的

set的实现，查找速度

B+树的实现原理，为什么innodb不用[红黑树](https://www.nowcoder.com/jump/super-jump/word?word=%E7%BA%A2%E9%BB%91%E6%A0%91)用B+树

mmap是什么

# **算法**：

将 ipv4 地址字符串 (仅包含数字，点，空格) 转化成 32 位整数

若有一个函数，返回0的概率是60%，返回1的概率是40%，如何利用这个函数获得50%的概率

\* 生成0 1 概率为p(1-p)

1. \* 生成1 0概率为(1-p)p
2. \* 则相等的

设计一个栈，O(1)复杂度求最小值

各种排序复杂度，最坏最好

散列函数

（1）直接定址法：

（2）相乘取整法：

（3）平方取中法：

（4）除留余数法：

（5）随机数法：

（1）开放定址法：

线性探测法（线性探测再散列）

平方探测法

（2）链地址法（开散列法）

（3）再哈希法：（双散列法）

（4）建立一个公共溢出区：

手撕域名反转；

# 设计模式

面向对象特点，什么是多态；

面向对象六大基本原则；

"封装、"多态"、"继承"

"单一职责原则"、"开放封闭原则"、"里氏替换原则"、"依赖倒置原则"、"接口分离原则"、"迪米特原则（高内聚低耦合）"

单例模式有几种写法？说说 怎么写单例模式才能保证多线程安全？

手撕线程安全的单例模式

写一个懒汉式单例模式

为什么单例模式要把构造函数放在private

说一下设计模式

# 操作系统：

## 进程线程/协程：

进程，线程区别 进程和线程的关系 进程中一定有线程吗，线程同步方式 \*\*\*

进程切换时，会发生什么，进程的状态 \*

进程和线程的区别，共享/不共享什么？线程间通讯?

多线程是啥概念

多核处理器的进程调度

对比I/O多路复用和多线程

中断、DMA、通道

进程调度[算法](https://www.nowcoder.com/jump/super-jump/word?word=%E7%AE%97%E6%B3%95)

进程间通信方式 线程通信的方式？ 说一下中断 \*

cpu核心数和线程还是进程关系更紧密？

线程里面sleep和wait什么区别（从方法所属类，调用的效果，是否释放锁答的）

sleep不释放锁和wait释放锁，底层怎么做的

## 锁

生产者消费者模型的好处

1. **解耦**

**2、支持并发**

**3、支持忙闲不均**

互斥锁有了解吗 互斥锁、自旋锁、读写锁 互斥锁和自旋锁，怎么选择 \*

|  |
| --- |
| 1. #include <pthread.h> //多线程、互斥锁所需头文件 2. pthread\_mutex\_t r\_mutex = PTHREAD\_MUTEX\_INITIALIZER; *//定义和初始化互斥锁* 3. pthread\_mutex\_t w\_mutex = PTHREAD\_MUTEX\_INITIALIZER; 4. int readers = 0; *//记录读者的个数* 6. 写模式： 7. pthread\_mutex\_lock(&w\_mutex); 8. 写写写…… 9. pthread\_mutex\_unlock(&w\_mutex);  12. 读模式： 13. pthread\_mutex\_lock(&r\_mutex); 15. if(readers == 0) 16. pthread\_mutex\_lock(&w\_mutex); 17. readers++; 18. pthread\_mutex\_unlock(&r\_mutex); 19. 读读读…… 20. pthread\_mutex\_lock(&r\_mutex); 21. readers- -; 22. if(reader == 0) 23. pthread\_mutex\_unlock(&w\_mutex); 24. pthread\_mutex\_unlock(&r\_mutex); |

互斥锁、读写锁、自旋锁都有什么区别 用互斥锁能实现读写锁吗，需要几个，讲一下思路

说一下死锁的条件 如何打破死锁

## 内存管理：

为什么有虚拟地址？虚拟地址怎么实现？

Lru和设计LFU

虚拟内存简述 虚拟内存解释 虚拟地址和内存地址的转换

**访问3次内存，第一次是段表，第二次是页表，第三次是真正物理内存**

页置换简述 页在你的理解中究竟是什么。

寄存器的作用

地址空间是什么

分段分页

共享内存的原理

操作系统里的缓存

缺页中断；  
页面置换[算法](https://www.nowcoder.com/jump/super-jump/word?word=%E7%AE%97%E6%B3%95)；

## 其他：

cache的用途

计算机的存储结构

代码编译有哪几个阶段，编译阶段能再细说一点在做什么吗？

同步和异步是什么（不限范围，随便谈一下你对于这两个词的理解）

用户态和内核态

同步io和异步io，谈谈你的理解

io多路复用？介绍一下select，poll和epoll

中断，系统调用

进程疯狂申请内存，new， kill掉以后内存会不会释放 会！

epoll是怎么实现的？有哪几个步骤？ET和LT的区别

CPU内部结构

# 计网：

## TCP UDP

udp，tcp 区别，用处

ip数据包

udp和tcp在哪一层，ip？http？http为什么？  
tcp、udp特点、区别？  
哪些场景用udp、哪些tcp？

moba[游戏](https://www.nowcoder.com/jump/super-jump/word?word=%E6%B8%B8%E6%88%8F)用哪种？但udp会丢包啊？

udp如何保证可靠性

顺序到达不一致怎么处理？从协议层入手还是应用层？  
tcp为什么要三次握手？两次不行？

为什么要四次挥手？为什么多一次？

三次握手和四次挥手，这里的seq是什么机制？\*

Tcp四次挥手过程，TIME\_WAIT的作用 \*\*

TCP协议 拥塞控制 流量控制 滑动窗口 TCP怎么保证可靠传输的 \*\*\*

拥塞控制，流量控制针对的分别是什么问题

TCP的字节流是什么机制

TCP握手连接确认号为什么要+1，不变不行吗？为什么？TCP头部字段以及序列号的作用；

套接字在哪一层，能封装哪些协议

## 应用层：

什么是cookie

端口号的最大个数

浏览器中输入一个url，到网页显示的过程描述

GET和POST的区别，HTTP还有哪些请求方法 HTTP中POST的方式有哪些？\*

端口是什么，位于哪一层

说一下HTTP和HTTPS HTTPS是使用非对称加密还是对称加密？https为啥安全\*\*

HTTPS的连接过程说一下吧（说了SSL的加密过程，包括证书CA、非对称和对称加密等）描述一下非对称加密和对称加密 两种加密如何配合使用 \*\*

HTTP和HTTPS有什么区别，SSL是怎样实现安全传输（具体过程、对称非对称）；

数字证书

## 基础与底层协议

主机A，主机B，分别有IP地址。请从ip层及以下的层面描述主机A怎么和主机B进行信息交互。

app向服务器发送一个请求，完整的过程

osi哪七层？\*

如果有很多从[客户端](https://www.nowcoder.com/jump/super-jump/word?word=%E5%AE%A2%E6%88%B7%E7%AB%AF)发来的请求，结果大量的失败，分析原因

网络中ip地址和mac地址的关系，为什么要有mac地址

说一下网络的信息传递

网络各层校验用的什么？ ARP协议？

ipv4地址分类（ABCDE，说了每一类IP的分段）

怎么判断大小端；

# Unity与图形学

## Unity

UGUI的渲染顺序是什么？

canvas有三种渲染模式可供选择，分别是Overlay,Camera,World  
其中Overlay为覆盖模式，即永远最后渲染，覆盖其他物体和UI。Camera为相机模式，渲染顺序依据相机。World为世界模式，并不因相机而改变，至于距离有关。  
实际工程中，我们会运用多个canvas，那么各种模式下渲染顺序又是如何？

第一种情况，当有多个canvas并且渲染模式都为Overlay。  
这种情况下，渲染顺序是由canvas组件下的Sort Order决定的，值越大的越后渲染。

第二种情况，当有多个canvas并且渲染模式都为Camera。  
这种情况下，渲染顺序首先由Rendener Camera的Depth值决定，值越大越后渲染。  
如果Depth值相同，那么由canvas组件下的Sortint Layer顺序决定，顺序越后则越后渲染。  
如果Depth值和Layer值都相同的情况下，渲染顺序由Order in Layer决定，值越大越后渲染。  
（注意：若Rendener Camera都是同一个摄像机，则不考虑Depth情况）

第三种情况，当有多个canvas并且渲染模式都为World。  
这种情况下，渲染顺序由canvas组件下的Sortint Layer顺序决定，顺序越后则越后渲染。  
如果Layer值都相同的情况下，渲染顺序由Order in Layer决定，值越大越后渲染。  
特别注意，当Layer和Order值都相同的情况下，此时渲染顺序由canvas距离Render Camera的距离决定，距离越近越后渲染。（  
只有World模式有这种情况，因其他模式下canvas都是位置固定不可移动的。并且以上情况只考虑canvas都出现在Render Camera摄像范围的情况下。）

第四种情况，当有多个canvas并且渲染模式都存在的情况下。  
这种情况下，首先Overlay模式的canvas永远是最后渲染，并且同为Overlay模式的canvas在Sort Order的值越大时越后渲染。  
其次，Camera和Overlay同时存在的情况下有两种情况：若使用的不同的相机，则由摄像机的Depth决定，值越大越后渲染。若使用的是相同的相机，则是由canvas距离摄像机的距离决定的，距离越近的越后渲染。

层级上，A在B的下面，同一个canvas，order和layer相同，可不可能A显示在B上面？

做过动画吗？骨骼动画为什么制作一个可以应用在所有对象上？  
线程协程什么关系？还是说两者一致？  
开启一个协程后他的第一次执行在update之前还是之后？

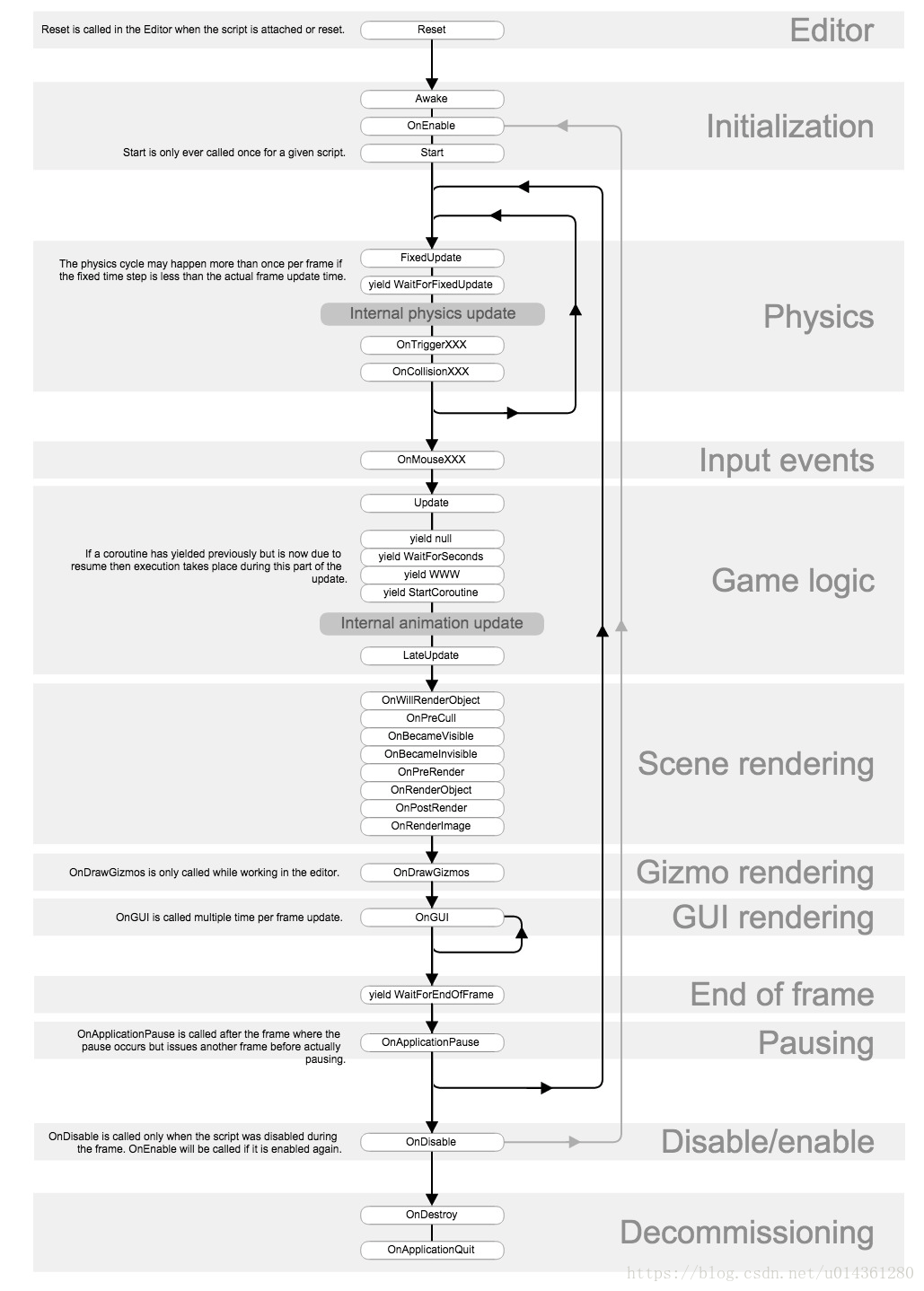
WWW用过吗？  
Unity只有一个主线程吗？

对shader怎么理解的，他扮演了个什么样的角色？自己写过吗？难度怎么样？

assetbundle在本地还是服务器上？为啥在本地？  
lua写过吗？多久？lua和c#本质的不同？优点？

粒子系统遵不遵循unity的生命周期？

粒子系统遵不遵循unity的生命周期？  
如果把timescale设为0那粒子系统还播不播？动画系统呢？



## 图形学

介绍下渲染流程

介绍那几个变换