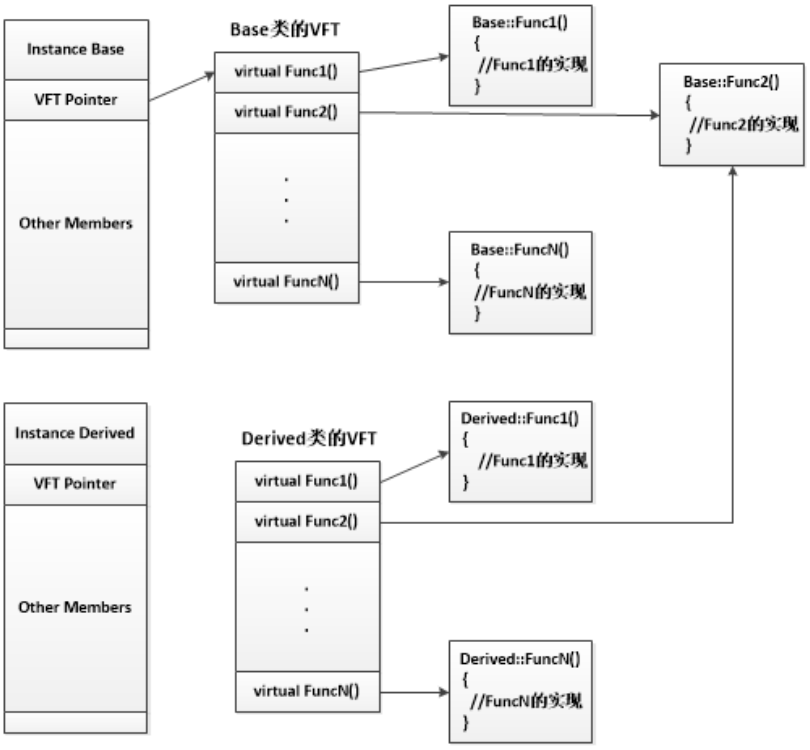
xb腾讯现场一面

C++

构造函数：



编译器见到这种继承层次结构后，知道Base定义了虚函数，并且在Derived类中覆盖了这些函数。在这种情况下，编译器将为实现了虚函数的基类和覆盖了虚函数的派生类分别创建一个虚函数表(Virtual Function Table,VFT)。也就是说Base和Derived类都将有自己的虚函数表。实例化这些类的对象时，将创建一个隐藏的指针VFT*，它指向相应的VFT。可将VFT视为一个包含函数指针的静态数组，其中每个指针都指向相应的虚函数。Base类和Derived类的虚函数表如下图所示：

https://blog.csdn.net/u011000290/article/details/50498683

3、底层机制

在每一个含有虚函数的类对象中，都含有一个VPTR，指向虚函数表。

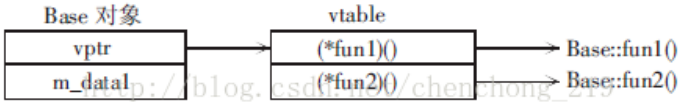


图 2 Base 对象内存空间示意图

派生类也会继承基类的虚函数，如果宅派生类中改写虚函数，虚函数表就会受到影响；表中元素所指向的地址不是基类的地址，而是派生类的函数地址。

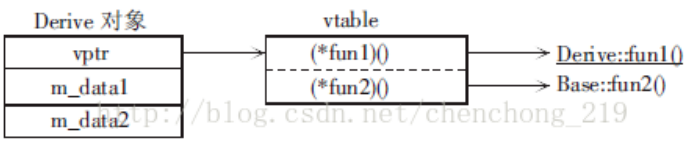


图 3 Derive 对象内存空间示意图

当执行语句pBase->fun1()时，由于PBase指向的是派生类对象，于是就调用的Deriver::fun1()。

构造函数为虚

析构函数为虚

手写lambda表达式的形式

https://www.cnblogs.com/DswCnblog/p/5629165.html

序号	格式
1	[capture list] (params list) -> return type {function body}
2	[capture list] (params list) {function body}
3	[capture list] {function body}

const 作为参数

C++ 11新增的默认会生成的函数

多态的实现机制

map和hashmap的区别，什么时候使用

删除map里元素，原来的迭代器还能用吗

引用，需要注意的

没看过STL源码

linux操作系统

shell编程

两个文件里有
qq号 时间
两个文件，找出两天都登陆了的qq号。
awk 求交集
编译main.cpp 依赖两个lib，lib1和lib2，lib1依赖lib2，如何写g++命令。

手写代码题

包含max的栈：
贴瓷砖问题：斐波那契数列

智力题

100本书：每人可以拿1-5本书，如何拿到最后一本书。保证最后剩6本。
11-12点约见面，早到的可以等15分钟，问相见的概率。组合数学。7/16

多线程

查看线程打开的文件

分析线程CPu占用

查看系统IO占用

主要是查看各个线程的信息

如何分析内存泄漏

如何创建共享内存

进程间通信的机制

fork后共享fd吗？共享锁吗，锁时共享的

锁有哪些锁，各有什么优缺点

有名管道无名管道

查看句柄泄漏:

如果频繁的打开文件，或者打开网络套接字而忘记释放就会有句柄泄露的现象。在linux系统中对进程可以调用的文件句柄数进行了限制，在默认情况下每个进程可以调用的最大句柄数是1024个，如果超过了这个限制，进程将无法获取新的句柄，而从导致不能打开新的文件或者网络套接字，对于线上服务器即会出现服务被拒绝的情况。在linux系统中可以通过ulimit-n查看每个进程限制的最大句柄数，通过ulimit -HSn 10240修改进程的最大句柄数。当句柄数目达到限制后，就回出现"too many files open"。
查看进程占用的句柄数有几种办法：
1） 通过cat/proc/pid/fd可以查看线程pid号打开的线程；
2） 通过lsof命令， /usr/sbin/lsof-p 21404 命令结果如下：
https://blog.csdn.net/daofengliu/article/details/38171953

网络

TCP 三次握手，四次挥手的过程

time wait

如果客户端发送了SYN就不管了会怎么办:DDOS如何防止DDOS?

https://blog.csdn.net/m13295105247/article/details/78958012

SYN Flood
SYN Flood是互联网上最经典的DDoS攻击方式之一，最早出现于1999年左右，雅虎是当时最著名的受害者。SYN Flood攻击利用了TCP三次握手的缺陷，能够以较小代价使目标服务器无法响应，且难以追查。
标准的TCP三次握手过程如下：客户端发送一个包含SYN标志的TCP报文，SYN即同步（Synchronize），同步报文会指明客户端使用的端口以及TCP连接的初始序号；
服务器在收到客户端的SYN报文后，将返回一个SYN+ACK（即确认Acknowledgement）的报文，表示客户端的请求被接受，同时TCP初始序号自动加1；
客户端也返回一个确认报文ACK给服务器端，同样TCP序列号被加1。
经过这三步，TCP连接就建立完成。TCP协议为了实现可靠传输，在三次握手的过程中设置了一些异常处理机制。第三步中如果服务器没有收到客户端的最终ACK确认报文，会一直处于SYN_RECV状态，将客户端IP加入等待列表，并重发第二步的SYN+ACK报文。重发一般进行3-5次，大约间隔30秒左右轮询一次等待列表重试所有客户端。另一方面，服务器在自己发出了SYN+ACK报文后，会预分配资源为即将建立的TCP连接储存信息做准备，这个资源在等待重试期间一直保留。更为重要的是，服务器资源有限，可以维护的SYN_RECV状态超过极限后就不再接受新的SYN报文，也就是拒绝新的TCP连接建立。 SYN Flood正是利用了上文中TCP协议的设定，达到攻击的目的。攻击者伪装大量的IP地址给服务器发送SYN报文，由于伪造的IP地址几乎不可能存在，也就几乎没有设备会给服务器返回任何应答了。因此，服务器将会维持一个庞大的等待列表，不停地重试发送SYN+ACK报文，同时占用着大量的资源无法释放。更为关键的是，被攻击服务器的SYN_RECV队列被恶意的数据包占满，不再接受新的SYN请求，合法用户无法完成三次握手建立起TCP连接。也就是说，这个服务器被SYN Flood拒绝服务了。

DDos攻击的解决方法

1. 异常流量清晰过滤，封ip
2. 防火墙
3. 阿里云盾，切换公网ip

为什么项目用udp

流量控制有哪些机制：滑动窗口，好像还有一个，Nagle算法，糊涂窗口综合症。

https://blog.csdn.net/yechaodechuntian/article/details/25429143

必须考虑传输速率
可以用不同的机制来控制TCP报文段的发送时机。如： <1>. TCP维持一个变量，它等于最大报文段长度MSS。只要缓存中存放的数据达到MSS字节时，就组装成一个TCP报文段发送出去。<2>. 由发送方的应用进程指明要求发送报文段，即TCP支持的推送(push)操作。<3>. 发送方的一个计时器期限到了，这时就把已有的缓存数据装入报文段(但长度不能超过MSS)发送出去。
Nagle算法：若发送应用进程把要发送的数据逐个字节地送到TCP的发送缓存，则发送方就把第一个数据字节先发送出去，把后面到达的数据字节都缓存起来。当发送方接收对第一个数据字符的确认后，再把发送缓存中的所有数据组装成一个报文段再发送出去，同时继续对随后到达的数据进行缓存。只有在收到对前一个报文段的确认后才继续发送下一个报文段。当数据到达较快而网络速率较慢时，用这样的方法可明显地减少所用的网络带宽。Nagle算法还规定：当到达的数据已达到 发送窗口大小的一半或已达到报文段的最大长度时，就立即发送一个报文段。
另，**糊涂窗口综合证**：TCP接收方的缓存已满，而交互式的应用进程一次只从接收缓存中读取1字节（这样就使接收缓存空间仅腾出1字节），然后向发送方发送确认，并把窗口设置为1个字节（但发送的数据报为40字节的的话）。接收，发送方又发来1个字节的数据（发送方的IP数据报是41字节）。接收方发回确认，仍然将窗口设置为1个字节。这样，网络的效率很低。要解决这个问题，可让接收方等待一段时间，使得或者接收缓存已有足够空间容纳一个最长的报文段，或者等到接收方缓存已有一半空闲的空间。只要出现这两种情况，接收方就发回确认报文，并向发送方通知当前的窗口大小。此外，发送方也不要发送太小的报文段，而是把数据报积累成足够大的报文段，或达到接收方缓存的空间的一半大小。

累计ACK的好处

项目：

主要问了文件系统项目

数据库

MySQL问了各种引擎，说不了解
了解别的数据库吗？key-value的，说了解redis，具体实现不了解。