**原文链接：<https://zhuanlan.zhihu.com/p/405416851>**

**一、个人信息**

1、自我介绍 项目(重要)

2、项目中最大的挑战，花费最长时间的项目，比较有成就感的事情

3、项目中印象最深的bug

4、有什么想问的吗

5、适合业务测试还是技术测试

6、为什么选择字节

7、分别用3个词描述自己的优缺点：团队协作能力强、学习能力强、主动乐观

**二、测试**

**1、对测试开发的认识，为什么想做测试开发，测试测开的理解**

**2、软件的分类**

按功能分为：系统软件、应用软件；按架构：单机、C/S、B/S

1. **什么是接口测试**

接口测试是测试系统组件间接口的一种测试。接口测试主要用于检测外部系统与系统之间以及内部各个

子系统之间的交互点。测试的重点是要检查接口参数传递的正确性，接口功能实现的正确性，输出结果

的正确性，以及对各种异常情况的容错处理的完整性和合理性。

1. **微信红包的测试用例(功能、ui、性能、安全、弱网、易用性、兼容、中断）**

功能测试

1.红包最多可以输入的金额；

2.红包一次性可以发送的最大个数；

3.在输入红包的钱数和个数时只能输入数字；

4.当余额不足时，红包发送失败；

5.群里发送的红包自己是否可以领取；

6.发送的红包别人是否可以领取；

7.红包超过24小时是否可以领取；

8.红包超时未领取，是否退回原账户；

9.是否可以自己选择支付方式；

10.红包描述可以输入的最大字符；

11.余额不足时，是否可以切换支付方式；

12. 给单人、群发红包均可发；

13.红包发送成功后，余额是否发生变化；微信是否会收到通知；

14.红包发送失败，余额不会产生变化；

UI界面测试

1.输入界面是否清晰可见；

2.红包界面颜色搭配是否美观；

3.输入金额界面是否有错别字；

性能测试

1.发红包成功后的跳转时间；

2.红包超时未领取后的退款时间；

3.网速较差时，发红包的时间；

4.同时支持多少人并发发红包

安全测试

1. 使用https请求后端接口，使用app无证书无法进行抓明文包；

2. 接口有用户身份验签。

弱网测试

1.弱网下是否可以发送红包；

2.弱网下发送红包的时间；

易用性测试

1.是否可以选择默认支付方式；

2.余额不足时，是否可以切换支付方式；

3.是否支持密码支付和指纹支付；

1. **微信评论功能的测试用例(功能、安全、网络、ui、性能、安全、弱网、易用性、兼容、中断）**

功能测试

1.点击发表评论能否正常弹出输入框，输入框内是否显示默认文字；

2.正常评论的输入内容限制（中文，英文，数字，字符等），能否输入特殊字符，表情，图片，语音评

论等；输入长度限制，支持的最短长度和最长长度，输入空；

3.是否对输入评论做过滤前后空格处理；

4.发表评论后对方及两人的共同好友可见，非好友不可见；且对方的朋友圈信息+1；

5.每个人发表评论以及该动态所允许评论个数有没有最大限制；

6.重复点击会不会发表重复的评论；

7.发表过的评论是否支持删除，本人可删除还是发表者可删除；

8.发表者是否可以与评论的好友正常回复消息，二人的共同好友是否可以回复消息；

9.评论之后退出该页面，再次进入朋友圈刷新评论消息是否还存在；

10.发表过的评论是否支持复制；

11.多条评论是否按照发表时间先后顺序显示还是按照最新回复时间显示；

12.评论能否显示评论人的昵称，若能显示是否正确 ；

13.评论后能否实时显示；

安全测试

1.是否对输入评论做敏感词汇处理；

2.被拉黑或者删除的好友不能发表评论；

3.未登录情况下能否访问发表评论接口；

4.用非好友的cookie信息，拿到好友的url，能否发表评论；

网络测试

1.wifi，热点，4g，5g；

2.不同运营商网络测试；

3.写评论或者点击发表时断网，是否提示友好；

4.网速较低时，是否有提示及响应时间是否可以接受；

5.网速导致的发送失败是否有重传机制；

中断测试

1.输入内容后返回，下次再点击评论，输入内容是否保留，以及这个下次有没有限制比如5分钟内？

2.切换到其他app再回到微信，会不会继续显示刚才的页面；

3.退出进程，再次打开微信会不会继续显示刚才的页面；

4.发表评论时有电话或者短信；

兼容性测试

1.不同机型，不同版本，不同客户端是否都支持评论；

易用性测试

1.页面布局是否与原设计效果图一致；（如输入框有内容，发送按钮高亮；内容清空，发送按钮置灰）

2.从测试和用户角度页面布局是否美观合理，操作是否简便，提示是否易于理解；

性能测试

1.大量用户并发评论时，该接口的响应时间是否符合用户期待，tps等性能参数是否合理；

1. **支付宝付款码的设计测试用例(功能、安全、网络、ui、性能、安全、弱网、易用性、兼容、中断）**
2. **黑盒测试的方法:**

边界值、等价类、错误推测法、场景法、因果图法、判定表组成法、正交实验设计

**8、白盒测试的方法：**

主要是检查程序的内部结构、逻辑、循环和路径。常用测试用例设计方法有：

逻辑覆盖：以程序的内部逻辑结构为基础，分为语句覆盖、判定覆盖、条件覆盖、判定-条件覆盖、

条件组合覆盖等。

基本路径测试：在程序控制流程的基础上，分析控制构造的环路复杂性，导出基本可执行路径集合，

从而设计测试用例。

1. **软件测试的流程**

(1)分析测试需求

(2)制订测试计划

(3)设计测试用例

(4)执行测试

(5)编写测试报告

1. **登录功能怎么设计测试用例？(功能、安全、网络、ui、性能、安全、弱网、易用性、兼容、中断）**

11、网上银行转账是怎么测的，设计一下测试用例。

12、给你一个网站，你应该如何测试？

13、一个有广告的纸杯子，请设计测试用例？

14、想象一个登录框，包括ID、密码、登录、取消，记住密码（复选框），尽可能的写出你想到的测试点？

15、测试淘宝购物车的测试案例

16、设计一下抖音上下滑动视频的测试用例

17、设计抖音app用户登录测试

18、中断测试

19、测试在一个场景下，推荐滤镜

20、朋友圈点赞评论测试用例

**三、情景问题**

1、如果抖音一个地区的许多用户无法下载视频的原因。

网络运营商服务异常、cdn配置、

1. 如果用户收到了银行短信提示已经扣款成功了，但是商家没有收到钱，你觉得会是什么问题。

用户付错账户、网络延迟、商家账户异常、商家看错账户

1. 输入一个url，请问发生了什么?

4、如何实现一个用户登录功能

5、高并发下减少事务带来的性能消耗？

6、如果一个API接口出现一个不稳定出现的bug，如何去确定？

7、如果提交bug，开发告诉你不存在，如何处理？

8、访问页面加载缓慢的原因以及如何解决

9、如何你和你的领导意见不一致，你会怎么做

10、如果图片上传失败，分析原因

11、针对评论功能，你如何设计接口，主要回答需要传递的参数有哪些？

12、app页面白屏了什么原因

13、全链路压测中，找到了某一个服务器CPU负载率100%，磁盘和内存使用率正常，请问你会怎么去分析可能的原因

14、设计买火车票的系统，输入出发站和到达站，返回所有可能的结果

15、添加购物车请求后发生了什么

淘宝用户支付失败的原因

**四、Python基础**

1、python的内存池机制：

python的内存机制是金字塔形。-1，-2层主要有操作系统进行操作，第0层是C中的malloc，free等内存分配和释放函数进行操作；第1层和第2层是内存池，有Python的接口函数PyMem\_Malloc函数实现，当对象小于256K时有该层直接分配内存；第3层是最上层，也就是我们对Python对象的直接操作。python又分为大内存和小内存，大小以256字节为界限，对于大内存使用Malloc进行分配，而对于小内存则使用内存池进行分配。Python的内存池又分为4个层次：Block、Pool、Arean、usedpool。

经由内存池登记的内存到最后还是会回收到内存池,并不会调用 C 的 free 释放掉，以便下次使用。对于简单的Python对象，例如数值、字符串，元组（tuple不允许被更改)采用的是复制的方式(深拷贝?)，也就是说当将另一个变量B赋值给变量A时，虽然A和B的内存空间仍然相同，但当A的值发生变化时，会重新给A分配空间，A和B的地址变得不再相同。

1. 浅拷贝和深拷贝的区别：

在python中，对象赋值实际上是对象的引用。当创建一个对象，然后把它赋给另一个变量的时候，python并没有拷贝这个对象，而只是拷贝了这个对象的引用。

（1）直接赋值,默认浅拷贝传递对象的引用而已,原始列表改变，被赋值的b也会做相同的改变

（2）copy浅拷贝，没有拷贝子对象，所以原始数据改变，子对象会改变

（3）深拷贝，包含对象里面的自对象的拷贝，所以原始对象的改变不会造成深拷贝里任何子元素的改变。

深拷贝:递归拷贝父对象和子对象,新开辟一块内存空间.新对象的修改不会影响旧对象

浅拷贝:只拷贝父对象,两者的内存地址一样,新对象的内容操作会影响旧对象的内容

1. python数组和列表的区别。

python没有数组，只有元组(tuple)和列表(list)。元组与列表最大的不同在于，元组一旦创建便不可改变；因此不像列表，元组不能够在末尾追加(append)元素，弹出(pop)元素；只能对元组中的元素进行索引t[0]，不能对其中的元组进行赋值t[0]=8。使用元组的好处在于对元组进行操作更为高效，适合存放一组常量。

1. python你常用哪些包，标准库：os、json、re、date、sys、
2. 面向过程和面向对象的语言区别

1、面向过程：是一种以过程为中心的编程思想。都是以什么正在发生为主要目标进行编程。就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用就可以了。

1. 面向对象：是一类以对象作为基本程序结构单位的程序设计语言，指用于描述的设计是以对象为核

心，而对象是程序运行时刻的基本成分。识认性，系统中的基本构件可识认为一组可识别的离散对象，对象具有唯一的静态类型和多个可能的动态类型，在基本层次关系的不同类中共享数据和操作。

1. 反射
2. python的数据结构：字符串、列表、字典、元祖、集合
3. dict和list的底层实现：

ist和tuple其实是用链表顺序存储的，也就是前一个元素中存储了下一个元素的位置，这样只要找到第一个元素的位置就可以顺藤摸瓜找到所有元素的位置，所以list的名字其实就是个指针，指向list的第一个元素的位置。

dict则为了快速查找使用了一种特别的方法，哈希表。哈希表采用哈希函数从key计算得到一个数字（哈希函数有个特点：对于不同的key，有很大的概率得到的哈希值也不同），然后直接把value存储到这个数字所对应的地址上。

1. python装饰器，作用，用法。

装饰器的作用就是用一个新函数封装旧函数（是旧函数代码不变的情况下增加功能）然后会返回一个新函数，新函数就叫做装饰器，一般为了简化装饰器会用语法糖@新函数来简化。

1. 编译型语言和解释型语言，解释型语言的优点。

编译型语言优点：运行速度快，代码效率高，编译后的程序不可修改，保密性较好。缺点：代码需要经过编译方可运行，可移植性差，只能在兼容的操作系统上运行。

解释型语言优点：可移植性较好，只要有解释环境，可在不同的操作系统上运行。缺点：运行需要解释环境，运行起来比编译的要慢，占用资源也要多一些，代码效率低，代码修改后就可运行，不需要编译过程。

1. 垃圾回收机制：

垃圾回收主要以引用计数为主，分代回收为辅。引用计数法的原理是每个对象维护一个ob\_ref，用来记录当前对象被引用的次数，也就是来追踪到底有多少引用指向了这个对象。为了解决维护引用计数消耗资源、无法解决循环引用的问题两个致命弱点，Python又引入了以下两种GC机制：标记-清除、检测循环引用。

1. GIL锁：

全局解释器锁，为了解决多线程修改同一块数据。python的线程是调用操作系统的源生线程，启动时就是调用C语言的C源生接口，python调用C语言接口的线程去执行任务时，必须上下文对应关系传给CPU处理。在Cpython解释器中执行每一个python线程，都会先锁住自己，以阻止别的线程执行。每一个线程在执行时，都会锁住GIL，阻止其他线程执行，每一个线程执行完毕，会释放GIL，允许其他线程开始利用资源。

1. is和==

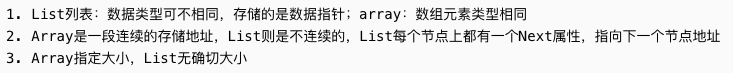
在讲is和==这两种运算符区别之前，首先要知道Python中对象包含的三个基本要素，分别是：id(身份标识)、type(数据类型)和value(值)。==是python标准操作符中的比较操作符，用来比较判断两个对象的value(值)是否相等。is也被叫做同一性运算符，这个运算符比较判断的是对象间的唯一身份标识，也就是id是否相同。只有数值型和字符串型的情况下，a is b才为True，当a和b是tuple，list，dict或set型时，a is b为False。

1. \_\_new\_\_和\_\_init\_\_区别

【同】二者均是Python面向对象语言中的函数，\_\_new\_\_比较少用，\_\_init\_\_则用的比较多。

【异】\_\_new\_\_是在实例创建之前被调用的，因为它的任务就是创建实例然后返回该实例对象，是个静态方法。\_\_init\_\_是当实例对象创建完成后被调用的，然后设置对象属性的一些初始值，通常用在初始化一个类实例的时候。是一个实例方法。

1. list和array的区别



1. 用python写出多线程，循环打印123

import threading

def num():

print("123")

if \_\_name\_\_ == "\_\_main\_\_":

for i in range(10):

t = threading.Thread(target = num)

1. **操作系统**
2. **读写锁，不同点，应用场景**

rwlock，区分读和写，处于读操作时，可以允许多个线程同时获得读操作。但是同一时刻只能有一个线程可以获得写锁。其它获取写锁失败的线程都会进入睡眠状态，直到写锁释放时被唤醒。

注意：写锁会阻塞其它读写锁。当有一个线程获得写锁在写时，读锁也不能被其它线程获取；写优先于读，当有线程因为等待写锁而进入睡眠时，则后续读者也必须等待

适用于读取数据的频率远远大于写数据的频率的场合。

1. **线程和进程的区别是什么**

(1) 调度:进程是资源分配的最小单位，线程是CPU调度的最小单位。

(2) 并发性:不仅进程之间可以并发执行,同一个进程的多个线程之间也可并发执行。

(3) 拥有资源:进程是拥有资源的一个独立单位，线程不拥有系统资源，但可以访问隶属于进程的资源。

(4) 系统开销:在创建或撤消进程时,由于系统都要为之分配和回收资源，导致系统的开销明显大于创建或撤消线程时的开销。

1. **什么是死锁，死锁的条件。（链接：https://blog.csdn.net/hd12370/article/details/82814348）**

所谓死锁，是指多个进程在运行过程中因争夺资源而造成的一种僵局，当进程处于这种僵持状态时，若无外力作用，它们都将无法再向前推进。原因：竞争资源、进程间推进顺序非法。产生死锁的4个必要条件：

(1)互斥条件：进程要求对所分配的资源进行排它性控制，即在一段时间内某资源仅为一进程所占用。

(2)请求和保持条件：当进程因请求资源而阻塞时，对已获得的资源保持不放。

(3)不剥夺条件：进程已获得的资源在未使用完之前，不能剥夺，只能在使用完时由自己释放。

(4)环路等待条件：在发生死锁时，必然存在一个进程--资源的环形链。

1. **如何预防死锁 (以确定的顺序获得锁、超时放弃)**

(1)资源一次性分配：一次性分配所有资源，这样就不会再有请求了：（破坏请求条件）

(2)只要有一个资源得不到分配，也不给这个进程分配其他的资源：（破坏请保持条件）

(3)可剥夺资源：当某进程获得了部分资源，但得不到其它资源，则释放已占有的资源（破坏不可剥夺条件）

(4)资源有序分配法：系统给每类资源赋予一个编号，每一个进程按编号递增的顺序请求资源，释放则相反

（破坏环路等待条件）

1. **避免死锁:**

* 预防死锁的几种策略，会严重地损害系统性能。因此在避免死锁时，要施加较弱的限制，从而获得较满意的系统性能。由于在避免死锁的策略中，允许进程动态地申请资源。因而，系统在进行资源分配之前预先计算资源分配的安全性。若此次分配不会导致系统进入不安全的状态，则将资源分配给进程；否则，进程等待。其中最具有代表性的避免死锁算法是银行家算法。
* **银行家算法：**首先需要定义状态和安全状态的概念。系统的状态是当前给进程分配的资源情况。因此，状态包含两个向量Resource（系统中每种资源的总量）和Available（未分配给进程的每种资源的总量）及两个矩阵Claim（表示进程对资源的需求）和Allocation（表示当前分配给进程的资源）。安全状态是指至少有一个资源分配序列不会导致死锁。当进程请求一组资源时，假设同意该请求，从而改变了系统的状态，然后确定其结果是否还处于安全状态。如果是，同意这个请求；如果不是，阻塞该进程知道同意该请求后系统状态仍然是安全的。

1. **检测死锁**

首先为每个进程和每个资源指定一个唯一的号码；然后建立资源分配表和进程等待表。

(1)jstack是java虚拟机自带的一种堆栈跟踪工具。

(2)Jconsole是JDK自带的监控工具，在JDK/bin目录下可以找到。

1. **解除死锁**

当发现有进程死锁后，便应立即把它从死锁状态中解脱出来，常采用的方法有：

* 剥夺资源：从其它进程剥夺足够数量的资源给死锁进程，以解除死锁状态；
* 撤消进程：可以直接撤消死锁进程或撤消代价最小的进程，直至有足够的资源可用，死锁状态消除为止；所谓代价是指优先级、运行代价、进程的重要性和价值等。

1. **堆和栈的区别以及存储模式有什么区别。(属于同一进程的线程，堆是共享的，栈是私有的。)**

栈和堆都是用来从底层操作系统中获取内存的。栈(Stack)是为执行线程留出的内存空间。堆（heap）是为动态分配预留的内存空间。栈由系统自动分配，速度较快。但程序员是无法控制的。堆是由new分配的内存，一般速度比较慢，而且容易产生内存碎片,不过用起来最方便。

在存储内容上，栈在函数调用时，在大多数的C编译器中，参数是由右往左入栈的，然后是函数中的局部变量。注意静态变量是不入栈的。当本次函数调用结束后，局部变量先出栈，然后是参数，最后栈顶指针指向函数的返回地址，也就是主函数中的下一条指令的地址，程序由该点继续运行。堆：一般是在堆的头部用一个字节存放堆的大小，堆中的具体内容由程序员安排。

1. **内存泄漏和内存溢出**

内存泄漏是指程序在申请内存后，无法释放已申请的内存空间。内存溢出是指程序申请内存时，没有足够的内存供申请者使用；或者说提供一块存储int数据的存储空间，但存储了long数据，则结果是内存不够用，报错OOM。内存泄漏的堆积最终会导致内存溢出。

1. **多线程和多进程，以及应用场景**
2. **进程通信方式**

(1)管道pipe：管道是一种半双工的通信方式，数据只能单向流动，而且只能在具有亲缘关系的进程间使用。进程的亲缘关系通常是指父子进程关系。

(2)命名管道FIFO：有名管道也是半双工的通信方式，但是它允许无亲缘关系进程间的通信。

(3)消息队列MessageQueue：消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。消息队列克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点。

(4)共享存储SharedMemory：共享内存就是映射一段能被其他进程所访问的内存，这段共享内存由一个进程创建，但多个进程都可以访问。共享内存是最快的 IPC 方式，它是针对其他进程间通信方式运行效率低而专门设计的。它往往与其他通信机制，如信号量，配合使用，来实现进程间的同步和通信。

(5)信号量Semaphore：信号量是一个计数器，可以用来控制多个进程对共享资源的访问。它常作为一种锁机制，防止某进程正在访问共享资源时，其他进程也访问该资源。因此，主要作为进程间以及同一进程内不同线程之间的同步手段。

(6)套接字Socket：套解口也是一种进程间通信机制，与其他通信机制不同的是，它可用于不同及其间的进程通信。

(7)信号 ( sinal ) ： 信号是一种比较复杂的通信方式，用于通知接收进程某个事件已经发生。

1. **多线程如何实现的。**
2. 计算机内存管理的方式
3. 进程调度方法
4. 什么是Linux用户态和内核态
5. 进程的状态，进程状态就绪和等待状态的区别是什么
6. 虚拟内存是干嘛的
7. 线程池
8. 线程安全的实现方式
9. 进程和线程的上下文切换
10. 协程

**六、计算机网络**

1、TCP三次握手，四次挥手的过程

2、为什么客户端要在TIME\_WAIT状态等待一段时间

3、CLOSING状态

4、如果server端没有收到第三次ack，但是收到了client端发送的数据，server端会怎么处理

5、讲一讲get和post

6、http状态码

7、TCP和UDP

8、一次HTTP请求的过程（输入URL到返回的全过程）

9、http和https

10、无状态和无连接

11、OSI七层模型，各层有哪些协议

12、HTTP协议 请求报文结构

13、TCP如何保证可靠传输，丢包怎么办

14、讲一下滑动窗口

15、TCP三次握手中SYN和ACK包有什么不同，包含什么

16、UDP想要可靠怎么实现 RUDP

17、半连接攻击

18、TCP粘包和拆包

19、cookie和session是什么，为什么要用cookie和session

20、DNS协议和作用

21、DNS的查询方式

22、ARP协议作用、工作方式

23、以太网数据包的大小最大为多少

24、三次握手过程中是否存在安全问题？描述一下存在什么样的安全问题？针对这样的安全问题如何防御？

25、web攻击，CSRF攻击

26、TCP如何进行拥塞控制？拥塞控制如何判断发生拥塞？

27、TCP快重传如何判断丢失？

28、https证书在哪存放

29、长连接和短连接以及他们分别适用的场景

七、数据库

sql语句

取一个月内的id分组

取一个年级中每个班级年龄最小的同学名字

成绩表输出前三名的成绩，后三名呢

拷贝A表的数据到B表

查询每门科目都大于80分的学生名字

筛选出每个小时的记录

考察where

考察聚合函数

where和having的区别

一个数据库sql查询重复个数

在学生成绩表中查询平均分不及格（低于60）的学生id及平均分

查询总成绩最高的人的姓名

user表中在一个时间段内登陆三次的用户id limit的用法

数据库引擎及他们之间的区别 InnoDB MyISAM MEMORY：

mysql的日志怎么查询

（多次）查询速度慢的原因，如何解决

数据库的事务是什么？怎么使用

主键、外键、索引的各自的含义以及区别

事务的特性（ACID），讲一下每个特性的意思

索引的类型

什么是聚簇索引、什么是非聚簇索引

索引的数据结构是什么

前缀索引

MySQL/MongoDb 端口号

数据库有几种表之间的连接形式（左连接，右连接，内连接，完全连接）

说说redis中的数据结构和应用场景

三大范式

数据库隔离等级

为什么要用b+树索引而不用hash

八、Linux

常见命令 查看进程ps 查看所有进程长格式ps -ef 打印当前文件夹文件 ls -al a所有文件 l详细信息 软链接 ln -s a target 硬链接 ln 创建目录 mkdir 修改权限 chmod 查看文件： 移动文件mv 也可以重命名 复制文件cp 删除文件 rm -r 递归删除 -f删除只读文件 通配符？单个字符，\*多个字符，[a-z]字符集 grep 正则搜索 显示行号-n 查询个数-c 反选-v 查看指定进程 ps -ef | grep pid 查看命令历史 history 查看磁盘空间df -hl 查看文件大小du 查找文件 find [目录] 以名字查找-name “h\*”以h开头 终止一个进程 kill -9 pid

如何防止新人误操作rm -rf

找进程号为199的进程 ls -ef|grep 199

某文件中含关键词的行数 grep -c "key" ./file

该目录下以.log结尾的文件中包含关键词a但不包含关键词b的行数 grep "a" ./file | grep -cv "b"

**九、智力题**

一根金条，需要一周每天都发工资，切割两次，请问怎么切割 把金条切分为1/7, 2/7, 4/7，每天找零

100个足球队两两比赛角逐出一支冠军队伍，请问至少比赛几次 每场比赛淘汰一只队伍，所以需要99场比赛

25匹马5个赛道，求前三名

100个豆子，五个囚犯，拿的豆子最多或者最少都要被处死，求谁的生存概率大一点