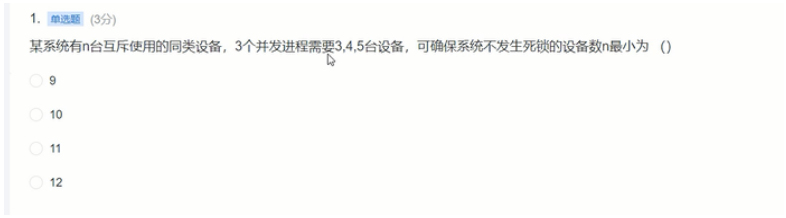
2021-8-2 shopee笔试

- 题型---15+5+2
  - 1

试题组	数量	总分
单选题	15	45
多选题	5	20
编程题	2	35

- 具体内容
  - 1.死锁，互斥
    - 截图
      - 1



    - 参考资料
      - 进程（线程）死锁详解
    - 解析
      - 多进程，多线程的并发执行虽然提升了系统资源的利用率，提高了系统的性能，但是并发执行也带来了新的问题-----死锁。
      - 死锁产生的主要原因：
        - 系统的资源不足。
        - 进程（线程）推进的顺序不对。
        - 资源的分配不当。
        - ----------
        - 当系统的资源很充沛的时候，每个进程都可以申请到想要的资源，那么出现死锁的概率就很低，线程的调度顺序和速度不同，也会导致死锁问题。
      - 死锁产生的四个必要条件：---产生死锁，必有以下四个情况

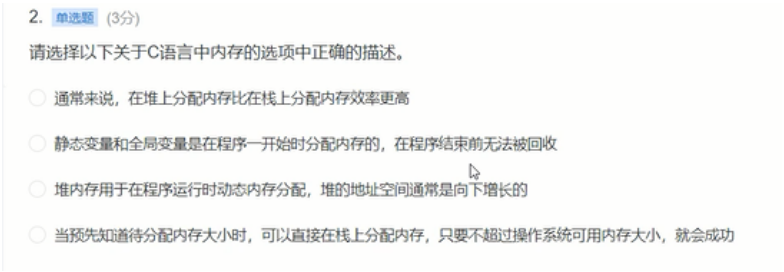
        - 互斥条件：进程（线程）申请的资源在一段时间中只能被一个进程（线程）使用。
        - 请求与等待条件：进程（线程）已经拥有了一个资源，但是又申请新的资源，拥有的资源保持不变。
        - 不可剥夺条件：在一个进程（线程）没有用完，主动释放资源的时候，不能被抢占。
        - 循环等待条件：多个进程（线程）之间存在资源循环链。
    - 答案
        - 1

在极端状态下：
进程1(3台)：申请到2台，无法工作；
进程2(4台)：申请到3台，无法工作；
进程3(5台)：申请到4台，无法工作；
申请总数：2+3+4=9，此时若只有9台，3个进程持续申请且申请不到，造成死锁。

所以，在此时我们可以让设备要使用的资源，再多上1台，就可以使三个进程中的某一个顺利执行，从而不发生死锁。

因此，可确保系统不发生死锁的设备数 n 最小为10.

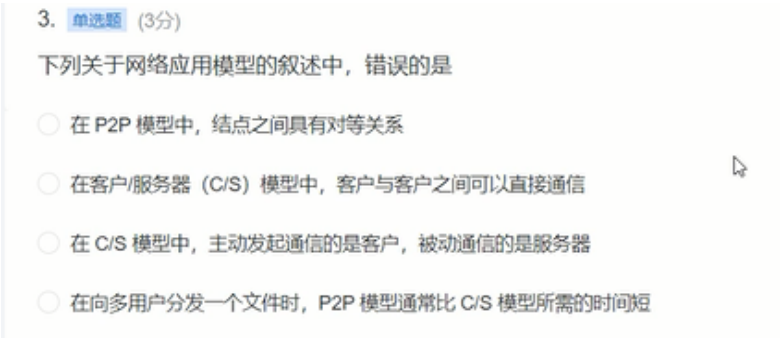
- 2.C语言内存
    - 截图
        - 2



    - 参考资料
        - 浅析栈区和堆区内存分配的区别
    - 解析
        - 1

- 全局区
  - 全局变量和静态变量的存储是放在一块的，初始化的全局变量和静态变量在一块区域， 未初始化的全局变量和未初始化的静态变量在相邻的另一块区域。 程序结束后由系统释放
- 常量区
  - 常量字符串就是放在这里的。 程序结束后由系统释放
- 程序代码区
  - 存放函数体的二进制代码。
- 分配内存效率比较
  - 栈由系统自动分配，速度较快。但程序员是无法控制的。
    - 存放函数的参数值，局部变量的值
    - 内存的分配是连续的---数组
  - 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.
    - 内存中的分布不是连续的--链表
    - 不同区域的内存块通过指针链接起来
    - new出来的内存空间存放在堆中，不受作用域管理，不会被系统自动回收，只有在使用delete删除或者整个程序结束后才会释放内存。
  - ---
  - 另外，在WINDOWS下，最好的方式是用VirtualAlloc分配内存，他不是在堆，也不是在栈是直接在**进程的地址空间**中保留一快内存，虽然用起来最不方便。但是速度快，也最灵活。
- 地址空间
  - 栈
    - 向低地址拓展的数据结构，连续
    - 这句话的意思是栈顶的地址和栈的最大容量是系统预先规定好的，在WINDOWS下，栈的大小是2M（也有的说是1M，总之是一个编译时就确定的常数），如果申请的空间超过栈的剩余空间时，将提示overflow。因此，能从栈获得的空间较小。
  - 堆
    - 向高地址拓展的数据结构，不连续

        - 这是由于系统是用链表来存储的空闲内存地址的，自然是不连续的，而链表的遍历方向是由低地址向高地址
    - 答案
        - 4
- 3. p2p和c/s
    - 截图
        - 3



    - 解析
        - b

下列关于网络应用模型的叙述中，错误的是

A. 在 P2P 模型中，结点之间具有对等关系

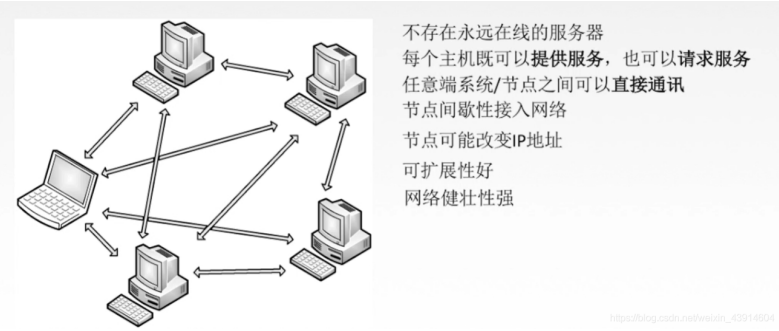
B. 在客户/服务器（C/S）模型中，客户与客户之间可以直接通信

C. 在 C/S 模型中，主动发起通信的是客户，被动通信的是服务器

D. 在向多用户分发一个文件时，P2P 模型通常比 C/S 模型所需的时间短

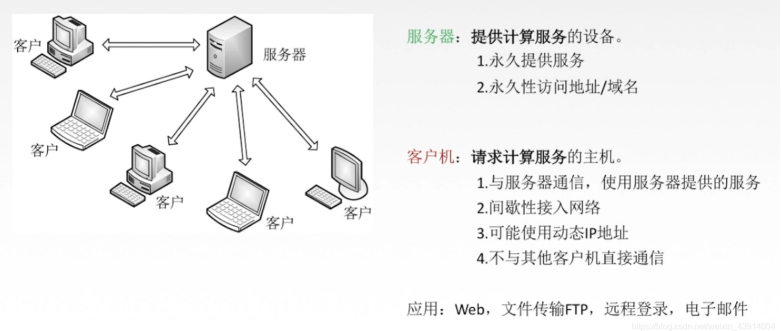
正确答案：B

        - 在 P2P 模型中，结点之间具有对等关系
        - ~~在客户/服务器（C/S）模型中，客户与客户之间可以直接通信~~
        - 在 C/S 模型中，主动发起通信的是客户，被动通信的是服务器
        - 在向多用户分发一个文件时，P2P 模型通常比 C/S 模型所需的时间短
        - p2p
            - 1

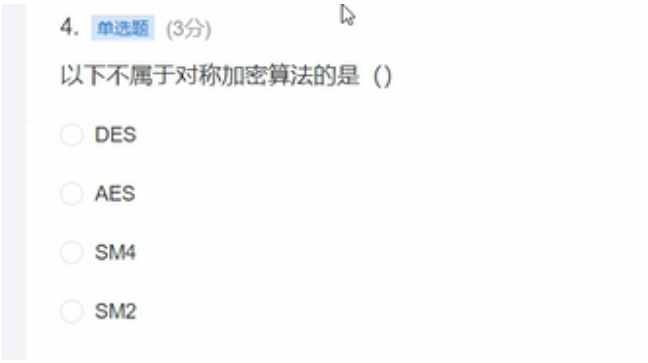


            - P2P模型也有缺点。
                - 在获取服务的同时,还要给其他结点提供服务，因此会占用较多的内存，影响整机速度。

- c/s
  - 1



- 4.对称加密算法
  - 截图
    - 4

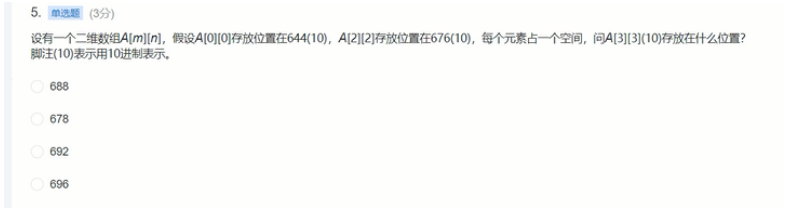


  - 参考资料
    - 国密算法概述 SM1、SM2、SM3、SM4、SM7、SM9、ZUC
  - 解析
    - AES，DES，3DES，RC
    - 基于“对称密钥”的加密算法主要有DES、3DES（TripleDES）、AES、RC2、RC4、RC5和Blowfish等。
    - 常用的对称加密算法有:**DES**、**3DES**、**DESX**、**Blowfish**、**IDEA**、**RC4**、**RC5**、**RC6**、**AES**。
    - SM4对称加密算法
      - 与DES和AES算法类似，SM4算法是一种分组密码算法。
      - 其分组长度为128bit，密钥长度也为128bit。
      - 加密算法与密钥扩展算法均采用32轮非线性迭代结构，以字（32位）为单位进行加密运算，每一次迭代运算均为一轮变换函数F。
      - SM4算法加/解密算法的结构相同，只是使用轮密钥相反，其中解密轮密钥是加密轮密钥的逆序
    - 其中SM1、SM4、SM7、祖冲之密码（ZUC）是对称算法；SM2、SM9是非对称算法；SM3是哈希算法。
    - 1

种类	算法类型	密钥长度	输入数据要求	输出数据特征
SM2	非对称加密算法	公钥64字节，私钥32字节。	长度小于（2^32-1）*32=137,438,953,440字节(大约1374亿多）。	输出长度是明文长度+96，有随机数参数，每次密文不同。
SM3	摘要算法	--	无要求。	固定长度，32字节。
SM4	对称加密算法	16字节	分组长度16字节，需要填充到16字节整数倍。有CBC和ECB两种模式，CBC需要设定初始值。	长度为16字节整数倍。

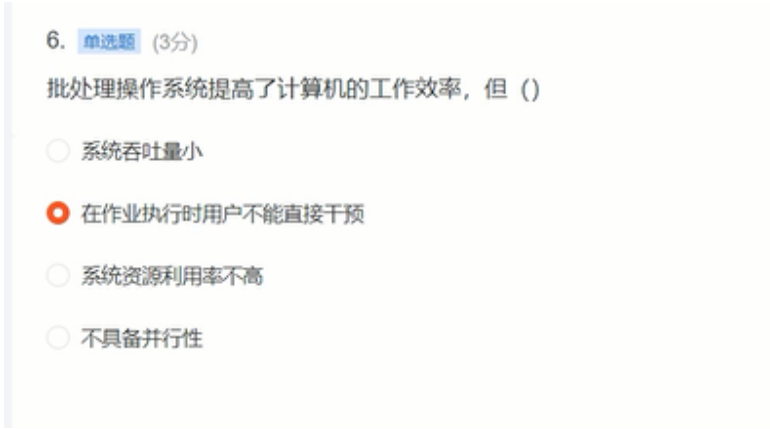
- DES
  - DES算法全称为Data Encryption Standard，即数据加密算法，它是IBM公司于1975年研究成功并公开发表的。DES算法的入口参数有三个：Key、Data、Mode。其中Key为8个字节共64位，是DES算法的工作密钥；Data也为8个字节64位，是要被加密或被解密的数据；Mode为DES的工作方式,有两种：加密或解密。

- 5.二维数组存储方式
  - 截图
    - 5



  - 解析
    - 22-00 = 22，2x+2=676-644=32,x=15
    - 33-22=11 ,16 16+676=692
    - 33-00=33，48，48+644 =692

- 6.批处理操作系统提高了计算机的工作效率，但
  - 截图
    - 6



  - 解析
    - 批处理系统，又名批处理操作系统。批处理是指用户将一批作业提交给操作系统后就不再干预，由操作系统控制它们自动运行。这种采用批量处理作业技术的操作系统称为批处理操作系统。批处理操作系统分为单道批处理系统

和多道批处理系统。批处理操作系统不具有交互性，它是为了提高CPU的利用率而提出的一种操作系统。

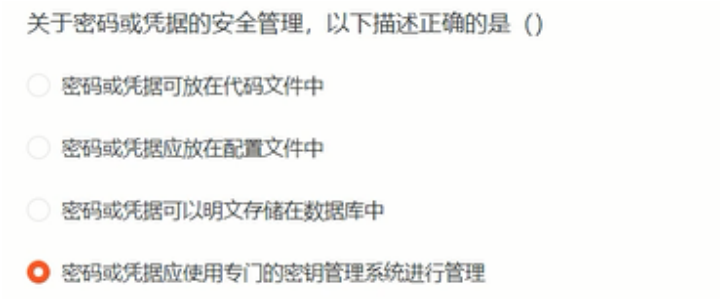
- 批处理系统的特点
  - 多道：在内存中同时存放多个作业，一个时刻只有一个作业运行，这些作业共享CPU和外部设备等资源。
  - 成批：用户和他的作业之间没有交互性。用户自己不能干预自己的作业的运行，发现作业错误不能及时改正。
  - 批处理系统的目的是提高系统吞吐量和资源的利用率。
  - -------
  - 多道处理系统的优点是由于系统资源为多个作业所共享，其工作方式是作业之间自动调度执行。并在运行过程中用户不干预自己的作业，从而大大提高了系统资源的利用率和作业吞吐量。其缺点是无交互性，用户一旦提交作业就失去了对其运行的控制能力，而且是批处理的，作业周转时间长，用户使用不方便。

- 7.关于XSS漏洞的分类
  - 截图
    - 7

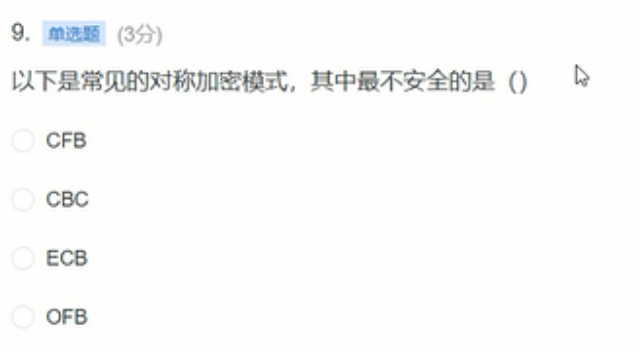
  - 解析
    - 分类依据
      - 1

- 分类标志有三种
  - XSS 攻击按**存在周期**或者是否把攻击数据存进服务器端，攻击行为是否伴随着攻击数据一直存在，可分为 非持久型 XSS 攻击 和 持久型 XSS 攻击。
  - XSS 攻击按**攻击方式**又可分为 反射型 XSS、DOM 型 XSS、存储型 XSS，其中 反射型 XSS 和 DOM 型 XSS 算是 非持久型 XSS 攻击，而 存储型 XSS 算是 持久型 XSS 攻击。
  - 按**攻击介质**分类
    - JSXSS
    - FlashXSS

- 8.密码和凭据 安全管理
  - 截图
    - 8



  - 解析
    - 密码或凭据应使用专门的密钥管理系统进行管理
- 9.常见的对称加密模式，最不安全的
  - 截图
    - 9



  - 参考资料
    - 对称加密算法常用的五种分组模式（ECB/CBC/CFB/OFB/CTR）
  - 解析
    - CFB
    - CBC
    - ECB
    - OFB
    - Q:为什么需要分组模式?

- A:明文的长度不固定，而分组密码只能处理特定长度的一块数据，这就需要对分组密码的算法进行迭代，以便将一段很长的明文全部加密，而迭代的方法就是分组的模式。
- 看缺点，ecb最不安全

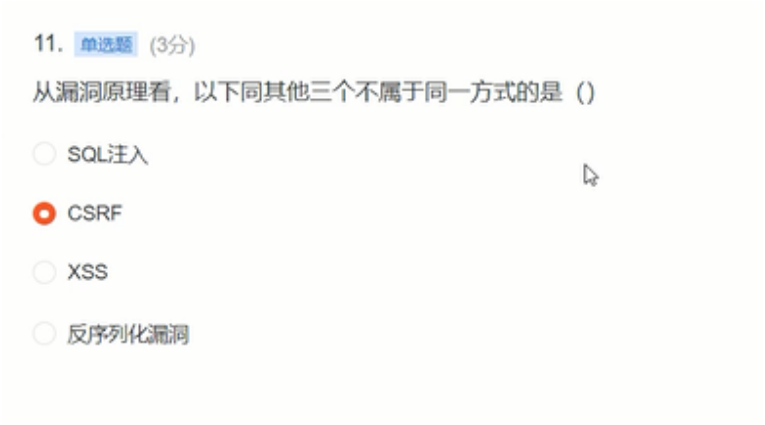
模式	名称	优点	缺点	备注
ECB 模式	Electronic CodeBook 电子密码本模式	• 简单 • 快速 • 支持并行计算（加密、解密）	• 明文中的重复排列会反映在密文中 • 通过删除、替换密文分组可以对明文进行操作 • 对包含某些比特错误的密文进行解密时，对应的分组会出错 • 不能抵御重放攻击	不应使用
CBC 模式	Cipher Block Chaining 密文分组链接模式	• 明文的重复排列不会反映在密文中 • 支持并行计算（仅解密） • 能够解密任意密文分组	• 对包含某些错误比特的密文进行解密时，第一个分组的全部比特以及后一个分组的相应比特会出错 • 加密不支持并行计算	推荐使用
CFB 模式	Cipher-FeedBack 密文反馈模式	• 不需要填充（padding） • 支持并行计算（仅解密） • 能够解密任意密文分组	• 加密不支持并行计算 • 对包含某些错误比特的密文进行解密时，第一个分组的全部比特以及后一个分组的相应比特会出错 • 不能抵御重放攻击	• 现在已不使用 • 推荐用 CTR 模式代替
OFB 模式	Output-FeedBack 输出反馈模式	• 不需要填充（padding） • 可事先进行加密、解密的准备 • 加密、解密使用相同结构 • 对包含某些错误比特的密文进行解密时，只有明文中相对应的比特会出错	• 不支持并行计算 • 主动攻击者反转密文分组中的某些比特时，明文分组中相对应的比特也会被反转	推荐用 CTR 模式代替
CTR 模式	CounTeR 计数器模式	• 不需要填充（padding） • 可事先进行加密、解密的准备 • 加密、解密使用相同结构 • 对包含某些错误比特的密文进行解密时，只有明文中相对应的比特会出错 • 支持并行计算（加密、解密）	主动攻击者反转密文分组中的某些比特时，明文分组中相对应的比特也会被反转	推荐使用

- 以上五种分组模式中，ECB模式很容易被破解，如今已经很少再使用，其余四种分组模式各有千秋。但极力推荐CBC模式和CTR模式，尤其是CTR模式，不需要填充，代码实现起来很方便。而且加密和解密的方法是一样的，并且可以实现并发分组，效率高，安全性也有保障
- Q:何时需要填充，何时不需要填充?
- A:观察分组模式的图示可以看出，加密后再进行亦或操作的不需要填充，而先进行亦或操作再加密的则需要填充，这是因为亦或操作需要两个相同长度的数据，一一对比计算！

- 10. SSRF
  - 截图
    - 10

  - 解析
    - SSRF
      - Server-Side Request Forger
      - 服务端请求伪造

        - 利用存在缺陷的web应用作为代理攻击远程和本地的服务器
- 11. 漏洞原理
    - 截图
        - 11



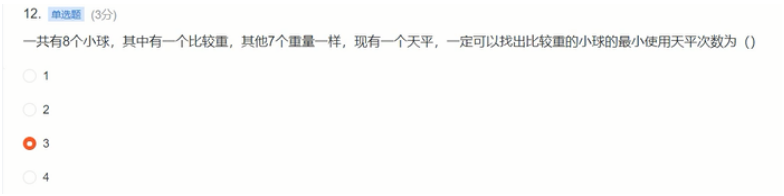
    - 解析
        - 注入漏洞
            - SQL注入
            - XSS
                - **利用的是用户对指定网站的信任**
            - PHP反序列化漏洞也叫PHP对象注入
            - CRLF注入---HTTP响应头注入漏洞
                - 在《HTTP | HTTP报文》一文中，我们介绍了HTTP报文的结构：状态行和首部中的每行以CRLF结束，首部与主体之间由一空行分隔。或者理解为首部最后一个字段有两个CRLF，首部和主体由两个CRLF分隔。
                - CRLF注入漏洞，是因为Web应用没有对用户输入做严格验证，导致攻击者可以输入一些恶意字符。攻击者一旦向请求行或首部中的字段注入恶意的CRLF，就能注入一些首部字段或报文主体，并在响应中输出，所以又称为HTTP响应拆分漏洞（HTTP Response Splitting）。
                - CRLF 指的是**回车符**(CR，ASCII 13，\r，%0d) 和**换行符**(LF，ASCII 10，\n，%0a)。
                -
            - CSV注入漏洞
                - 浅谈CSV注入漏洞
                - **CSV公式注入(CSV Injection)**是一种会造成巨大影响的攻击向量。攻击包含向恶意的EXCEL公式中注入可以输出或以CSV文件读取的参数。当在Excel中打开CSV文件时，文件会从CSV描述转变为原始的Excel格式，包括Excel提供的所有动态功能。在这个过程中，CSV中

的所有Excel公式都会执行。当该函数有合法意图时，很易被滥用并允许恶意代码执行。

- 命令注入漏洞
  - 即 Command Injection。是指通过提交恶意构造的参数破坏命令语句结构，从而达到执行恶意命令的目的。
  - 在Web应用中，有时候会用到一些命令执行的函数，如php中system、exec、shell_exec等，当对用户输入的命令没有进行限制或者过滤不严导致用户可以执行任意命令时，就会造成命令执行漏洞。
- 模板注入漏洞
  - SSTI
- **SpEL表达式注入漏洞**
- XML注入漏洞
- 跳转漏洞
- os命令
- ORM注入
- LDAP和表达式语言（EL）或OGNL注入

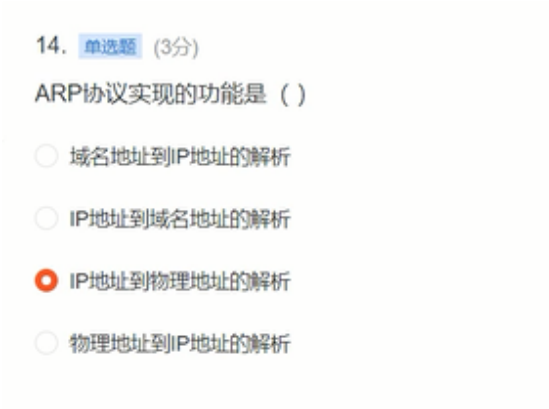
- CSRF
  - cross site　Request forgery, 跨站请求伪造
  - **利用的是网站对用户网页浏览器的信任。**
  - web中用户身份验证的一个漏洞：**简单的身份验证只能保证请求发自某个用户的浏览器，却不能保证请求本身是用户自愿发出的。**

- 12.球
  - 截图
    - 12

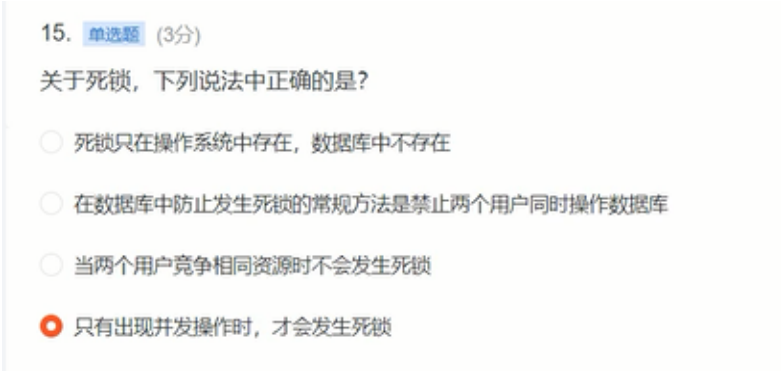


  - 解析--2次
    - 6+2
      - 3：3
        - 1：1
      - 1：1
- 13.HTTP，HTTPS
  - 截图
    - 13

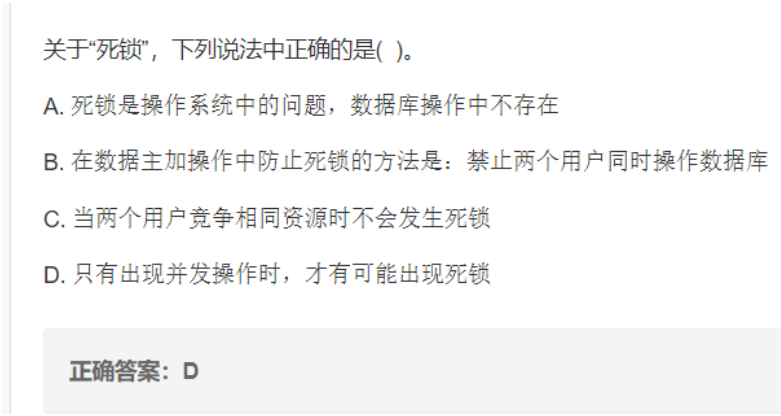
- 解析
  - HTTPS （全称：Hyper Text Transfer Protocol over SecureSocket Layer），是以安全为目标的 HTTP 通道，在HTTP的基础上通过传输加密和身份认证保证了传输过程的安全性 [1][ ]() 。HTTPS 在HTTP 的基础下加入SSL，HTTPS 的安全基础是 SSL，因此加密的详细内容就需要 SSL。 HTTPS 存在不同于 HTTP 的默认端口及一个加密/身份验证层（在 HTTP与 TCP 之间）。这个系统提供了身份验证与加密通讯方法。它被广泛用于万维网上安全敏感的通讯，例如交易支付等方面 [2] 。
  - SSL(Secure Sockets Layer 安全套接字协议),及其**继任者**传输层安全（Transport Layer Security，TLS）是为网络通信提供安全及数据完整性的一种安全协议。TLS与SSL在传输层与应用层之间对网络连接进行加密。
  - HTTPS 协议是由 HTTP 加上 TLS/SSL 协议构建的可进行加密传输、身份认证的网络协议，主要通过数字证书、加密算法、非对称密钥等技术完成互联网数据传输加密，实现互联网传输安全保护。设计目标主要有三个。
    - 数据保密性
    - 数据完整性
    - 身份校验安全性
- 14 ARP协议实现的功能
  - 截图
    - 14



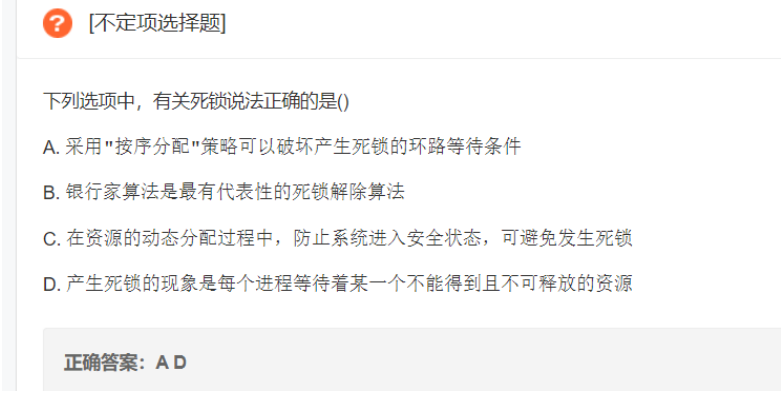
- 15.死锁
  - 截图
    - 15



- 解析
  - D



  - 死锁是指两个或者两个以上的进程在执行的过程中，因竞争共享资源而造成的相互等待的现象。
  - 死锁在操作系统中有，在数据库操作中也存在
  - 在数据库操作中防止死锁的方法是可串行化控制
  - 死锁产生的四个条件，互斥，不可剥夺，请求和保持，循环等待
  - 2

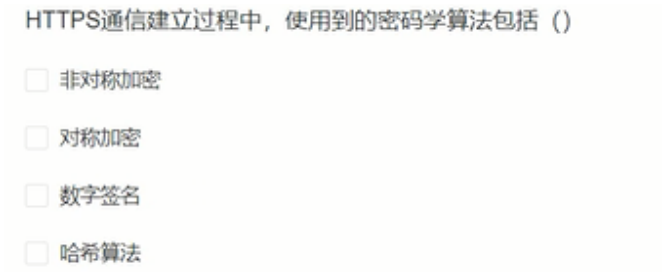


  - B、银行家算法是**避免死锁**；
  - C、在资源的动态分配过程中，防止系统进入**不安全状态**，可避免发生死锁。

- ----------多选题
- 1.短信验证码机制，安全漏洞
  - 截图
    - 1

- 参考资料
  - 手机验证码常见漏洞总结
- 解析
  - 验证码爆破
  - 短信轰炸
    - 在测试的过程中，对短信验证码接口进行重放，导致大量发送恶意短信
    - 无限制，任意下发
  - 验证码返回前端
  - 验证码不失效

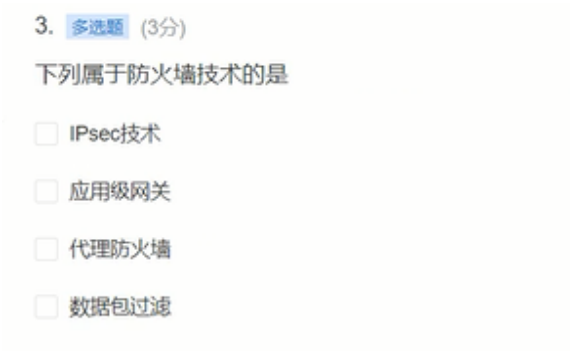
- 2.HTTPS通信建立过程中，使用到的密码学算法包括
  - 截图
    - 2



  - 解析
    - 非对称加密---发送对称加密算法和对称密钥
    - 对称加密---发送消息
    - 数字签名--验证数据的完整性，数据本身是否加密不属于数字签名的控制范围
      - 数字签名技术就是对“非对称密钥加解密”和“数字摘要“两项技术的应用，它将摘要信息用发送者的私钥加密，与原文一起传送给接收者。接收者只有用发送者的公钥才能解密被加密的摘要信息，然后用HASH函数对收到的原文产生一个摘要信息，与解密的摘要信息对比。如果相同，则说明收到的信息是完整的，在传输过程中没有被修改，否则说明信息被修改过，因此数字签名能够验证信息的完整性。
      - 数字签名的过程如下：
      - 明文 --> hash运算 --> 摘要 --> 私钥加密 --> 数字签名

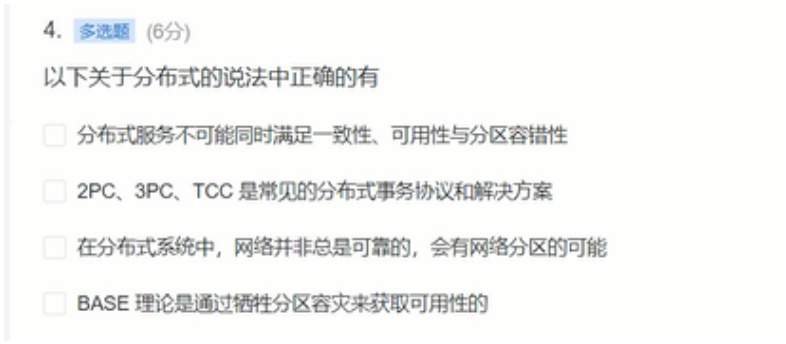
- ——————————————————
- 哈希算法--摘要算法
  - 数字摘要是采用单项Hash函数将需要加密的明文“摘要”成一串固定长度（128位）的密文，这一串密文又称为数字指纹，它有固定的长度，而且不同的明文摘要成密文，其结果总是不同的，而同样的明文其摘要必定一致。“数字摘要“是https能确保数据完整性和防篡改的根本原因。
  - 哈希算法（Hash）又称摘要算法（Digest），它的作用是：对任意一组输入数据进行计算，得到一个固定长度的输出摘要。
  - 哈希算法最重要的特点就是：
    - 相同的输入一定得到相同的输出；
      - 不同的输入大概率得到不同的输出。
  - 哈希算法的目的就是为了验证原始数据是否被篡改。

- 3.防火墙技术
  - 截图
    - 3



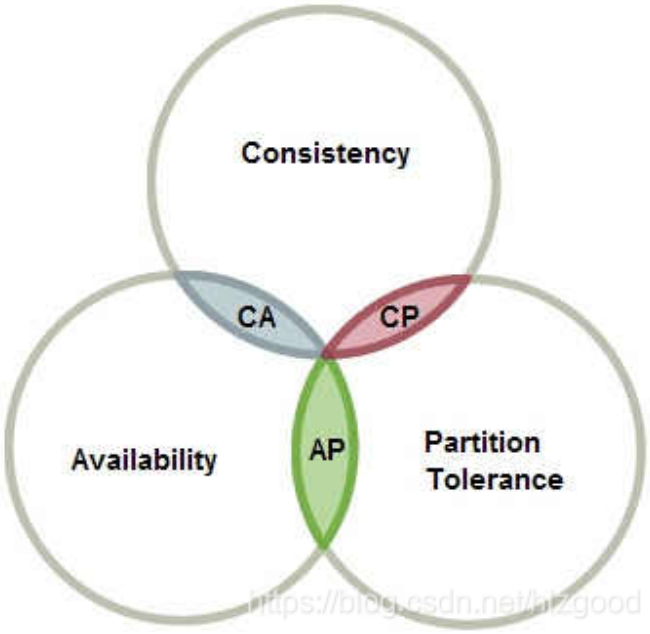
  - 参考资料
    - 网络安全：包过滤防火墙和代理防火墙（应用网关防火墙）
    - 什么是IPsec？
  - 解析-2-3-4
    - 从具体的实现技术上看，防火墙可以分为包过滤防火墙和应用网关防火墙两类
      - 包过滤防火墙----网络层和传输层--包过滤路由器
        - 无状态包过滤防火墙
        - 有状态包过滤防火墙
      - 访问控制列表（Access Control List ,ACL）由按序排列的过滤规则构成，工作在防火墙某个接口的特定方向上，以in表示流入数据包，以out表示流出数据包。
      - 应用网关防火墙
        - 应用网关防火墙以代理服务器（Proxy Server）技术为基础。代理服务器作用在应用层，在内网主机与外围主机之间进行信息交换。

        - 指采用IPSec协议来实现远程接入的一种VPN技术，IPSec全称为Internet Protocol Security，是由Internet Engineering Task Force (IETF) 定义的安全标准框架，在公网上为两个私有网络提供安全通信通道,通过加密通道保证连接的安全——在两个公共网关间提供私密数据封包服务
        - IPsec（Internet Protocol Security）是为IP网络提供安全性的协议和服务的集合，它是VPN（Virtual Private Network，虚拟专用网）中常用的一种技术。由于IP报文本身没有集成任何安全特性，IP数据包在公用网络如Internet中传输可能会面临被伪造、窃取或篡改的风险。通信双方通过IPsec建立一条IPsec隧道，IP数据包通过IPsec隧道进行加密传输，有效保证了数据在不安全的网络环境如Internet中传输的安全性。
- 4.分布式
    - 截图
        - 4

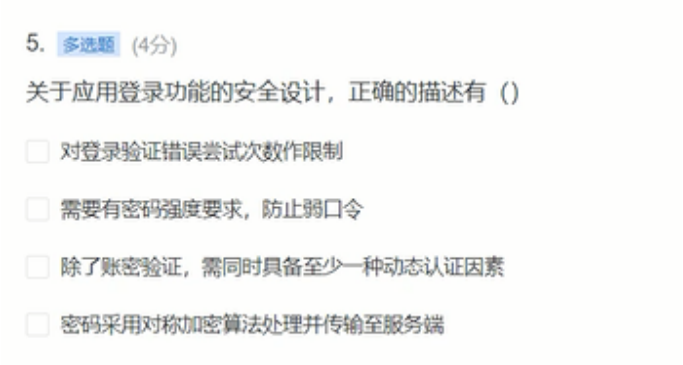


    - 参考资料
        - 面试被问分布式事务（2PC、3PC、TCC），这样解释没毛病！
        - 分布式-基础理论
    - 解析
        - 分布式服务不可能同时满足一致性、可用性与分区容错性
        - 2PC、3PC、TCC是常见的分布式事务协议和解决方案
        - 在分布式系统中，网络并非总是可靠的，会有网络分区的可能
        - BASE理论是通过牺牲分区容灾来获取可用性的
        - -------
    - CAP由以下三个指标组成：
        - C（Consistency）：一致性
            - 一致性，意思是在一个分区进行写操作后，所有分区的数据都要保持一致的读操作。
        - A（Availability）：可用性
            - 可用性，意思是只要收到请求，系统就必须给出响应。
            - 分布式系统有多个分区，即使有少数分区的服务崩溃了，还能继续提供服务，这就是服务的可用性。
        - P（Partition tolerance）：分区容错

        - 分区容错，意思是允许分区之间的通信失败。
    - 可以看出，CAP这三个指标不能同时做到。



    - CA-----这种情况在分布式系统中几乎是不存在的。首先在分布式环境下，网络分区是一个自然的事实。因为分区是必然的，所以如果舍弃P，意味着要舍弃分布式系统。那也就没有必要再讨论CAP理论了。
    - BASE理论提出通过牺牲强一致性来获得可用性，并允许数据在一段时间内是不一致的，但最终达到一致状态。
- 5.登录功能
    - 截图
        - 5



    - 解析
        - 登录次数限制
        - 强密码
        - 动态认证
        - 加密算法
            - BCrypt比MD5更安全
            - 在md5的基础上手动加盐（salt）处理
            - salt---一串随机字符串

- -------编程题
  - 1.寻找数组平衡点
    - 代码
      - 1

```
def findBalancedIndex(arr):
    for temp in range(1,len(arr)):
        # a=sum(arr[:temp])
        # print(a)
        # b=sum(arr[temp+1:])
        # print(b)
        if sum(arr[:temp])==sum(arr[temp+1:]):
            #print(temp)
            return temp
result1 = findBalancedIndex([1,2,3,4,6])
result2 = findBalancedIndex([1,2,1])
result3 = findBalancedIndex([1,2,2,2,1])
print(result1)
print(result2)
print(result3)
```

  - 2. 数的划分
    - 代码
      - 1

```
# import lib
#https://blog.csdn.net/elma_tww/article/details/86538836
# def functions
#dp[i][j]表示将整数i划分为j份的方案数
#dp[i-j],拿出j个数，分别放上1，dp[i-j][2],2表示分为两份
def divide(n,k,dp):
    for i in range(1,n+1):
        for j in range(1,k+1):
            if i>=j:#划分的分数不能超过给定的整数
                dp[i][j]=dp[i-j][j]+dp[i-1][j-1]#裂项相消
                print(i,j,dp[i][j])
    return dp[n][k]

#main
if __name__ == "__main__":
    n,k = map(int,input().split(','))
    # n = int(n)
    # k = int(k)
    dp = [[0] *(k+1) for i in range(n+1)]
    dp [0][0]=1
    #print(dp)
    result = divide(n,k,dp)
    print(result)
```