# 中国电信股份有限公司云计算分公司笔试题

## www页面访问的大致过程如下:

用户在浏览器中输入要访问的www页面的 (1) 地址;浏览器通过 (2) 查询上述输入信息所指的WEB服务器的IP地址;浏览器通过网络与该IP地址处的WEB服务器的 (3) 服务端之间建立一条 (4) 连接 ; 浏览器依照相关协议发送 (5) 命令 ; WEB 服务器根据命令取出文档,发送回来;浏览器释放连接,显示该文档.

1. A. URL B. EMS C. NDS D. DNS
2. A. NAT B.EMS C. NDS D. DNS
3. A. HTML B.HTTP C. SMTP D.SNMP
4. A. RTP B. IP C. TCP D. UDP
5. A. TCP B. GET C. UDP D. PUT

1. A 2.D 3. B 4 .C 5 .B

## 2. 当 n=5 时 , 下列函数返回值是:( A )

int foo(int n){

if (n<2)return n ;

retrun foo(n-1)+foo(n-2);

}

1. 5 B. 7 C. 8 D. 1

## 3.在关系模型中,实现”关系中不允许出现相同的元组”的约束是通过( B )

1. 候选键 B. 主键 C. 外键 D. 超键

## 4.下列程序的输出是: ( D )

#define add(a,b) a+b

Int main()

{

Print(“%d”, 5\*add(3,4))

Return 0;

}

1. 23 B. 35 C. 16 D.19

解析：

    宏是完全的文本替换，宏替换时容易犯的错误。使用宏时，带上括号是安全的做法。

## 5. IP 地址 131.153.12.71 是一个( B )类的IP地址.

1. A B. B C. C D. D

## 6. S市 A , B 共有两个区, 人口比例为3:5 ,据历史统计 A区的犯罪率为 0.01%,B区为 0.015% ,现有一起新案件发生在 S 市 那么案件发生在A区的可能性有多大( C )

1. 37.5% B. 32.5% C. 28.6% D. 26.1%

解析：犯罪率可以理解为AB两区的犯罪人数与总人口数的比。由此不难列出下式：

    ( 3\*0.01% ) / ( 3\*0.01% + 5\*0.015% ) = 0.2587 = 28.6%

## 7. 假设某计算机具有1MB的内存(目前使用的计算机往往具有64MB以上的内存),并按字节编址,为了能存取该内存各地址的内容,其地址寄存器至少需要二进制\_\_③\_位,为使4字节组成的字段从存储器中一次读出，要求存放在存储器中的字边界对齐，一个字的地址码应\_①\_。若存储周期为200ns,且每个周期可访问4个字节,则该存储器带宽为\_\_④\_\_bit/s 假如程序员所用的地址为\_\_③\_\_\_而真正访问内 存地址称为\_\_②\_\_

A (1) 10 (2) 16 (3) 20 (4) 32

B (1)最低两位00 (2) 最低两位为10 (3) 最低两位为00 (4)最高两位为10 C. (1)20M (2)40M (3) 80M (4) 160M

D. (1) 有效地址 (2) 程序地址 (3) 逻辑地址 (4) 物理地址

E. (1) 指令地址 (2) 物理地址 (3)内存地址 (4) 数据地址

## 8. 一台主机要实现通过局域网与另一个局域网通信,需要做的工作是( C).

1. 配置域名服务器
2. 定义一条本机指向所在网络的路由
3. 定义一条本机指向所在网络网关的路由
4. 定义一条本机指向目标网络网关的路由

## 9. Linux 文件权限一共10位长度,分成四段,第三段表示的内容是( C).

1. 文件类型
2. 文件所有者的权限
3. 文件所有者所在组的权限
4. 其他用户的权限

## 10. PHP 和 MYSQL 的联合使用 解决了( C ).

1. 在Proxy 上处理数据库的访问问题
2. 在WWW服务器上处理黑客的非法访问问题
3. 在WWW服务器上处理数据库的访问问题
4. 在Sendmail 邮件系统上处理数据库的访问问题

## 11. 以下关于HTTP 状态码的描述,错误的是?( )

A. 100 代表客户端应当继续发送请求

B. 2xx 代表请求已成功被服务器接收,理解,并接受

C. 301 代表被请求的资源已永久移动到新位置,用于重定向

D. 4xx 代表服务器在处理请求的过程中有错误或者异常状态发生

## 12. 下面代码会输出什么:

Def f(x,L=[])

For i in range(x):

L.append(i\*i)

Print(L)

f(2)

f(3,[3,2,1])

f(3)

**答案:** [0, 1]

[3, 2, 1, 0, 1, 4]

[0, 1, 0, 1, 4]

## 13. 什么是 lambda 函数?它有什么好处?

lambda只是一个表达式,它用来创建一个函数对象

当lambda表达式执行时，返回的是冒号后面的表达式的值

lambda表达式创建的函数只能包含一条表达式

lambda 比函数简单,且可以随时创建和销毁，少占用内存空间

## 14. Python里面 match()和search()的区别?

match（）函数只检测RE是不是在string的开始位置匹配， search()会扫描整个string查找匹配, 也就是说match（）只有在0位置匹配成功的话才有返回，如果不是开始位置匹配成功的话，match()就返回none

## 15. http 连接请求中 get 和 post 的区别 , 你还了解其他的方式么

get 和 post 的区别:

1. GET使用URL或Cookie传参。而POST将数据放在BODY中。

2. GET的URL会有长度上的限制，则POST的数据则可以非常大。

3. POST比GET安全，因为数据在地址栏上不可见。

其他方式:

Delete 删除数据。可以通过Get/Post来实现。

Put，增加、放置数据，可以通过Get/Post来实现。

## 16. 简述一下 socket TCP 和HTTP 之间的关系

最大区别socket是网络层，http是应用层.  
socket是一个程序组件，它支持TCP,UDP等网络通讯协议，也就是通过socket这个东西你可以和任何互联网或局域网上的计算机通讯。TCP,UDP是一个网络层协议，网络层协议不管你发的内容是啥，他只负责把你想法的东西发到对面，发的是啥，他完全不管，因为他不是应用层。http是应用层协议，在socket的TCP层之上，他是管你发送的内容是什么的，比如你发的是啥，图片还是文字还是视频数据流，发成功了没有，用什么语言发，都做了详细的定义，这样可以保证写软件的人只要按照约定处理数据，就能得到想要的结果。

## 17. Session 和 cookie 的联系与区别 : session 为什么说是安全的:

cookie 和session 的区别：  
1、cookie数据存放在客户的浏览器上，session数据放在服务器上。  
2、cookie不是很安全，别人可以分析存放在本地的COOKIE并进行[COOKIE欺骗](https://www.baidu.com/s?wd=COOKIE%E6%AC%BA%E9%AA%97&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y4uHm4n1ndnW99nHTkmWb10ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWD1rHcLnj0dnWbdPjf1nHb3r0" \t "_blank),考虑到安全应当使用session。  
3、session会在一定时间内保存在服务器上。当访问增多，会比较占用你服务器的性能,考虑到减轻服务器性能方面，应当使用COOKIE。  
4、单个cookie保存的数据不能超过4K，很多浏览器都限制一个站点最多保存20个cookie。  
cookie 和session 的联系：  
session是通过cookie来工作的  
session和cookie之间是通过$\_COOKIE['PHPSESSID']来联系的，通过$\_COOKIE['PHPSESSID']可以知道session的id，从而获取到其他的信息

## 18. 你看过 Django 的admin 源码么;看过flask 的源码么? 你如何理解开源?

## 19. 进程间 IPC(inter-Process Communication) 有哪些?

linux常用的进程间的通讯方式

（1）、管道(pipe)：管道可用于具有亲缘关系的进程间的通信，是一种半双工的方式，数据只能单向流动，允许一个进程和另一个与它有共同祖先的进程之间进行通信。

（2）、命名管道(named pipe)：命名管道克服了管道没有名字的限制，同时除了具有管道的功能外（也是半双工），它还允许无亲缘关系进程间的通信。命名管道在文件系统中有对应的文件名。命名管道通过命令mkfifo或系统调用mkfifo来创建。

（3）、信号（signal）：信号是比较复杂的通信方式，用于通知接收进程有某种事件发生了，除了进程间通信外，进程还可以发送信号给进程本身；linux除了支持Unix早期信号语义函数sigal外，还支持语义符合Posix.1标准的信号函数sigaction（实际上，该函数是基于BSD的，BSD为了实现可靠信号机制，又能够统一对外接口，用sigaction函数重新实现了signal函数）。

（4）、消息队列：消息队列是消息的链接表，包括Posix消息队列system V消息队列。有足够权限的进程可以向队列中添加消息，被赋予读权限的进程则可以读走队列中的消息。消息队列克服了信号承载信息量少，管道只能承载无格式字节流以及缓冲区大小受限等缺点。

（5）、共享内存：使得多个进程可以访问同一块内存空间，是最快的可用IPC形式。是针对其他通信机制运行效率较低而设计的。往往与其它通信机制，如信号量结合使用，来达到进程间的同步及互斥。

（6）、内存映射：内存映射允许任何多个进程间通信，每一个使用该机制的进程通过把一个共享的文件映射到自己的进程地址空间来实现它。

（7）、信号量（semaphore）：主要作为进程间以及同一进程不同线程之间的同步手段。

（8）、套接字（Socket）：更为一般的进程间通信机制，可用于不同机器之间的进程间通信。起初是由Unix系统的BSD分支开发出来的，但现在一般可以移植到其它类Unix系统上：Linux和System V的变种都支持套接字。

## 20 软件开发常用的设计模式有哪几种?

其中创建型有：   
 一、Singleton，[单例模式](https://www.baidu.com/s?wd=%E5%8D%95%E4%BE%8B%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)：保证一个类只有一个实例，并提供一个访问它的全局访问点   
 二、Abstract Factory，抽象工厂：提供一个创建一系列相关或相互依赖对象的接口，而无须指定它们的具体类。   
 三、Factory Method，工厂方法：定义一个用于创建对象的接口，让子类决定实例化哪一个类，Factory Method使一个类的实例化延迟到了子类。   
 四、Builder，建造模式：将一个复杂对象的构建与他的表示相分离，使得同样的构建过程可以创建不同的表示。   
 五、Prototype，[原型模式](https://www.baidu.com/s?wd=%E5%8E%9F%E5%9E%8B%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)：用原型实例指定创建对象的种类，并且通过拷贝这些原型来创建新的对象。  
行为型有：  
 六、Iterator，[迭代器模式](https://www.baidu.com/s?wd=%E8%BF%AD%E4%BB%A3%E5%99%A8%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)：提供一个方法顺序访问一个聚合对象的各个元素，而又不需要暴露该对象的内部表示。   
 七、Observer，[观察者模式](https://www.baidu.com/s?wd=%E8%A7%82%E5%AF%9F%E8%80%85%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)：定义对象间一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知自动更新。   
 八、Template Method，模板方法：定义一个操作中的算法的骨架，而将一些步骤延迟到子类中，TemplateMethod使得子类可以不改变一个算法的结构即可以重定义该算法得某些特定步骤。   
 九、Command，[命令模式](https://www.baidu.com/s?wd=%E5%91%BD%E4%BB%A4%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)：将一个请求封装为一个对象，从而使你可以用不同的请求对客户进行参数化，对请求排队和记录请求日志，以及支持可撤销的操作。   
 十、State，[状态模式](https://www.baidu.com/s?wd=%E7%8A%B6%E6%80%81%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)：允许对象在其内部状态改变时改变他的行为。对象看起来似乎改变了他的类。   
 十一、Strategy，[策略模式](https://www.baidu.com/s?wd=%E7%AD%96%E7%95%A5%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)：定义一系列的算法，把他们一个个封装起来，并使他们可以互相替换，本模式使得算法可以独立于使用它们的客户。   
 十二、China of Responsibility，职责链模式：使多个对象都有机会处理请求，从而避免请求的送发者和接收者之间的耦合关系  
 十三、Mediator，中介者模式：用一个中介对象封装一些列的对象交互。   
 十四、Visitor，[访问者模式](https://www.baidu.com/s?wd=%E8%AE%BF%E9%97%AE%E8%80%85%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)：表示一个作用于某对象结构中的各元素的操作，它使你可以在不改变各元素类的前提下定义作用于这个元素的新操作。   
 十五、Interpreter，解释器模式：给定一个语言，定义他的文法的一个表示，并定义一个解释器，这个解释器使用该表示来解释语言中的句子。   
 十六、Memento，[备忘录模式](https://www.baidu.com/s?wd=%E5%A4%87%E5%BF%98%E5%BD%95%E6%A8%A1%E5%BC%8F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)：在不破坏对象的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态。   
结构型有：   
 十七、Composite，组合模式：将对象组合成[树形结构](https://www.baidu.com/s?wd=%E6%A0%91%E5%BD%A2%E7%BB%93%E6%9E%84&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1Y3m1ubnWbLuHTYrHcvnWT30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EPjfzrHcYnjR1" \t "_blank)以表示部分整体的关系，Composite使得用户对单个对象和组合对象的使用具有一致性。   
 十八、Facade，外观模式：为子系统中的一组接口提供一致的界面，fa?ade提供了一高层接口，这个接口使得子系统更容易使用。   
 十九、Proxy，代理模式：为其他对象提供一种代理以控制对这个对象的访问   
 二十、Adapter,适配器模式：将一类的接口转换成客户希望的另外一个接口，Adapter模式使得原本由于接口不兼容而不能一起工作那些类可以一起工作。   
 二十一、Decrator，装饰模式：动态地给一个对象增加一些额外的职责，就增加的功能来说，Decorator模式相比生成子类更加灵活。  
 二十二、Bridge，桥模式：将抽象部分与它的实现部分相分离，使他们可以独立的变化。   
 二十三、Flyweight，享元模式

## 21. 已知Apache 服务的访问日志按天记录在服务器本地目录/app/logs下,由于磁盘空间紧张,现在要求只能保留最近7天访问日志!请问如何解决?如果有多种办法,请逐条列出

方法一：find /app/logs/ -type f -name "\*.log"-mtime +7 |xargs rm -f

方法二rm -f $( find /app/logs/ -type f -name"\*.log" -mtime +7)

方法三：find /app/logs/ -type f -name "\*.log"-mtime +7 -exec rm -f {} \;

## 22. 磁盘报错: nospace left on device 磁盘满了, 但是df-h 查看空间没有满,为什么?怎么处理?

用df -i查看一下/data分区的索引节点(inode)，发现已经用满(IUsed=100%)，导致系统无法创建新目录和文件。  
inode译成中文就是索引节点，每个存储设备（例如硬盘）或存储设备的分区被格式化为文件系统后，应该有两部份，  
一部份是inode，而inode呢，就是用来存储这些数据的信息，这些信息包括文件大小、属主、归属的用户组、读写权限等。  
另一部份是Block，Block是用来存储数据用的。inode为每个文件进行信息索引，所以就有了inode的数值。操作系统根据指令，能通过inode值最快的找到相对应的文件。  
　而这台服务器的Block虽然还有剩余，但inode已经用满，因此在创建新目录或文件时，系统提示磁盘空间不足.

**原因:/data/cache目录中存在数量非常多的小字节缓存文件，占用的Block不多，但是占用了大量的inode。**

**解决方案：**  
　　 1、删除/data/cache目录中的部分文件，释放出/data分区的一部分inode，特别要留意那些spool出来的文件，这种文件一般会占用比较多的节点，因为比较小而且零碎，同时要多留意日志文件信息等  
  
　　2、用软连接将空闲分区/opt中的newcache目录连接到/data/cache，使用/opt分区的inode来缓解/data分区inode不足的问题：  
　　ln -s /opt/newcache /data/cache

　　3、更换服务器，用高配置的服务器替换低配置的服务器。很多时候用钱去解决问题比用技术更有效，堆在我办公桌上5台全新的 DELL PowerEdge 2950 服务器即将运往IDC机房。一般不建议