

## 目录

# Contents

 <b>技巧 1 BAT 等企业如何面试 .....</b>	<b>1</b>
1. 面试流程及问题解析 .....	1
2. 面试时的基本礼仪及着装 .....	7
 <b>技巧 2 你的职场生涯可以这样来规划 .....</b>	<b>9</b>
1. 缺乏工作经验的应届毕业生 .....	9
2. 更换工作的程序员 .....	11
3. 快乐地工作 .....	12
 <b>技巧 3 66 道面试题让你轻松拿到 Offer .....</b>	<b>13</b>
1. 程序设计基本概念 .....	13
2. 指针与引用 .....	16
3. 面向对象 .....	20
4. 继承与接口 .....	22
5. 树、图、哈希表 .....	26
6. 字符串 .....	27
7. 软件测试 .....	32
8. 操作系统 .....	33
9. 数据库与 SQL 语言 .....	37
10. 计算机网络及分布式系统 .....	38
11. 网络协议 .....	40
12. 网络安全 .....	41
13. 英语面试 .....	42



## 技巧 1

## 1. 面试流程及问题解析

无论是程序员还是其他职业的面试，虽然都是提交简历、电话面试、面谈、拿到 Offer 等这一系列过程，但是每一步都很重要！首先简历要精炼、漂亮、大方，尽量突出自己的优势，避开自己的劣势。电话面试还好，因为不用面对面交谈，这样可以更好地把握自己的语言。面谈是最关键的一步，准备不充分就会紧张，一紧张就容易出错。不过很多面试官都可以接受面试者的紧张，只要不太过分，问题就不大。一般来说中大型企业有很多轮的面试，就拿微软来说，官方渠道需要 12 轮面试，即使内部推荐也需要 4 轮，而且只要有一个面试官说你不行，你就没戏了。搞定面试官有很多方法，但是最重要的一条就是：面试官也是人，所以要以最短的时间，在他/她了解你之前——了解他/她，并合乎时宜地与他/她展开对话。最后就是要用平常心去面对，千万不要执着于某个职位，这有可能会令你失态导致失败，而平常心有时反倒会有意想不到的效果。

### （一）简历

据统计，80%的简历都是不合格的。不少企业的人事管理者抱怨很多简历的格式很糟糕。那么如何写好一份简历呢？

首先，简历的总篇幅要控制在 2 页左右，如下图所示。

[illegible]

太长的简历反而会淹没一些有价值的闪光点，但也不要为了减少篇幅而字体到影响阅读的程度。

其次，简历一定要真实客观，按照实际情况填写，不要填写任何虚假内容，否则即使侥幸通过筛选，面试时也会露出马脚的，对以后的职业生涯也会有很严重的影响。

另外，不要过分谦虚。简历上注明你的求职岗位，要根据该岗位的要求，有侧重地描述自己。

最后，在文字、排版、格式上不要出现错误，一份干净整洁的简历是对面试企业最起码的尊重。如果应聘外资企业或跨国公司，则一定要认真对待英文简历的编写，因为它会体现你的实际英文水平。

下面是英文简历模板的链接。

## （二）笔试

笔试是非常关键的一个环节，不论你有多大的才干和多么广博的知识，如果未能通过笔试，则无缘下面的面试。

目前几乎所有的 IT 公司都是以 C/C++ 或 Java 为主进行笔试，外企的话还要求具备一定的英语基础。编程语言本身并没有高低之分，但相对来说，考到 Delphi 或者 VB 的可能性很小。作为应届毕业生，如果只学过 VB、VF，而对 C 系语言不熟的话，在笔试中就会很吃亏。下面是一些阿里巴巴 B2B 第一阶段面试题目。

**题目 1** 两个二进制数的异或结果是多少？

**题目 2** 递归函数最终会结束，那么这个函数一定（不定项选择题）：

- A. 使用了局部变量
- B. 有一个分支不调用自身
- C. 使用了全局变量或者使用了一个或多个参数

**题目 3** 以下函数的结果是什么？

---

```
Int cal(int x)
{
    if(x==0)
        return 0;
    else
        return x+cal(x-1);
}
```

---

#### 题目 4 以下程序的结果是什么？

```

Void foo(int*a,int* b)
{
    *a = *a+*b
    *b = *a-*b
    *a = *a-*b
}
Void main()
{
    int a=1, b=2, c=3
    foo(&a,&b);
    foo(&b,&c)
    foo(&c,&a)
    printf("%d,%d,%d",a,b,c);
}

```

#### 题目 5 下面哪项不是链表相对于数组的特点？

- A 方便删除
- B 方便插入
- C 长度可变
- D 储存空间小

#### 题目 6 $T(n) = 25T(n/5) + n^2$ 的时间复杂程度是什么？

#### 题目 7 $n$ 个顶点、 $m$ 条边的全连通图，至少去掉几条边才能构成一个树？

#### 题目 8 正则表达式 $(01; 10; 1001; 0110)^*$ 与下列哪个表达式一样？

1.  $(0; 1)^*$
2.  $(01; 01)^*$
3.  $(01; 10)^*$
4.  $(11; 01)^*$
5.  $(01; 1)^*$

#### 题目 9 如何减少换页错误？

- A 进程倾向于占用 CPU
- B 访问局部性 (locality of reference) 满足进程要求
- C 进程倾向于占用 I/O
- D 使用基于最短剩余时间 (shortest remaining time) 的调度机制
- E 减少页大小

**题目 10** 实现  $N \times N$  矩阵的乘法，矩阵由一维数组表示。

**题目 11** 找到单向链表中间那个元素，如果有两个则取前面一个。

**题目 12** 有长度为  $n$  的整数数组，找出其中任意  $(n-1)$  个乘积最大的那一组，只能用乘法，不能用除法。要求对算法的时间复杂度和空间复杂度做出分析，不要求写程序。

从上述笔试题来看，题目本身都不是太难。对于企业而言，不管是阿里还是微软，招聘者都只是 HR 而不是神仙，不要把题目想得太神秘了。但是也不要太轻敌，事先要了解面试的公司，并根据情况做好相应的准备。

### （三）电话面试

电话面试，一般是面试者提交简历后，面试开始的第一步，由公司 HR 负责，主要是对简历上一些模糊信息的确认、对应聘职位简单技术问题的提问，以及对英文能力的考察。

电话面试之前要调整好电话，保证通话质量，以及电话的畅通。

电话面试常见的问题如下。

**面试例题 1：**请你介绍一下自己。

这个问题需要选择重点来回答，至于什么时候上的小学、什么时候初中毕业就不用讲了。你可以这么说：

A：我是某某大学毕业的，我的成绩优异。我的研究方向是计算机软件与理论。我的本科教育给了我宽广的视野，我涉猎的课程有英语、网络和程序设计。我对编程方面有很深的了解。以及可以说说你的业余爱好。

**面试例题 2：**你最大的弱点是什么？

这个问题需要正面来回答，每个人都有弱点，最佳策略是承认你的弱点，但同时要表明你在予以克服，并有克服弱点的能力和计划。可能的话，也可以说一项会给公司带来好处的弱点，例如“我是一个很执着的人，一件微不足道的小事也要把它做到完美”。对于一个应届毕业生而言，缺乏工作经验这种不是很大的弱点，你也可以说出来。

A：我缺乏工作经验，但是我正在学习。

B：我缺乏远见。但是我可以多读书。

C：我只是刚毕业。

**面试例题 3：**你对本公司的情况了解吗？

A：了解一点。正如你刚才所提到的那样，贵公司是一家美资公司。据

我所知，××公司是一家世界闻名的生产数据库产品和应用软件的公司。你们的产品×××和美国某数据库公司数据库都深受世界各地用户的欢迎。

**面试题例4：**在你的项目中用到了哪种程序？在此过程中你学到了什么？

A：基于ASP.NET和SQL 2000平台基础，我们完成了××大学网络选课系统。学校中的每个人都可以在网上选择、取消、查询课程。这个项目是一个B/S结构系统；代码是用C#编写的，在.NET平台上运行。在项目中，我使用ADO接口来支持数据库程序。在此之后，我参加了对整个系统的测试。

为了实现后台数据库，我进行了数据库的设计，包括主键、外键的设计、链接、视图及其他，并运用SQL语句实现数据的查找、删除和修改。其中虽然遇到了大量的困难，但我尽我所能地去克服了它们，最终获得了提升和自我的丰富。

**面试题例5：**你如何理解你所应聘的职位？你未来几年的规划是什么？

A：我应聘的是QA职位，QA意味着质量保证。我们应该很好地与研发部门交流。我计划对项目的结构做深入的了解，与我的同事们很好地合作。希望能充分展示我在这个行业的能力和智慧。

B：也许有机会，我将会从事管理工作。我在上一个工作中积累了一些行政经验，我将来也许会运用我组织和计划上的经验和技巧来完成工作。

**面试题例6：**在你眼中，管理者是什么样的？

A：项目管理者是那种负责项目从概念到实施整个过程的计划、协调、控制的人或公司，他（们）能满足项目的需求，并确保项目在有限的成本内按时完成，达到要求的质量标准。

B：项目管理者是一个策划者、协调者和领导者，是研发人员的好朋友，也是项目流程的报告者。

**面试题例7：**你目前所在的公司，有什么吸引你的地方吗？

不要说“没有，我讨厌现在的公司”。可以选择说一说目前公司和所申请公司共同的优点。

如果这是你的第一份工作，你可以谈谈在学习或实习期间喜欢什么。

**面试题例8：**你理想的工作环境是什么？

A：我喜欢在一个和谐的环境中工作。每个人尽最大的努力为公司做事，完成一份工作后我们会得到满足和成就感，我的努力能够得到公司的嘉许。

作为公司的一员我能从他人身上受益，我也会令他人受益。

**面试题 9:** 你觉得要获得职业上的成功需要具备什么样的特质及能力？

A: 首先，要有极强的爱好，以及扎实、广泛、艺术性的教育为背景，特别是在技术为主导的职业领域中。这些基本条件，加上对学习的热爱、解决问题的能力、广泛的兴趣、成功与求胜的决心、很强的沟通技巧、努力工作的能力，都是能够帮助我在职业生涯中取得成功的重要品质。如今新技术不断取代旧的技术，为了成功我们要成为能跟上步伐的快速学习者，否则你将被他人击倒。所有的优点联合起来，塑造成一个能在不断变换的信息社会中生存的坚实团队中的一员。我相信我拥有的品质，我有信心成为贵公司成功团队中的一员。

**面试题 10:** 你的同事或同学是怎么描述你的？

你可以通过这个问题来向面试官展示你的社交意识，可以说通过与同事或同学的交流协作，你了解到了自己在别人眼中的样子。

同时，你可以用这个问题来表明你有自我认知感。你知道自己的长处和短处，以及你的长处或短处能给团队带来什么。

诚实回答，不要过度自嘲。

**面试题 11:** 你有什么问题要问我吗？

如果你对公司已经提前了解，并且做足了功课，可以提问与你职位相关的问题。例如，这个职位承担的职能，以及公司这个职位还有哪些期望等。彰显你对这一职位的认识，且具有积极进取的心态。

另外，如果你对公司晋升和薪酬机制不太清楚，尽量不要直接问工资、社保、五险一金等具体问题，尤其不要向非 HR 面试官问这类问题，可以侧面询问这个职位的成长线、薪酬与成长的匹配关系、薪酬策略是标杆型还是追赶型（市场分位）、自己怎么努力才能加快成长？

如果你对企业文化有所了解，可以问如果我有幸加入这个团队，我如何尊崇公司的文化、更好地融入团队等这类开放性的问题。一方面显示你愿意融入文化，有团队意识；另一方面也给面试官一次“谆谆教导你”的机会。这类问题，任何面试官都可以回答你，并且会讲得头头是道，你频频点头就可以了。面试官讲得越嗨，给你加分的可能性就越大。

#### （四）面试

对于求职者而言，面试是重要的环节，一定要守时。如果不能按时参加



面试，则要事先通知对方。着装上不需要过分准备，干净、舒服就好。IT公司对程序员的着装没有什么要求，但是精神状态一定要好，这样会让你显得很自信。

一个比较好的面试能够了解求职者擅长哪方面而哪方面不足，应根据应聘职位等级进行准备。

应聘初级职位，会针对你的编程能力和以往的项目经验进行重点考查。如果面试官针对你做的某个项目反复提问的话，要么面试官在这个方面很精通，要么就是未来的职位需要这方面的技术。

应聘中级职位，不但会考查代码编写能力，而且会对软件架构或相关行业知识进行考查。

应聘高级职位，应聘者肯定对技术或某个行业有一定程度的了解，这时主要看你与职位的契合程度、企业文化的配比性及整体感觉。

面谈的时候，要与面试官保持目光接触，显示出你的友好、真诚、自信和果断。遇到不太熟悉的问题时，也无需紧张。面试过程中是允许沉默的，你可以用这个时间来思考。

## （五）签约

到这个阶段就证明你已经拿到了 Offer，正式的 Offer 应该提明确薪水（税前还是税后）、补助（税前还是税后）、工作岗位、工作时间、工作地点、保险、公积金等信息。

在签约前，一定要向 HR 或其他人打听清楚户口、待遇、工作内容、加班 / 出差情况、培训、发展机会、签约年限及违约金等情况。

## 2. 面试时的基本礼仪及着装

女装的选择：女装选择的范围比较广，职业套装是比较稳妥的选择，但是穿套装要注意不要把自己弄得老气横秋，可以选择色彩鲜艳或者款式活泼的衬衣搭配，体现稳重又活泼的个性。也可以选择休闲的套装，注意款式简洁，不要有繁复的装饰。夏季宜选择有领子的衬衣，尽量不穿圆领、鸡心领的 T 恤。需要注意以下几点。

（1）如果不习惯职业套装就不要勉强。

（2）鞋跟不要太高。

（3）不要穿吊带、背心式衣服。

（4）衣服材质不能太透明。

男性的着装搭配要注意四个不能：不能苟同、不能太标新立异、不能太

含糊、不能太随便。

(1) 不能苟同。现在都流行穿正装面试,如西装、领带、西裤及闪亮的皮鞋等,但要看面试什么样的工作,应与工作要求相符。

(2) 不能太标新立异。衣着有个性但不另类。

(3) 不能太含糊。对待自己也好,对待考官也好,要有装扮上的礼貌,注重细节,衣服不能褶皱、不能邋遢。

(4) 不能太随便要讲究和谐,衣服不能太随便,哪怕是个性使然,也要穿着大方。

注意,运动装轻松、随意、舒适,但不适合面试。

发式:男女发式均应简洁明快,男士不宜留长发,女士的发型可以多样化,但要注意答题时如果需要用手去捋头发,说明发式不适宜,应该束发。不宜过于时尚,发饰最好不超过一个。男士最好不染发,女士染发也应该注意颜色自然。

礼仪、举止:礼仪举止往往会直接反映一个人的素质,所以仪表举止在面试时是专门有评价分数的。面试其实就是给考官一个总体的印象,然后让考官给出一个评价。这个测试从考生叩门时就开始了,所以应该注意每一个细节。

### (1) 进门

进入面试室之前,应轻叩房门二、三下(若有工作人员导引则不必如此),再轻轻推门而入。进门后,背对考官,将房门关上。注意拉住把手,动作要轻,如果门是碰锁,最好先旋起锁舌,关上门后,再放开。然后自然地扫视一下整个房间,确定面试考场的基本布局(包括自己的座椅位置),然后走向自己的位置。

### (2) 招呼

站立在自己的位置上时,身体要正对考官,一般大规模的面试,考官是不会和考生握手的,当然假如考官主动伸出手,你要毫不迟疑地伸手和考官握手。然后略带微笑注视主考官,主考官会示意你坐下,可以微微欠身点头表示感谢,也可以轻轻地说声“谢谢”。

注意:虽然说是礼多人不怪,但是我们还是要注意自然,遵守约定俗成。曾有个面试者进试场后一边大声说“各位考官好!”一边一个九十度的鞠躬,吓考官们一跳,这样的礼仪反而弄巧成拙。

### (3) 坐姿

入座(离座时也一样)时动作要轻盈和缓,从容不迫。不要径直跌坐在

位子上。落座后，不要坐得太满，最好坐椅面的一半到三分之二的部分，这样人容易坐直。双手自然放在桌上或膝上即可。

注意：切忌晃腿、搓手、东张西望等小动作，现在的年轻人有玩笔的习惯，在考场上一定要避免。假如积习已深，难以控制自己，就把笔放在桌上，需要时再拿。

#### （4）目光

对话的同时目光也要进行交流，这是对人起码的尊重。所以面试时不管自己心里多紧张也要注视主考官，否则一是失礼，二是把自己胆怯的一面暴露了出来。如果实在不敢注视考官的眼睛，也可以把目光对着考官的额头。

#### （5）离席

当考官提示面试结束时，不管自我感觉如何，还是要注意礼节。起立，面对面试官，微微欠身点头表示感谢，口中也可以轻轻地说声“谢谢”，然后再离去。出门时仍要注意把握门拉手，出去、面向门并轻轻关上考场门。

#### （6）冷场应急措施

面试时如果碰到难以的回答问题而冷场的话会很难堪，也会加剧考生的紧张心理，这时可以这样来回答：对不起，由于我学习不够，对这个问题所涉及的知识缺乏了解，我以后一定加强学习，拓展自己的知识面。态度要谦逊、诚恳、落落大方。

## 技巧2 你的职场生涯可以这样来规划

### 1. 缺乏工作经验的应届毕业生

即将毕业的同学们对自己的目标职位都很模糊，只要是计算机相关的工作都想试一下。但是现在公司看重的除了学生的沟通、团队协作、学习、外语等能力之外，也会关注应届毕业生在校及实习经历中与目标职位相关的经验。假设跟随导师或在实习过程中做过相关项目，那么在应聘时就更为容易一些。

例如，国内某高校的学生，通过课余时间开发了一个搜索引擎，对比那些商用搜索引擎，其搜索效率、数据量并不出色，但是该生同过自己编写的搜索引擎，详细了解了网络编程、网页爬虫等领域的知识。这个引擎也表现了他的专业技能水平，从而为他赢得了前往某国际网络公司工作的机会。

所以在大学期间，我们可以通过参加各种编程大赛和技术社团的活动来丰富自己的经验，以获取公司对你专业技能的肯定。

当然一份有分量的简历只是第一步。有分量指的是成绩尚可，有让他们感兴趣的实习经历，有一定的获奖经历，担任过一定的职务，英语能力还行。这样才能让你从众多的应聘者中被选拔出来参加初试，然后接下来的初试就要看你的真正功力了。

初试的要点在于基本功扎实、自信乐观、英语交流能力好、够聪明、够机灵。其中，基本功扎实并且够聪明尤为重要。某位毕业生参加 × × 公司的面试，过了印度技术官的英语技术面试，第二天参加他们的 Aptitude Test (智商测试)，误认为是态度测试 (Attitude Test)，结果没发挥好。智商测试通常会让你在很短的时间内做大量的逻辑题和智力题。不要在前面的题目上浪费太多时间，后面的题目往往更加简单。

没有必要因为自己的学校不够好而显得不够自信，只要打好技术功底，多参加正规的实践项目，找工作还是很容易的。此外不必太在意某行业整体的形势，整体形势好了，个人形势未必好，往往整体形式好了，个人容易盲目乐观，在准备不充分的情况下，很容易被淘汰。即使明年的整体形势比今年还要好，招人的公司比今年还要多，还是建议大家要脚踏实地，做好充分的心理准备和知识储备，摆正心态，尽自己最大的努力。写论文的同时抽空复习下基础课程，如数据结构、C、C++、TCP/IP、操作系统、计算机网络、UML、OOA&OOP，以及自己做过的项目等。

可能你找到了工作，而且不止一个，但是 Offer 再多也只能签一个公司。不用去羡慕那些手头有很多好 Offer 的人，他们其实很纠结，这是一种甜蜜的烦恼。用表格罗列出你最在意的方面，对几家公司做详细的比较(见下表)。做选择有时候很感性，理性的数据往往不如公司的一名普通员工给你的印象更能影响你的决定。

比较内容	权重指数	A 公司	B 公司
发展机会	20%		
公司前景	10%		
技术方向	10%		
培训机制	5%		
公司性质	15%		
人员规模	10%		
企业文化	5%		
行业排名	5%		
薪酬	20%		

## 2. 更换工作的程序员

我们首先要明白盲目而频繁的跳槽对我们的职业生涯是有害无益的，招聘方往往十分关注求职者的稳定性。一般来说，每份工作维持 1 年以上，或能够在一家公司工作满 3 年，才会对公司所在行业及这家公司有比较深入的了解。决定更换工作时，我们先要搞清楚要在哪个方向继续自己的职业生涯。假设目前你是某家公司的开发人员，要应聘更大规模公司的同等职位，我们应该注意下面两点。

首先，比起创业型公司，大公司的开发流程要求会更加规范和严格，有时候我们必须放弃某些编程习惯。严格的开发流程对文档的依赖性很大，我们必须做到文档优先。这样的一种环境，可能是初入大公司的程序员最难适应的一点。

其次，小公司里那种 Superman 型的程序员在大公司里很少见到。我曾经听某个程序员朋友抱怨他们公司的架构师连 ASP 代码都不会写，其实这是很正常的事情。架构师的工作是将业务需求变成计算机软件的模块和类，他们不需要了解具体代码的编写，只需要分析几种软件平台的实现难度和效率差异就够了。当然，大公司也有所谓的技术高手，但这种技术高手并不是精通几种开发语言的“万能钥匙”，而是对某种技术有深入理解、能够解决深层次问题的人。

中国的 IT 界，“技而优则仕”的比较多。很多技术出身的人员做到管理岗位后，关注的仍然是技术细节。但实际上，人员的管理也是门很大的学问。技术主管的个人风格会影响整个团队的氛围。如果主管不善沟通、只关心 Dead line，那么整个团队将会毫无活力，主管的技术再高超也不会得到信服。如果主管善于沟通、关心下属，那么整个团队就会生机勃勃，即使加班也有劲头。

假设你已不想再做开发，而要转向测试或其他相关岗位（如实施、技术支持甚至培训、售前等），那么一定要认真向目前在做这份工作的人员了解他们的实际职责与相关要求，确认是否可以接受转换岗位后带来的挑战。如果确定，则可以选择具有相同行业背景的目标职位，并且调整好自己的心理状态，给自己一段时间来适应这种改变。刚开始时感觉无从下手或者有较大落差是很正常的，最起码要在半年之后才能证实你和这个岗位的匹配度。

如果你现在已经有了较为明确的职业生涯发展规划，推荐使用倒推法使之切合实际并行之有效。以一个普通程序员为例，我们可以首先为自己的目

标设置一个年限，并列出现实现这个目标所需要的专业技能，然后使用倒推法确定我们的阶段目标，直至将这个阶段目标倒推至一个月后，那它就会是一个很具体的目标了。只要你坚持去做，就会逐步实现自己的最终目标。

除此之外，你还要时时关注业界动态，尽可能多地参加在职培训并且补充外语方面的技能。这样才能保持你继续前进的步伐。当然，最重要的是我们要把握好自己，把握好自己要走的路。其实任何一个职位都需要我们努力工作，任何一份工作都无法“钦定”我们的终身。求职“just a job”而已。找不到不用悲悲切切，找到了也不用狂喜。这只是人生中的众多历练之一。

### 3. 快乐地工作

年轻人在职业生涯的开始阶段需要注意的是，要做让自己感到快乐的事情，不要让自己今后几十年的人生总是提心吊胆，更不值得为了一份工作赔上自己的青春年华。人还是要看长远一点。很多时候，看起来最近的路，其实是最远的路；看起来最远的路，其实是最近的路。要让自己在职业的道路上走得更远，首先要让自己工作得快乐。如果一份工作让你觉得不快乐，甚至很受罪，那么你就是那个坚持不到终点的选手，即使你坚持到终点了，这样痛苦的人生有意思吗？其次，要对未来做好规划，尽量让自己劳逸结合。要知道那是个很漫长的过程，不要在一开始就把力气和耐心耗尽了，当力气和耐心耗尽而又遭遇挫折时，大多数人会陷入沮丧悲观，跳槽换工作也就变成很自然的事情。对于初入职场还不能很好控制自己心态的人，掌握好自己的节奏，不要跟着别人的脚步乱了自己的节奏，清楚自己在做什么，清楚自己的目标。至于别人上去了还是下去了，让他去吧，就当没看见。

对一个初入职场的人来说，入行后一定要跟个好领导、好老师。刚进社会的人做事情往往没有经验，需要有人言传身教。对于一个人的发展来说，一个好领导是非常重要的。所谓“好”的标准，不是他让你少干活多拿钱，而是达到以下3个标准。

首先，好领导要有宽广的心胸。如果一个领导每天都会发脾气，那几乎可以肯定他不是个心胸宽广的人。能发脾气的时候却不发脾气的领导，多半是非常厉害的领导。有些领导最大的毛病是容忍不了能力比自己强的人，所以常常看到的一个现象是，领导很有能力，手下一群庸才或者手下一群闲人。如果看到这样的环境，还是不要去的好。

其次，领导要愿意从下属的角度来思考问题，这一点其实是从面试的时候就能发现的。如果这位领导总是从自己的角度来考虑问题，几乎不听你说

什么，这就危险了。从下属的角度来考虑问题并不代表同意下属的说法，但他必须了解下属的立场，下属为什么这么想，然后他才有办法说服你。只关心自己怎么想的领导往往难以获得下属的信服。

第三，领导敢于承担责任。出了问题就把责任往下推，有了功劳就往自己身上揽，这样的领导不跟也罢。选择领导，要选择关键时刻能扛得住的领导，能够为下属的错误买单的领导，因为这是他作为领导的责任。

有可能，你碰不到好领导。因为他坐领导的位置，所以他的话就比较有道理，这是传统观念官本位的误区，可能有大量的这种无知、无能的领导，这对于你其实是好事。如果将来有一天你要超过他，你希望他比较聪明还是比较笨？相对来说这样的领导其实不难搞定，只是你要把自己的身段放下来而已。多认识一些人，多和比自己强的人打交道，同样能找到好的老师，不要和同样郁闷的人一起控诉社会、控诉老板，这帮不上你，只会让你更消极。职场上最忌讳的是你还在这家公司却又不不停地抱怨公司本身。正确的做法是和那些比你强的人打交道，看他们是怎么想的、怎么做的，学习他们，最终提升自己的能力。

希望所有读者都能快乐地工作，不断提升。

### 技巧 3 66 道面试题让你轻松拿到 Offer

#### 1. 程序设计基本概念

**面试题 1：**下面 C++ 代码的输出结果是什么？（中国某杀毒软件公司 2010 年 7 月笔试题）

```
C/C++ code
int i =1;
void main()
{
    int i =i;
}
```

- A. main() 里的 i 是一个未定义值
- B. main() 里的 i 值为 1
- C. 编译器不允许这种写法
- D. main() 里的 i 值为 0

**解析：**当面试者看到 `int i=1`，i 变量从声明的那一刻开始就是可见的了，



main() 里的 i 不是 1，因为它和 main() 外的 i 无关，而是一个未定义值。

**答案：**A

**面试题 2：**以下代码的输出结果是什么？（中国某通讯企业 H 公司 2007 年 7 月面试题）

---

```
#include <stdio.h>
main()
{
    int b=3
    int arr[]={6,7,8,9,10};
    sint *ptr=arr;
    *(ptr++)+=123;
    printf("%d,%d\n",*ptr,*(++ptr));
}
```

---

A.8 8

B.130 8

C.7 7

D.7 8

**解析：**C 中 printf 计算参数时是从右到左压栈的。几个输出结果分别如下。

1.printf( "%d\n",\*ptr); 此时 ptr 应指向第一个元素 6。

2.\* (ptr++)+=123 应为 \*ptr=\*ptr+123;ptr++, 此时 ptr 应指向第二个元素 7。

3.printf( "%d\n", \*(ptr-1)); 输出第一个元素 129，注意此时是经过计算的。printf( "%d\n",\*ptr); 输出第二个元素 7，此时 ptr 还是指向第二个元素 7。

4.printf( "%d,%d\n",\*ptr,\*(++ptr)); 从右到左运算，第一个是 (++ptr)，也就是 ptr++，\*ptr=8，此时 ptr 指向第三个元素 8，所以全部为 8。

**答案：**A

**面试题 3：**下面两段程序有两种写法，你青睐哪种？为什么？（美国某计算机嵌入式公司 2005 年 10 月面试题）

---

```
A.
//a is a variable
写法 1:
if( 'A'==a) {
    a++;
}
```

---



写法 2:

```
if(a=='A'){
    a++;
}
```

B.

写法 1:

```
for(i=0;i<8;i++) {
    X= i+Y+J*7;
    printf ("%d",x);
}
```

写法 2:

```
S=Y+J*7;
for(i=0;i<8;i++){
    printf("%d",i+s);
}
```

**答案:**

A: 第一种写法 ‘A’ ==a 比较好些。这时如果把 “==” 误写成 “=” 的话, 因为编译器不允许对常量赋值, 就可以检查到错误。

B: 第二种写法好一些, 将部分加法运算放到了循环体外, 提高了效率。缺点是程序不够简洁。

**面试题例 4:** 在 C++ 程序中调用被 C 编译器编译后的函数, 为什么要加 extern "C"?

**答案:** C++ 语言支持函数重载, C 语言不支持函数重载。函数被 C++ 编译后在库中的名字与 C 语言的不同。假设某个函数的原型为 void foo(int x, int y), 该函数被 C 编译器编译后在库中的名字为 \_foo, 而 C++ 编译器则会产生像 \_foo\_int\_int 之类的名字。

C++ 提供了 C 连接交换指定符号 extern "C" 解决名字匹配问题。

**面试题例 5:** 评价一下 C 与 C++ 的各自特点。如果一个程序既需要大量运算, 又要有一个好的用户界面, 还需要与其他软件大量交流, 应该怎样选择合适的语言?

**答案:** C 是一种结构化语言, 重点在于算法和数据结构。C 程序的设计首先考虑的是如何通过一个过程, 对输入 (或环境条件) 进行运算处理得到输出 (或实现过程 (事务) 控制)。而对于 C++, 首先考虑的是如何构造一个对象模型, 让这个模型能够契合与之对应的问题域, 这样就可以通过获取对象的状态信息得到输出或实现过程 (事务) 控制。

对于大规模数值运算，C/C++ 和 Java/.NET 之间没有明显的性能差异。不过，如果运算涉及向量计算、矩阵运算，可以使用 FORTRAN 或者 MATLAB 编写计算组件（如 COM）。

大规模用户界面相关的软件可以考虑使用 .NET 进行开发（Windows 环境下），.NET 同 COM 之间的互操作十分容易，同时 .NET 对数据库访问的支持也相当好。

## 2. 指针与引用

**面试题例 6：**下面的程序哪里有错？（中国某计算机硬件公司 2005 年 12 月面试题）

---

```
#include <iostream>
using namespace std;
int main()
{
    int iv;                //1
    int iv2=1024;          //2
    int iv3=999;           //3
    int &reiv              //4;
    int &reiv2 = iv;        //5
    int &reiv3 iv;          //6
    int *pi;               //7
    *pi=5;                 //8
    pi=&iv3;               //9
    const double di ;      //10
    const double maxWage.=10.0; //11
    const double minWage =0.5;
    const double *pc = &maxWage; //12
    cout << pi;
    return 0;
}
```

---

**答案：**

1. 正确，很正常地声明了一个整型变量。
2. 正确，很正常地声明了一个整型变量，同时初始化这个变量。
3. 正确，理由同上。
4. 错误，声明了一个引用，但引用不能为空，必须同时初始化。
5. 正确，声明了一个引用 reiv2，同时初始化了，也就是 reiv2 是 iv 的别名。
6. 正确，理由同上。

7. 正确，声明了一个整数指针，但是并没有定义这个指针所指向的地址。
8. 错误，整数指针 pi 并没有指向实际的地址。在这种情况下就给它赋值是错误的，因为赋的值不知道该放到哪里去，从而造成错误。
9. 正确，整数指针 pi 指向 iv3 的实际地址。
10. 错误，const 常量赋值时，必须同时初始化。
11. 正确，const 常量赋值并同时初始化。
12. 正确，const 常量指针赋值并同时初始化。

### 面试题 7：运行“测试”后会发什么？

```
#include <iostream.h>

void GetMemory(char *p, int num)
{
    p = (char *)malloc(sizeof(char) * num);
}

int main()
{
    char *str = NULL;
    GetMemory(str, 100);
    strcpy(str, "hello");
    return 0;
}
```

**解析：**由于 void GetMemory(char \*p, int num) 中的 \*p 实际上是主函数中的一个 str 的一个副本，编译器总是要为函数的每个参数制作临时副本。在本例中，p 申请了新的内存，只是把 p 的内存地址改变了，但 str 丝毫未变。因为 GetMemory 没有返回值，因此 str 并不指向 p 所申请的那段内存，所以函数 GetMemory 并不能输出任何东西。事实上，每执行一次 GetMemory 就会申请一块内存，但申请的内存却不能有效释放，结果是内存一直被独占，最终造成内存泄露。

**答案：**程序崩溃。因为 GetMemory 并不能传递动态内存，Test 函数中的 str 一直都是 NULL。

**面试题 8：**句柄和指针的区别和联系是什么？（英国某计算机图形图像公司面试题）

**解析：**句柄是一个 32 位的整数，实际上是 Windows 在内存中维护的一个对象（窗口等）内存物理地址列表的整数索引。因为 Windows 的内存管

理经常会将当前空闲对象的内存释放掉，当需要访问时再重新提交到物理内存，所以对象的物理地址是变化的，不允许程序直接通过物理地址来访问对象。程序将想访问的对象的句柄传递给系统，系统根据句柄检索自己维护的对象列表就能知道程序想访问的对象及其物理地址了。

句柄是一种指向指针的指针。我们知道，所谓指针是一种内存地址。应用程序启动后，组成这个程序的各对象是驻留在内存的。如果简单地理解，似乎我们只要获知这个内存的首地址，就可以随时用这个地址访问对象。但是，如果真的这样认为，那么就大错特错了。我们知道，Windows 是个以虚拟内存为基础的操作系统。在这种系统环境下，Windows 内存管理器经常在内存中来回移动对象，以此来满足各种应用程序的内存需要。对象被移动意味着它的地址变化了。如果地址总是如此变化，我们该到哪里去找该对象呢？为了解决这个问题，Windows 操作系统为各应用程序腾出一些内存地址，用来专门登记各应用对象在内存中的地址变化，而这个地址（存储单元的位置）本身是不变的。Windows 内存管理器移动对象在内存中的位置后，把对象新的地址告知这个句柄地址来保存。这样我们只需记住这个句柄地址就可以间接地知道对象具体在内存中的哪个位置。这个地址是在对象装载（Load）时由系统分配的，当系统卸载时（Unload）又释放给系统。但是，必须注意的是，程序每次重新启动，系统不能保证分配给这个程序的句柄还是原来的那个句柄，而且绝大多数情况下的确不一样。假如我们把进入电影院看电影看成是一个应用程序的启动运行，那么系统给应用程序分配的句柄总是不一样，这和每次电影院售给我们的门票总是不同的座位是一样的道理。

HDC 是设备描述表句柄。CDC 是设备描述表类。用 GetSafeHwnd 和 FromHandle 可以互相转换。

**答案：**句柄和指针其实是两个截然不同的概念。Windows 系统用句柄标记系统资源，隐藏系统的信息。你只要知道有这个东西，然后去调用就行了，它是个 32bit 的 uint。指针则标记某个物理内存地址。

### 面试题 9：指针和引用的差别？

**答案：**

（1）非空区别。在任何情况下都不能使用指向空值的引用。一个引用必须总是指向某个对象。因此如果你使用一个变量并让它指向一个对象，但是该变量在某些时候也可能不指向任何对象，这时你应该把变量声明为指针，因为这样你可以赋空值给该变量。相反，如果变量肯定指向一个对象，例如

你的设计不允许变量为空，这时你就可以把变量声明为引用。不存在指向空值的引用这个事实意味着使用引用的代码效率比使用指针要高。

(2) 合法性区别。在使用引用之前不需要测试它的合法性。相反，指针则应该总是被测试，防止其为空。

(3) 可修改区别。指针与引用的另一个重要的区别是指针可以被重新赋值以指向另一个不同的对象。但是引用则总是指向在初始化时被指定的对象，以后不能改变，但是指定的对象其内容可以改变。

(4) 应用区别。总的来说，在以下情况下应该使用指针：一是考虑到存在不指向任何对象的可能（在这种情况下，能够设置指针为空），二是需要能够在不同的时刻指向不同的对象（在这种情况下，你能改变指针的指向）。如果总是指向一个对象并且一旦指向一个对象后就不会改变指向，那么应该使用引用。

**面试题例 10：**以下描述正确的是（ ）。（中国某网络企业 XL 公司 2010 年 7 月面试题）

- A. 函数的形参在函数未调用时预分配存储空间。
- B. 若函数的定义出现在主函数之前，则可以不必再说明。
- C. 若一个函数没有 return 语句，则什么值都不返回。
- D. 一般来说，函数的形参和实参的类型应该一致。

**解析：**

- A. 错误的，调用到实参才会分配空间。
- B. 函数需要在它被调用之前被声明，这个跟 main() 函数无关。
- C. 错误的，在主函数 main 中可以不写 return 语句，因为编译器会隐式返回 0；但是在一般函数中没 return 语句是不行的。
- D. 正确

**答案：**D

**面试题例 11：**C++ 中有了 malloc/free，为什么还需要 new/delete？

**答案：**malloc 与 free 是 C++/C 语言的标准库函数，new/delete 是 C++ 的运算符。它们都可用于申请动态内存和释放内存。

对于非内部数据类型的对象而言，只用 malloc/free 无法满足动态对象的要求。对象在创建的同时要自动执行构造函数，对象在消亡之前要自动执行析构函数。由于 malloc/free 是库函数而不是运算符，不在编译器控制权限之内，不能够把执行构造函数和析构函数的任务强加于 malloc/free。

因此 C++ 语言需要一个能完成动态内存分配和初始化工作的运算符 new, 以及一个能完成清理与释放内存工作的运算符 delete。new/delete 不是库函数, 而是运算符。

### 3. 面向对象

**面试题 12:** 以下选项中哪个不是面向对象设计。

- A: 继承
- B: 里氏代换原则
- C: 开闭原则
- D: 多态
- E: 防御式编程

**解析:** 面向对象设计的三原则: 封装, 继承, 多态。

**答案:** E

**面试题 13:** C++ 中的空类默认产生哪些类成员函数? (中国某综合软件公司 2005 年面试题)

---

```
class Empty
{
    Public;
};
```

---

**解析:** 类的概念问题。

**答案:** 对于一个空类, 编译器默认产生 4 个成员函数, 即默认构造函数、析构函数、复制构造函数和赋值函数。

**面试题 14:** structure 是否可以 constructor、destructor 及成员函数? 如果可以, 那么 structure 和 class 还有区别吗? (中国某计算机硬件公司 2005 年 11 月面试题)

**答案:** 区别是 class 中的变量默认是 private, struct 中的变量默认是 public。struct 可以有构造函数、析构函数, 之间也可以继承。C++ 中的 struc 其实和 class 意义一样, 唯一不同的就是 struct 里面默认的访问控制是 public, class 中默认的访问控制是 private。C++ 中存在 struct 关键字的唯一意义就是为了让 C 程序员有个归属感, 是为了让 C++ 编译器兼容以前用 C 开发的项目。

**面试题 15：**析构造函数可以为 virtual 型，构造函数则不能。那么为什么构造函数不能为虚呢？（某移动通信企业 2004 年面试题）

**答案：**虚函数采用一种虚调用的办法。虚调用是种可以在只有部分信息的情况下工作的机制，特别允许我们调用一个只知道接口而不知道其准确对象类型的函数。但是如果创建一个对象，你势必要知道对象的准确类型，因此构造函数不能为虚。

**面试题 16：**如果虚函数非常有效，我们是否可以把每个函数都声明为虚函数？

**答案：**不行，这是因为虚函数是有代价的：由于每个虚函数的对象都必须维护一个 v 表，因此在使用虚函数的时候都会产生一个系统开销。如果仅是一个很小的类，且不想派生其他类，那么根本没必要使用虚函数。

**面试题 17：**下面关于复制构造函数的说法哪一个是正确的？（中国某综合软件公司 2005 年面试题）

- A. 给每一个对象复制一个构造函数。
- B. 有一个默认的复制构造函数。
- C. 不能复制队列。
- D. 以上结果都正确。

**答案：**B

**面试题 18：**下面所列举的类，哪个不需要复制构造函数？（中国某杀毒软件公司 2004 年面试题）

一个矩阵类：动态分配，对象的建立是利用构造函数，删除是利用析构造函数。

- A. 一个花名册类：每一个对象对照着唯一的 ID。
- B. 一个 word 类：对象是字符串类和模板类。
- C. 一个图书馆类：由一系列书籍对象构成。

**解析：**

按照题意，寻找一个不需要复制构造函数的类。

A 选项要定义复制构造函数。

B 选项中，不自定义复制构造函数的话，势必造成两个对象的 ID 不唯一。至于说自定义了复制构造函数之后，如何保证新对象的 ID。当然语义上有损失，不是完全意义上的复制，但在这儿只能在保持语义和实现目的之间来

一个折中。

选 C 的原因是使用默认的复制构造，string 子对象和 vector 子对象的类都是成熟的类，都有合适的赋值操作，复制构造函数以避免“浅复制”问题。

D 选项显然是定义复制构造函数。

**答案：**C

**面试题 19：**什么是多态？

**答案：**开门，开窗户，开电视。在这里的“开”就是多态！

多态性可以简单地概括为“一个接口，多种方法”，在程序运行的过程中才决定调用的函数。多态性是面向对象编程领域的核心概念。

多态（Polymorphisn），按字面的意思就是“多种形状”。多态性是允许你将父对象设置成为和它的一个或更多的子对象相等的技术，赋值之后，父对象就可以根据当前赋值给它的子对象的特性以不同的方式运作。简单地说就是，允许将子类类型的指针赋值给父类类型的指针。多态性在 Object Pascal 和 C++ 中都是通过虚函数（Virtual Function）实现的。

#### 4. 继承与接口

**面试题 20：**公有继承和私有继承的区别是什么？（中国某计算机金融软件公司 2005 年面试题）

- A. 没有区别。
- B. 私有继承使对象都可以继承，只是不能访问。
- C. 私有继承使父类中的函数转化成私有。
- D. 私有继承使对象不能被派生类的子类访问。

**解析：**A 肯定错，因为子类只能继承父类的 protected 和 public，所以 B 也是错误的。

C 的叙述不全面，而且父类可能有自己的私有方法成员，所以也是错误的。

**答案：**D

**面试题 21：**一个类有 5 个虚方法，下列说法正确的是哪项？（英国某图形软件公司 2009 年 10 月面试题）

- A. 类中的每个对象都有第一个虚方法的地址，每一个方法都有下一个虚方法的地址。
- B. 类中的每个对象都有一个链表用来保存虚方法地址。



C. 类中的每个对象都保存 5 个虚方法的地址。

D. 类中的每个对象有一个结构用来保存虚方法地址。

**解析：**每个对象里有虚表指针，指向虚表，虚表里存放了虚函数的地址。虚函数表是顺序存放虚函数地址的，不需要用到链表（link list）。

**答案：**B

**面试题 22：**什么虚函数效率低？

**答案：**因为虚函数需要一次间接的寻址，而一般的函数可以在编译时定位到函数的地址。虚函数（动态类型调用）是要根据某个指针定位到函数的地址。多增加了一个过程，效率肯定会低一些，但带来了运行时的多态。

**面试题 23：**请评价多重继承的优点和缺陷。

**答案：**多重继承在语言上并没有什么很严重的问题，但是标准本身只对语义做了规定，而对编译器的细节没有做规定。所以在使用时（即使是继承），最好不要对内存布局等有什么假设。此类的问题还有虚析构造函数等。为了避免由此带来的复杂性，通常推荐使用复合。

（1）多重继承本身并没有问题，如果运用得当可以收到事半功倍的效果。不过大多数系统的类层次往往有一个公共的基类，就像 MFC 中的 Cobject，或 Java 中的 Object。而这样的结构如果使用多重继承，稍有不慎，将会出现一个严重现象——菱形继承，这样的继承方式会使得类的访问结构非常复杂。但并非不可处理，可以用 virtual 继承（并非唯一的方法）及 Loki 库中的多继承框架来掩盖这些复杂性。

（2）从哲学上来说，C++ 多重继承必须要存在，这个世界本来就不是单根的。从实际用途上来说，多重继承不是必需的，但这个世界上有多少东西是必需的呢？对象不过是一组有意义的数据集合及其上的一组有意义的操作，虚函数（晚期绑定）也不过是一堆函数人口表，重载也不过是函数名扩展，这些东西都不是必需的，而且对它们的不当使用都会带来问题。但是没有这些东西行吗？很显然，不行。

（3）多重继承在面向对象理论中并非是必要的——因为它不提供新的语义，可以通过单继承与复合结构来取代。而 Java 则放弃了多重继承，使用简单的 interface 取代。多重继承是把双刃剑，应该正确地对待。况且，它不像 got，不破坏面向对象语义。跟其他任何威力强大的东西一样，用好了会带来代码的极大精简，用坏了那就不用多说了。

C++ 是为实用而设计的，在语言里有很多东西存在着各种各样的“缺陷”。

所以，对于这种有“缺陷”的东西，它的优劣就要看使用它的人。C++ 不回避问题，它只是把问题留给使用者，从而给大家更多的自由。像 Ada、Pascal 这类定义严格的语言，从语法上回避了问题，但并不是真正解决了问题，而使人做很多事时束手束脚（当然，习惯了就好了）。

（4）多重继承本身并不复杂，对象布局也不混乱，语言中都有明确的定义。真正复杂的是使用了运行时多态（virtual）的多重继承（因为语言对于多态的实现没有明确的定义）。为什么非要说多重继承不好呢？如果这样的话，指针不是更容易出错，运行时多态不是更不好理解吗？

因为 C++ 中没有 interface 这个关键字，所以不存在所谓的“接口”技术。但是 C++ 可以很轻松地做到这样的模拟，因为 C++ 中的不定义属性的抽象类就是接口。

（5）要了解 C++，就要明白有很多概念是 C++ 试图考虑但是最终放弃的设计。你会发现很多 Java 和 C# 中的东西都是 C++ 考虑后放弃的。不是说这些东西不好，而是在 C++ 中它将破坏 C++ 作为一个整体的和谐性，或者 C++ 并不需要这样的东西。举一个例子来说明，C# 中有一个关键字 base 用来表示该类的父类，C++ 却没有对应的关键字。为什么没有？其实 C++ 中曾经有人提议用一个类似的关键字 inherited，来表示被继承的类，即父类。这样一个好的建议为什么没有被采纳呢？因为这样的关键字既不必需又不充分。不必需是因为 C++ 有一个 typedef \* inherited，不充分是因为有多个基类，你不可能知道 inherited 指的是哪个基类。

很多其他语言中存在的时髦的东西在 C++ 中都没有，这之中有的是待改进的地方，有的是不需要，我们不能一概而论，需要具体问题具体分析。

**面试题 24：**什么是虚指针？（美国某移动通信企业面试题）

**答案：**虚指针或虚函数指针是一个虚函数的实现细节。带有虚函数的类中的每一个对象都有一个虚指针指向该类的虚函数表。

**面试题 25：**在分布式系统中，不适用 RTTI 的一个合理解释是？（中国某互联网公司 2010 年 6 月面试题）

- A. RTTI 太慢了。
- B. RTTI 不是一个标准行为。
- C. RTTI 行为不可预期及缺乏扩展性。
- D. RTTI 函数在运行时会失败。

**解析：**C++ 引入中的每个特性，都是从程序员平时生活中逐渐精化而来

的。在不正确的场合使用它们必然会引起逻辑、行为和性能上的问题。对于上述特性，应该只在必要、合理的前提下才使用。

C++ 引入的额外开销体现在以下两方面。

### (1) 编译时开销

模板、类层次结构、强类型检查等新特性，以及大量使用了这些新特性的 C++ 模板、算法库都明显地增加了 C++ 编译器的负担。但是应当看到，这些新机能在不增加程序执行效率的前提下，明显降低了广大 C++ 程序员的工作量。

### (2) 运行时开销

运行时开销恐怕是程序员最关心的问题之一了。相对于传统 C 程序而言，C++ 中有可能引入额外运行时开销特性包括：虚基类、虚函数、RTTI（`dynamic_cast` 和 `typeid`）、异常、对象的构造和析构。

虚基类，从直接虚继承的子类中访问虚基类的数据成员或其虚函数时，将增加两次指针引用（大部分情况下可以优化为一次）和一次整型加法的时间开销。定义一个虚基类表，定义若干虚基类表指针的空间开销。

虚函数的运行开销有进行整型加法和指针引用的时间开销。定义一个虚表，定义若干个（大部分情况下是一个）虚表指针的空间开销。

RTTI 的运行开销主要有进行整型比较和取址操作（可能还会有两次整形加法）所增加的时间开销。定义一个 `type_info` 对象（包括类型 ID 和类名称）的空间开销。“`dynamic cast`”用于在类层次结构中漫游，对指针或引用进行自由的向上、向下或交叉转化。“`typeid`”则用于获取一个对象或引用的确切类型。一般地讲，能用虚函数解决的问题就不要用“`dynamic cast`”，能够用“`dynamic_cast`”解决的就不要用“`typeid`”。

关于异常，对于几乎所有编译器来说，在正常情况（未抛出异常）下，try 块中的代码执行效率和普通代码一样高，而且由于不再需要使用传统上通过返回值或函数调用来判断错误的方式，代码的实际执行效率还会进一步提高。抛出和捕捉异常的开销也只是在某些情况下会高于参数返回和函数调用的开销。

关于构造和析构，开销也不总是存在的。对于不需要初始化/销毁的类型，并没有构造和析构的开销，相反对于那些需要初始化/销毁的类型来说，即使用传统的 C 方式实现，也至少需要与之相当的开销。

实事求是地讲，RTTI 是有用的。但因为一些理论上及方法论上的原因，它破坏了面向对象的纯洁性。

首先，它破坏了抽象，使本来不应该被使用的方法和属性被不正确地使用。其次，因为运行时类型的不确定性，它把程序变得更脆弱。第三点，也是最重要的一点，它使程序缺乏扩展性。当加入了一个新的类型时，你也许需要仔细阅读你的 `dynamic_cast` 或 `instanceof` 的代码，必要时改动它们，以保证这个新的类型的加入不会导致问题。而在这个过程中，编译器将不会给你任何帮助。

很多人一提到 RTTI，总是侧重于它的运行时的开销。但是，相比于方法论上的缺点，这点运行时的开销真是无足轻重的。

总的来说，RTTI 因为它的方法论上的一些缺点，它必须被非常谨慎地使用。今天面向对象语言的类型系统中的很多东西就是产生于避免 RTTI 的各种努力。

**答案：C**

**面试题 26：**一个参数可以既是 `const` 又是 `volatile` 吗？解释为什么。

**答案：**第一个问题：可以。一个例子就是只读的状态寄存器。它是 `volatile`，因为它可能被意想不到地改变；它又是 `const`，因为程序不应该试图去修改它。

第二个问题：可以。尽管这并不很正常。一个例子是当一个中断服务子程序修改一个指向一个 `buffer` 的指针时。

**面试题 27：**十进制数  $-10$  的三进制 4 位数补码形式是 \_\_\_\_。

- A.0101
- B.1010
- C.2121
- D.2122

**解析：**负数的补码：对于二进制而言， $-10$  的补码为  $-(2^8 - |-10|) = -(256 - 10) = -246 = 11110110$

同理，4 位三进制的补码： $10$  的补码为  $3^4 - |-10| = 71 = 2212$ 。

**答案：D**

## 5. 树、图、哈希表

**面试题 28：**在百度或淘宝搜索时，每输入字符都会出现搜索建议，例如输入“北京”，搜索框下面会以北京为前缀，展示“北京爱情故事”“北京公交”“北京医院”等搜索词。实现这类技术后台所采用的数据结构是什么

么? (中国某搜索引擎公司 2012 年 6 月笔试题)

**答案:** Trie 树, 又称单词查找树、字典树, 是一种树形结构, 是一种哈希树的变种, 是一种用于快速检索的多叉树结构。典型应用是用于统计和排序大量的字符串 (但不仅限于字符串), 所以经常被搜索引擎系统用于文本词频统计。它的优点是: 最大限度地减少无谓的字符串比较, 查询效率比哈希表高。Trie 树的核心思想是空间换时间。利用字符串的公共前缀来降低查询时间的开销以达到提高效率的目的。

对于搜索引擎, 一般会保留一个单词查找树的前  $N$  个字 (全球或最近热门使用的); 对于每个用户, 保持 Trie 树最近前  $N$  个字为该用户使用的结果。

用户点击任何搜索结果后, Trie 树可以非常迅速并异步获取完整的部分 / 模糊查找, 然后预取数据, 再用一个 Web 应用程序发送一个较小的组结果到浏览器。

**面试题 29:** 用二进制来编码字符串 “abcdabaa”, 需要能够根据编码, 解码回原来的字符串, 最少需要多长的二进制字符串?

- A. 12
- B. 14
- C. 18
- D. 24

**解析:** 哈夫曼编码问题: 字符串 “abcdabaa” 有 4 个 a、2 个 b、1 个 C、1 个 d。构造哈夫曼树如下所示。

a 编码 0 (1 位), b 编码 10 (2 位), c 编码 110 (3 位), d 编码 111 (3 位)。这个字符串的总长度为:  $1*4+2*2+3*1+3*1=14$ 。

**答案:** B

## 6. 字符串

**面试题 30:** 已知 strcpy 函数的原型是 `char *strcpy(char *strDest, const char *strSrc)`; 其中 strDest 是目的字符串, strSrc 是源字符串。不调用 C++/C 的字符串库函数, 请编写函数 strcpy。strcpy 能把 strSrc 的内容复制到 strDest, 为什么还要 char \* 类型的返回值?

**解析:** 先来看第一个问题, 实现一个字符串复制的函数, 基本思路是使用两个指针指向新旧字符串, 然后进行逐一复制。

复制过程中我们需要注意, 字符串的截止条件, 并且需要考虑传入参数

的合法性判断，对意外情况作出处理。

这个题目算是比较简单的，只要仔细注意上边的两个问题即可。

**答案：**

---

```
char *strcpy(char *strDest, const char *strSrc)
{
    if(strDest == NULL || strSrc == NULL)
    {
        return NULL;
    }
    char *p = strDest;
    //也可以用while循环
    for(; *strSrc != '\0'; strDest++, strSrc++){
        *strDest = *strSrc;
    }
    *strDest = '\0'; //复制完毕之后一定要加一个结束符号
    return p;
}
```

---

**面试题 31：**怎样将整数转换成字符串数，并且不用函数 itoa?

**答案：**

---

```
#include <iostream>
using namespace std;

int main ()
{
    int num = 12345, i = 0, j = 0;
    char temp[7], str[7];
    while(num)
    {
        temp[i] = num % 10 + '0';
        //将整数num从后往前的每一位数转换成char保存在temp中
        i++;
        num = num / 10;
    }
    temp[i] = '\0';
    cout << "temp: " << temp << endl;
    i = i - 1;
    //反转temp
    while (i >= 0)
    {
        str[j++] = temp[i--];
    }
}
```

---

```

    }
    str[j]=0;
    cout<<"string:"<<str<<endl;
    return 0;
}

```

如果可以使用 itoa 函数的话，则十分简单，如下所示。

```

#include <iostream>
#include <stdlib.h>
using namespace std;

//使用 itoa 函数
int main ()
{
    int num=12345;
    char str[7];

    itoa(num,str,10);
    cout<<"integer:"<<num<<endl<<"string:"<<str<<endl;
    return 0;
}

```

### 面试题 32：模拟实现 strstr。（找子串）

**答案：**

```

{
    int m = haystack.size(), n = needle.size();
    if (n == 0)
        return 0;
    for (int i = 0; i < m - n + 1; i++)
    {
        int j = 0;
        while(haystack[i + j] == needle[j])
        {
            j++;
            if (j == n)
                return i;
        }
        j++;
    }
    return -1;
}

```

**面试题 33：模拟实现 atoi。****答案：**

---

```
int my_atoi(const char* str)
{
    int num = 0;
    int sign = 1;
    const int n = strlen(str);
    int i = 0;

    while (str[i] == ' ' && i < n)
        i++;
    if (str[i] == '+')
        i++;
    else if (str[i] == '-')
    {
        sign = -1;
        i++;
    }
    for (; i < n; i++)
    {
        if (str[i] < '0' || str[i] > '9')
            break;
        if (num > INT_MAX / 10 || (num == INT_MAX / 10 && (str[i] - '0') > INT_
MAX % 10))
            return sign == -1 ? INT_MIN : INT_MAX;
        num = num * 10 + str[i] - '0';
    }
    return num * sign;
}
```

---

**面试题 34：找出最长回文子串。**

**解析：**本题的解决方法是从头到尾地遍历字符串 S，以每个字符为中心，向两端不停地扩展，同时判断以当前字符为中心的两端字符是否相等，在判断的过程中找到最大的回文串长度，并记录下回文串开始的最左端，这样就找到了最大的回文串。但是这样做就遇到了一个问题：“abcb”这种个数为奇数的回文串可以计算出来，但是若是“abba”这样个数为偶数的情况，该如何计算？遇到这种情况，则可以判断相邻两字符是否相等来判断是否为回文串，针对 s = “abba”，我们发现 i=1 时，s[i]=s[i+1]，然后同时向两端扩，发现 s[0]=s[2]，这样该串也为回文串，所以，在判断以某字符为中



心的时候，要分两种情况，即回文串中字符的个数为奇数和为偶数的情况。  
时间复杂度是  $O(n^2)$ 。

**答案：**

```
//最长回文串
string longestPalindrome(string s)
{
    string res = "";
    int len = s.size();
    if (len == 1)
        return s;
    int maxlen = 0, curlen = 0, sbegin;
    int left, right;
    for (int i = 0; i < len; ++i)
    {
        if (len % 2 == 0)
        {
            left = i - 1;
            right = i + 1;
        }
        else
        {
            left = i;
            right = i + 1;
        }
        while (left >= 0 && right < len && s[left] == s[right])
        {
            curlen = right - left;
            if (curlen > maxlen)
            {
                maxlen = curlen;
                sbegin = left;
            }
            left--;
            right++;
        }
    }
    res = s.substr(sbegin, maxlen + 1);
    //substring()为前闭后开
    return res;
}
```

## 7. 软件测试

**面试题 35：**自动化测试为何重要？（美国某数据库公司 2009 年 11 月面试题）

**解析：**自动化测试从根本上提高 QA 们的职业素质，让 QA 们彻底摆脱重复繁重的测试工作，而更着重在 QA 的流程上已经完成项目质量的重复保证上。此外某些人对 QA 有些偏见：对 QA 的普遍认识就是只是测试而已。因为他们能看到 QA 最直接的劳动就是在反反复复勤勤恳恳地测试。

**答案：**自动化测试可以让测试人员从枯燥无味的手工重复性测试中解放出来，并且提高工作效率，通过自动化测试结果来分析功能和性能上的缺陷。

**面试题 36：**描述一个测试结束的准则。（美国某数据库公司 2009 年 11 月面试题）

**答案：**一个测试结束的标准是已提交的 bug 是否已经全部解决并已验证关闭。一般来说，bug 验证率在 95% 以上，并且没有大的影响功能的 bug 处于未解决状态，就可以测试通过。

**面试题 37：**在一个测试计划中能包含哪些内容，如可用的人力资源？（美国某数据库公司 2009 年 11 月面试题）

**答案：**在一个测试计划中可以包含需要测试的产品的主要功能模块，列出需要测试的功能点，并标明侧重点；测试的策略和记录（测试工具的确认，测试用例等文档模板，测试方法的确定）；测试资源配置（确定测试每一阶段的任务和所需资源）。

**面试题 38：**请描述功能测试和可用性测试之间的区别。（美国某数据库公司 2009 年 11 月面试题）

**解析：**本题涉及几个测试的重要概念：

Functional testing（功能测试），也称为 behavioral testing（行为测试），根据产品特征、操作描述和用户方案，测试一个产品的特性和可操作行为以确定它们满足设计需求。本地化软件的功能测试，用于验证应用程序或网站对目标用户能正确工作。使用适当的平台、浏览器和测试脚本，以保证目标用户的体验将足够好，就像应用程序是专门为该市场开发的一样。

功能测试也叫黑盒子测试或数据驱动测试，只需考虑各个功能，不需要考虑整个软件的内部结构及代码。一般从软件产品的界面、架构出发，按照

需求编写出来的测试用例，输入数据在预期结果和实际结果之间进行评测，进而提出使产品更加符合用户使用的要求。

可用性测试是用户在和系统（网站、软件应用程序、移动技术或任何用户操作的设备）交互时对用户体验质量的度量。可用性（Usability）是交互式 IT 产品 / 系统的重要质量指标，指的是产品对用户来说有效、易学、高效、好记、少错和令人满意的程度，即用户能否用产品完成他的任务？效率如何？主观感受怎样？实际上是从用户角度所看到的产品质量，是产品竞争力的核心。

**答案：**功能测试主要是黑盒测试，由测试人员进行，主要验证产品是否符合需求设计的要求；可用性测试主要是由用户（或者测试人员模拟用户行为）来进行的测试，主要是对产品的易用性进行测试，包括有效性（effectiveness）、效率（efficiency）和用户主观满意度（satisfaction）。其中有效性指用户完成特定任务和达到特定目标时所具有的正确和完整程度；效率指用户完成任务的正确和完整程度与所使用资源（如时间）之间的比率，满意度指用户在使用产品过程中所感受到的主观满意和接受程度。

**面试题 39：黑盒测试和白盒测试的区别。**

**答案：**白盒测试是通过程序的源代码进行测试而不使用用户界面。这种类型的测试需要从代码句法发现内部代码在算法、溢出、路径、条件等中的缺点或者错误，进而加以修正。

黑盒测试是通过使用整个软件或某种软件功能来严格进行测试，而并不检查程序的源代码，也不很清楚地了解该软件的源代码程序具体是怎样设计的。

## 8. 操作系统

**面试题 40：试解释操作系统原理中的作业、进程、线程、管程各自的定义。（中国某通讯公司 2009 年 10 月笔试题）**

**答案：**作业：用户在一次解题或一个事务处理过程中要求计算机系统所做工作的集合。它包括用户程序、所需要的数据及控制命令等。作业是由一系列有序的步骤组成的。

进程：一个程序在一个数据集合上的一次运行过程。所以一个程序在不同数据集合上运行，乃至一个程序在同样数据集合上的多次运行都是不同的进程。

**线程**：线程是进程中的一个实体，它是被系统独立调度和执行的基本单位。

**管程**：管程实际上是定义了一个数据结构和在该数据结构上的能为并发进程所执行的一组操作，这组操作能同步进程和改变管程中的数据。

**面试题 41**：进程间的通信如何实现？（日本某家电 / 通信 / IT 企业面试题）

**答案**：现在最常用的进程间通信的方式有信号、信号量、消息队列、共享内存。

所谓进程通信，就是不同进程之间进行一些“接触”。这种接触有简单，也有复杂。机制不同，复杂度也不一样。通信是一个广义上的意义，不仅仅指传递一些 message。它们的使用方法是基本相同的，所以只要掌握了一种使用方法，然后记住其他的使用方法就可以了。信号和信号量是不同的，它们虽然都可用来实现同步和互斥，但前者是使用信号处理器来进行的，后者是使用 P、V 操作来实现的。消息队列是比较高级的一种进程间通信方法，因为它真的可以在进程间传送 message，连传送一个“I seek you”都可以。

一个消息队列可以被多个进程所共享（IPC 就是在这个基础上进行的）；如果一个进程的消息太多，一个消息队列放不下，也可以用多于一个的消息队列（不过可能管理会比较复杂）。共享消息队列的进程所发送的消息中除了 message 本身外还有一个标志，这个标志可以指明该消息将由哪个进程或者是哪类进程接受。每一个共享消息队列的进程针对这个队列也有自己的标志，可以用来声明自己的身份。

**面试题 42**：在 Windows 编程中互斥器(mutex)的作用和临界区(critical section)类似，请说一下二者间的主要区别。（中国某杀毒软件公司 2005 年面试题）

**解析**：多线程编程问题。

**答案**：两者的区别是 mutex 可以用于进程之间互斥，critical section 是线程之间的互斥。

**面试题 43**：请描述进程和线程的差别。（美国某软件公司 2005 年面试题）

**答案**：进程是程序的一次执行。线程可以理解为进程中执行的一段程序片段。在一个多任务环境中，下面的概念可以帮助我们理解两者间的差别。

进程间是独立的，这表现在内存空间、上下文环境上；线程运行在进程空间内。一般来讲（不使用特殊技术），进程无法突破进程边界存取其他进程内的存储空间；而线程由于处于进程空间内，所以同一进程所产生的线程共享同一内存空间。

同一进程中的两段代码不能够同时执行，除非引入线程。

线程是属于进程的，当进程退出时该进程所产生的线程都会被强制退出并清除。线程占用的资源要少于进程所占用的资源。进程和线程都可以有优先级。

进程间可以通过 IPC 通信，但线程不可以。

**面试题 44：**下面哪个选项不是 PE 文件？

- A. EXE
- B. DLL
- C. COM
- D. DOC

**解析：**PE 文件被称为可移植的执行体，是 Portable Execute 的全称，常见的 EXE、DLL、OCX、SYS、COM 都是 PE 文件。PE 文件是微软 Windows 操作系统上的程序文件（可能是间接被执行，如 DLL）

**答案：**D

**面试题 45：**Windows 将遵循下面的哪种搜索来定位 DLL？

- (1) 进程的当前工作目录
- (2) 包含 EXE 文件的目录
- (3) 列在 Path 环境变量中的一系列目录
- (4) Windows 系统目录
- (5) Windows 目录

A.12453 B.12543 C.21453 D.21345

**解析：**Windows 平台的大多数程序都使用各种动态链接库（DLL）来避免重复实现功能。操作系统为每个程序加载若干个 DLL，具体由程序的类型决定。当程序不指定 DLL 的绝对位置时，将使用默认的搜索顺序来找到它。默认情况下，操作系统所使用的搜索顺序为：

- (1) 内存
- (2) KnownDLLs

- (3) 清单与 .local
- (4) 应用程序目录
- (5) 当前工作目录
- (6) 系统目录
- (7) 路径变量

**答案：**C

**面试题例 46：**简述 Windows 内存管理的几种方式和优缺点。（中国某互联网公司面试题）

**答案：**Windows 内存管理方式主要分为：页式管理、段式管理、段页式管理。

页式管理的基本原理是将各进程的虚拟空间划分成若干个长度相等的页（page）；页式管理把内存空间按页的大小划分成片或者页面，然后把页式虚拟地址与内存地址建立一一对应的页表；并用相应的硬件地址变换机构来解决离散地址变换问题。页式管理采用请求调页或预调页技术来实现内外存储器的统一管理。其优点是没有外碎片，每个内碎片不超过页的大小。缺点是程序全部装入内存，要求有相应的硬件支持。例如地址变换机构缺页中断的产生和选择淘汰页面等都要求有相应的硬件支持。这增加了机器成本，也增加了系统开销。

段式管理的基本思想就是把程序按内容或过程函数关系分成段，每段有自己的名字。一个用户作业或进程所包含的段对应一个二维线形虚拟空间，也就是一个二维虚拟存储器。段式管理程序以段为单位分配内存，然后通过地址影射机构把段式虚拟地址转换为实际内存物理地址。其优点是可以分别编写和编译，可以针对不同类型的段采取不同的保护，可以按段为单位来进行共享，包括通过动态链接进行代码共享。缺点是会产生碎片。

段页式管理：为了实现段页式管理，系统必须为每个作业或进程建立一张段表以管理内存分配与释放、缺段处理等。另外由于一个段又被划分成了若干页。每个段又必须建立一张页表以把段中的虚页变换成内存中的实际页面。显然与页式管理时相同，页表中也要有相应的实现缺页中断处理和页面保护等功能的表项。段页式管理是段式管理与页式管理方案结合而成的，所以具有它们两者的优点。但反过来说，由于管理软件的增加，复杂性和开销也就随之增加了。另外需要的硬件以及占用的内存也有所增加，使得执行速度下降。

## 9. 数据库与 SQL 语言

**面试题 47：**存储过程和函数的区别是什么？（美国某搜索引擎公司面试题）

**答案：**存储过程是用户定义的一系列 SQL 语句的集合，涉及特定表或其他对象的任务，用户可以调用存储过程。而函数通常是数据库已定义的方法，它接收参数并返回某种类型的值并且不涉及特定用户表。

**面试题 48：**游标的作用是什么，如何知道游标已经到最后？（中国某计算机金融软件公司面试题）

**答案：**游标用于定位结果集的行。通过判断全局变量 @@FETCH\_STATUS 可以判断其是否到了最后。通常此变量不等于 0 表示出错或到了最后。

**面试题 49：**触发器分为事前触发和事后触发，这两种触发有何区别？语句级触发和行级触发有何区别？（美国某计算机软件公司面试题）

**答案：**事前触发器运行于触发事件发生之前，而事后触发器运行于触发事件发生之后。语句级触发器可以在语句执行前或后执行，而行级触发在触发器所影响的每一行触发一次。

**面试题 50：**解释聚集索引和非聚集索引之间的区别。

**答案：**经典教科书对聚集索引的解释是，聚集索引的顺序就是数据的物理存储顺序。而对于非聚集索引的解释是索引顺序与数据物理排列顺序无关。正是因为如此，一个表最多只能有一个聚集索引

在 SQL Server 中，索引是通过二叉树的数据结构来描述的，我们可以这么理解聚集索引：索引的叶节点就是数据节点。而非聚集索引的叶节点仍然是索引节点，只不过有一个指针指向对应的数据块。

聚集索引确定表中数据的物理顺序。聚集索引类似于电话簿（电话簿按照字母簿排序），后者按姓氏排列数据。由于聚集索引规定数据在表中的物理存储顺序，因此一个表只能包含一个聚集索引。但该索引可以包含多个列（组合索引），就像电话簿按姓氏和名字进行组织一样。

聚集索引对于那些经常要搜索范围值的列特别有效。使用聚集索引找到包含第一个值的行后，便可以确保包含后续索引值的行在物理相邻。例如，如果应用程序执行的一个查询经常检索某一日期范围内的记录，则使用聚集

索引可以迅速找到包含开始日期的行，然后检索表中所有相邻的行，直到到达结束日期。这样有助于提高此类查询的性能。同样，如果对从表中检索的数据进行排序时经常要用到某一列，则可以将该表在该列上聚集（物理排序），避免每次查询该列时都进行排序，从而节省成本。

使用非聚集索引，非聚集索引与课本中的索引类似。数据存储在一个地方，索引存储在另一个地方，索引带有指针指向数据的存储位置。索引中的项目按索引键值的顺序存储，而表中的信息按另一种顺序存储（这可以由聚集索引规定）。如果在表中未创建聚集索引，则无法保证这些行具有任何特定的顺序。

有索引就一定检索得快吗？答案是否定的。有些时候用索引还不如不用索引快。比如说我们要检索表中的所有 8000 条记录，如果不用索引，需要访问 8000 条  $\times$  1000 字节 / 8K 字节 = 1000 个页面。如果使用索引的话，首先检索索引，访问 8000 条  $\times$  10 字节 / 8K 字节 = 10 个页面得到索引检索结果，再根据索引检索结果去对应数据页面。由于是检索所有数据，所以需要再访问 8000 条  $\times$  1000 字节 / 8K 字节 = 1000 个页面将全部数据读取出来，共访问了 1010 个页面。这显然不如不用索引快。

## 10. 计算机网络及分布式系统

**面试题 51：**在 OSI 参考模型中，物理层的作用是

（1）。对等实体在一次交互作用中传送的信息单位称为（2），它包括（3）两部分。上下层实体之间的接口称为服务访问点（SAP），网络层的服务访问点也称为（4），通常分为（5）两部分。（中国某综合软件公司 2005 年面试）

- （1）A. 建立和释放连接    B. 透明传输比特流  
C. 协议数据单元    D. 发送和接收用户
- （2）A. 接口数据    B. 服务数据单元  
C. 协议数据单元    D. 交互数据单元
- （3）A. 控制信息和用户数据    B. 接口信息和用户数据  
C. 接口信息和控制信息    D. 控制信息和效验信息
- （4）A. 用户地址    B. 网络地址  
C. 端口地址    D. 网卡地址
- （5）A. 网络号和端口号    B. 网络号和主机地址  
C. 超网号和子网号    D. 超网号和端口地址

**解析：**网络问题。



OSI 参考模型有 7 层。其分层原则如下。

- (1) 根据不同层次的抽象分层。
- (2) 每层应当有一个定义明确的功能。
- (3) 每层功能的选择应该有助于制定网络协议的国际标准。
- (4) 各层边界的选择应尽量节省跨过接口的通信量。
- (5) 层数应足够多, 以避免不同的功能混杂在同一层中, 但也不能太多, 否则体系结构会过于庞大。

根据以上标准, OSI 参考模型分为物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

- 物理层涉及在信道上传输的原始比特流。
- 数据链路层的主要任务是加强物理层传输原始比特流的功能, 使之对应的网络层显现为无错线路。发送包把输入数据封装在数据帧, 按顺序传出去并处理接收方回送的确认帧。
- 网络层关系到子网的运行控制, 其中一个关键问题是确认从源端到目的端如何选择路由。
- 传输层的基本功能是从会话层接收数据而且把其分成较小的单元传递给网络层。
- 会话层允许不同机器上的用户建立会话关系。
- 表示层用来完成某些特定的功能
- 应用层包含着大量人们普遍需要的协议。

**答案:** B,C,A,B,B。

**面试题 52:** TCP 和 UDP 有什么区别? (中国某金融软件公司 2005 年面试题)

**解析:** 举例说明两者间的区别。

TCP 连接就像打电话, 两者之间必须有一条不间断的通路, 数据不到达对方, 对方就一直在等待, 除非对方直接挂电话。先说的话先到, 后说的话后到, 有顺序。

UDP 就像寄一封信, 发信者只管发, 不管到。但是你的信封上必须写明对方的地址。发信者和收信者之间没有通路, 靠邮局联系。信发到时可能已经过了很久, 也可能根本没有发到。先发的信未必先到, 后发的也未必后到。

**答案:** TCP 是传输控制协议, 提供的是面向连接、可靠的字节流服务。客户和服务彼此交换数据前, 必须先在双方之间建立一个 TCP 连接, 之

后才能传输数据。TCP 提供超时重发、丢弃重复数据、检验数据、流量控制等功能，保证数据能从一端传到另一端。

UDP 是用户数据报协议，是一个简单的面向数据报的运输层协议。UDP 不提供可靠性，它只是把应用程序传给 IP 层的数据报发送出去，但是并不保证它们能到达目的地。由于 UDP 在传输数据报前不用在客户和服务端之间建立一个连接，且没有超时重发等机制，故而传输速度很快。

## 11. 网络协议

**面试题 53：**如果蠕虫病毒攻击了一个家用 PC 的 A 类地址主机的话，这个地址最有可能接收很多 \_\_\_\_。（中国某杀毒软件公司 2005 年 10 月面试题）

- A.HTTP 回应包
- B.DNS 回应包
- C.ICMP 目的无法抵达包
- D.ARP 回应

**解析：**大量发出 IP 请求，肯定很多不可达，返回不可达错误。

**答案：**C

**面试题 54：**如何编写 Socket 套接字？（中国某通信企业 2008 年面试题）

**解析：**Socket 相当于进行网络通信两端的插座，只要对方的 Socket 和自己的 Socket 有通信联接，双方就可以发送和接收数据了。其定义类似于文件句柄的定义。如果你要编写的是一个服务程序，那么先调用 `socket()` 创建一个套接字，调用 `bind()` 绑定 IP 地址和端口，然后启动一个死循环，循环中调用 `accept()` 接受连接。对于每个接受的连接，可以启动多线程方式进行处理，在线程中调用 `send()`、`recv()` 发送和接收数据。

如果你要编写的是一个客户端程序，那么就简单多了。先调用 `socket()` 创建一个套接字，然后调用 `connect()` 连接服务器，之后就是调用 `send()`、`recv()` 发送和接收数据了。

**答案：**服务器端程序编写如下。

（1）调用 `ServerSocket(int port)` 创建一个服务器端套接字，并绑定到指定端口上。

（2）调用 `accept()`，监听连接请求，接收连接，返回通信套接字。

(3) 调用 Socket 类的 `getOutputStream()` 和 `getInputStream` 获取输出流和输入流, 开始网络数据的发送和接收。

(4) 关闭通信套接字 `.Socket.close()`。

客户端程序编写:

调用 `Socket()` 创建一个流套接字, 并连接到服务器端。

调用 Socket 类的 `getOutputStream()` 和 `getInputStream` 获取输出流和输入流, 开始网络数据的发送和接收。

关闭通信套接字 `.Socket.close()`。

## 12. 网络安全

**面试题 55:** 入侵检查与防火墙有何不同, 各有什么优缺点? (中国某通信企业 2008 年面试题)

**答案:** 防火墙的优点: 它能增强机构内部网络的安全性, 用于加强网络间的访问控制, 防止外部用户非法使用内部网的资源, 保护内部网络的设备不被破坏, 防止内部网络的敏感数据被窃取。防火墙系统决定了哪些内部服务可以被外界访问; 外界的哪些人可以访问内部的哪些服务, 以及哪些外部服务可以被内部人员访问。

防火墙的缺点: 对于发生在内网的攻击无能为力; 部分攻击可以绕过防火墙, 防火墙发现不了; 防火墙的策略是静态的, 不能实施动态防御; 等等。

入侵检测的优势: 入侵监测系统扫描当前网络的活动, 监视和记录网络的流量, 根据定义好的规则来过滤从主机网卡到网线上的流量, 提供实时报警。大多数的入侵检测系统可以提供关于网络流量非常详尽的分析。它们可以监视任何定义好的流量。很多系统对 FTP、HTTP 和 Telnet 流量都有默认的设置, 还有其他的流量, 如 NetBus、本地和远程登录失败, 等等。也可以自己定制策略。如果定义了策略和规则, 便可以获得 FTP、SMTP、Telnet 和任何其他的流量。这种规则有助于追查该连接和确定网络上发生过什么, 以及现在正在发生什么。这些程序在需要确定网络中策略实施的一致性情况时是非常有效的工具。

入侵检测的缺点: 目前入侵检测技术的方法主要停留在异常检测统计方法和误用检测方法上, 这两种方法都还存在这样或那样的问题。网络入侵技术在不断地发展, 入侵的行为表现出不确定性、多样性等特点。网络应用的发展又带来新的安全问题。如高速网络技术出现流量大的特点, 那么基于网络的入侵检测系统如何适应这种情况? 基于主机审计数据怎样做到既减少数

据量，又能有效地检测到入侵？入侵检测研究领域急需其他学科知识提供新的入侵检测解决方法。入侵检测只是试图发现计算机网络中的安全问题，要解决网络安全的问题还需要其他的网络安全技术。另外，入侵检测系统本身还存在安全问题。入侵检测系统也可能会受到攻击。

综上所述，其实防火墙和入侵检测各有优劣。打个比方，防火墙就相当于一栋大楼外的门卫系统，而入侵检测就相当于大楼内的监控系统，两者缺一不可。应该将入侵检测系统与防火墙联动起来，当入侵检测系统发现有入侵行为时，应及时报告防火墙，以阻断入侵。

**面试题 56：**端口是做什么用的，有什么漏洞吗？（某中美合资通信企业 2008 年面试题）

**答案：**25 端口为 SMTP（Simple Mail Transfer Protocol，简单邮件传输协议）服务器所开放，主要用于发送邮件，如今绝大多数邮件服务器都使用该协议。例如在使用电子邮件客户端程序的时候，在创建账户时会要求输入 SMTP 服务器地址，该服务器地址默认情况下使用的就是 25 端口。

端口漏洞：利用 25 端口，黑客可以寻找 SMTP 服务器，用来转发垃圾邮件。25 端口被很多木马程序开放，例如 Ajan、Antigen、Email Password Sender、ProMail、Trojan、Tapiras、Terminator、WinPC、WinSpy 等。拿 WinSpy 来说，通过开放 25 端口，可以监视计算机正在运行的所有窗口和模块。

### 13. 英语面试

现在不管是国企还是外企，在招聘时都非常看重应聘者的英语交际能力，公司往往通过英语面试，对应聘者的英语交际能力进行考查。下面对参加英语面试的应聘者提出 4 个建议。

**建议一：精心设计一个自然的开场白。**

例如：

C（应聘者）：May I come in?（我能进来吗？）

I（考官）：Yes, please. Oh, you are Jin Li, aren't you?（请进。哦，你是李劲吧？）

C: Yes, I am.（对，我是。）

I: Please sit here, on the sofa.（请坐在沙发上。）

C: Thank you.（谢谢。）

采用“Excuse me. Is this personnel department?”（请问，这里是人力资源部吗？）或“Excuse me for interrupting you. I'm here for an

interview as requested.”（不好意思打搅了，我是依约来应聘的。）类似的问法都比较合适。在确定了面试场所和面试官之后，简洁地用“Good morning / afternoon”向面试官打招呼，也能令在场的人加深对你的印象。

### **建议二：不要害怕外表冷冰冰的考官。**

一些用人单位与面试者的最初交流是比较冷冰冰的。例如：

I: Your number and name, please. ( 请告知你的号码和名字。)

C: My number is sixteen and my name is Zhixin Zhang. ( 我是 16 号，我叫张志新。)

这时，面试者不要觉得有压力，面试官的态度冷漠并不是针对你个人，你只需要照实简洁地回答即可。在某些情况下，面试官会问些看起来比较普通和随意的问题，但实际上是暗藏深意的，例如：

How did you come? A very heavy traffic? ( 你怎么来的？路上很堵吧？)

你的回答可以是这样：“Yes, it was heavy but since I came here yesterday as a rehearsal, I figured out a direct bus line from my school to your company, and of course, I left my school very early so it doesn't matter to me.”（是的，很堵。但我昨天已经预先来过一次并且找到了一条从我们学校直达贵公司的公交路线，并且在今天提早从学校出发。因此，路上的拥堵并没有影响到我。）这样的回答就点出了你的计划性及细心程度。

### **建议三：巧用过渡语，表明自己用心听问题。**

面试者在面试时可以用一些类似“As you mentioned”（正如您所说的）或者“As far as I know”（据我所知）之类的句子，表示你一直在认真听对方的谈话。此外，你还可以选择“As it is shown in my resume”（正如我的简历所提到的）或“As my previous experience shows”（如我之前的工作经验所示）之类的表达法。

另外，在面试之前，面试者对应聘公司应有所了解。例如公司的规模、业务、未来发展等，这些往往被面试者忽略了。对公司文化有深刻理解，是你超出其他应聘者的一个亮点。但如果实在不了解，就应根据所知诚实回答。

**面试题 57:** Can you sell yourself in two minutes? Go for it. ( 你能在两分钟内自我推荐吗？大胆试试吧！)

A: With my qualifications and experience, I feel I am hardworking, responsible and diligent in any project I undertake. Your organization

could benefit from my analytical and interpersonal skills. ( 依我的资格和经验, 我觉得我对所从事的每一个项目都很努力、负责、勤勉。我的分析能力和与人相处的技巧, 对贵单位必有价值。 )

**面试题 58:** Tell us about your project experiences? ( 告诉我你的项目经验是什么? )

A: X X Personnel Managing System is a system of auto-manage the persons' information. it is designed by PowerDesign. The project is a C/S architecture system. It is based on a Microsoft SQL database, and the UI is developed by Delphi 7. In this project, I designed the schema of database, programmed database connectivity using Delphi 7 and ADO. (X X 人事管理系统是一个自动的人事信息管理系统, 它是用 PowerDesign 设计的。整个项目是 C/S 系统架构。它的后台基于微软的 SQL 数据库, 前台设计由 Delphi 7 完成。在这个项目中我负责系统的架构及数据库的连接。)

A: Base on ASP.NET+SQL 2000, We finished XX University Network Course-selected System. Everybody in this school can select, cancel, query course in network. The project is a B/S architecture system; the code is developed by Visual C#, and run on the .NET plat. In this project, I used the ADO interface which is provided by the Database program and after that, I joined the testing of whole system. ( 基于 ASP.NET+SQL 2000 的平台, 我们实现了 X X 大学网络选课系统。学校内的任何人都能够在网上选择、取消、查询课程。整个项目是 B/S 的系统架构。项目基于 .NET 平台, 前端代码是用 C# 完成的。在项目中我们使用 ADO 接口实现数据库的支持。整个系统架设结束后, 我参加了系统的测试工作。)

**面试题 59:** Give me a summary of your current job description. ( 对你目前的工作, 能否做个概括的说明。 )

A: I have been working as a computer programmer for five years. To be specific, I do system analysis, trouble shooting and provide software support. ( 我干了 5 年的电脑程序员。具体地说, 我做系统分析、解决问题及软件供应方面的支持。 )

**面试题 60:** Why did you leave your last job? ( 你为什么离职呢? )

A: Well, I am hoping to get an offer of a better position. If opportunity

knocks, I will take it. (我希望能获得一份更好的工作。如果机会来临,我会抓住。)

A: I feel I have reached the “glass ceiling” in my current job. I feel there is no opportunity for advancement. (我觉得目前的工作已经达到顶峰,即没有升迁的机会。)

**面试题 61:** How do you rate yourself as a professional? (作为一位专业人员,你如何评估自己呢?)

A: With my strong academic background, I am capable and competent. (凭借我良好的学术背景,我可以胜任自己的工作,而且我认为自己很有竞争力。)

A: With my teaching experience, I am confident that I can relate to students very well. (依我的教学经验,我相信能与学生相处得很好。)

A: My background has been focused on preparing me for the IT field, so I can exhibit my ability right away. I already have obtained the educational technology and skills I am confident of my ability to learn quickly in any assignment which I'm not familiar for the moment. (我的背景使我非常适合 IT 领域,我会在此展现我的能力。我已经获得了教育方面的能力和技巧,我确信我的能力能够迅速地学习那些我暂时不了解的任务。)

A: I realize that there are many other college students who have the ability to do this job. I also have that ability. But I also bring an additional quality that makes me the very best person for the job — my attitude for excellence. I am a multi-tasked individual who work well under pressure. My ability to be trained in any area would definitely be a good reason to hire me to work for this firm. (我知道有很多大学生有能力去做这个工作。我也有这个能力。但是相对于这些人而言,我的积极态度决定了我是做这件工作的最佳人选。我是一个可以承担多种任务压力的人。我可以胜任公司的各个领域并为公司努力工作。)

**面试题 62:** What contribution did you make to your current (previous) organization? (你对目前/从前的工作单位有何贡献?)

A: I have finished three new projects, and I am sure I can apply my experience to this position. (我已经完成了 3 个新项目,我相信我能将我的经验用在这份工作上。)



**面试题 63:** What do you think you are worth to us? (你如何知道我们对我们有价值呢?)

A: I feel I can make some positive contributions to your company in the future. (我觉得我对贵公司能做些积极的贡献。)

**面试题 64:** What make you think you would be a success in this position? (你如何知道你能胜任这份工作?)

A: My graduate school training combined with my internship should qualify me for this particular job. I am sure I will be successful. (我在研究所的训练, 加上实习工作, 使我适合这份工作。我相信我能成功。)

**面试题 65:** Are you a multi-tasked individual? (你是一位可以同时承担数项工作的人吗?) Do you work well under stress or pressure? (你能承受工作上的压力吗?)

A: Yes, I think so. The trait is needed in my current (or previous) position and I know I can handle it well. (这种特点就是我目前 (先前) 工作所需要的, 我知道我能应付自如。)

**面试题 66:** What will it take to attain your goals, and what steps have you taken toward attaining them? (你将通过什么手段达到你的成功? 你将采取哪些步骤?)

A: I have finished writing a book recently. And currently I am learning a lot of network certification. I obtained my first certification through self-study, so I am learning and keeping up with the latest technology trends. To be successful in this career, one should have to be an excellent problem solver, critical thinker, and team oriented. (我刚刚写完一本书。目前我在做一个网络认证, 是完全通过自学完成的第一个认证, 所以我一直学习, 与最新的技术同步。为了在职场取得成功, 我们必须成为一个优秀的问题解决专家, 一个具有批判性的思考者, 并以团队利益为主导。)